intel®

Microcomputer Boards and Systems

Order Number 280407-004

intel®

Microcomputer Boards and Systems

1990

LITERATURE

To order Intel Literature or obtain literature pricing information in the U.S. and Canada call or write Intel Literature Sales. In Europe and other international locations, please contact your local sales office or distributor.

INTEL LITERATURE SALES
P.O. BOX 7641
Mt. Prospect, IL 60056-7641

In the U.S. and Canada
call toll free
(800) 548-4725

CURRENT HANDBOOKS

Product line handbooks contain data sheets,application notes, article reprints and other design information.

TITLE	LITERATURE ORDER NUMBER
SET OF 11 HANDBOOKS (Available in U S and Canada only)	**231003**
EMBEDDED APPLICATIONS	270648
8-BIT EMBEDDED CONTROLLERS	270645
16-BIT EMBEDDED CONTROLLERS	270646
16/32-BIT EMBEDDED PROCESSORS	270647
MEMORY	210830
MICROCOMMUNICATIONS (2 volume set)	231658
MICROCOMPUTER SYSTEMS	280407
MICROPROCESSORS	230843
PERIPHERALS	296467
PRODUCT GUIDE (Overview of Intel's complete product lines)	210846
PROGRAMMABLE LOGIC	296083
ADDITIONAL LITERATURE (Not included in handbook set)	
AUTOMOTIVE SUPPLEMENT	231792
COMPONENTS QUALITY/RELIABILITY HANDBOOK	210997
INTEL PACKAGING OUTLINES AND DIMENSIONS (Packaging types, number of leads, etc)	231369
INTERNATIONAL LITERATURE GUIDE	E00029
LITERATURE PRICE LIST (U.S. and Canada) (Comprehensive list of current Intel Literature)	210620
MILITARY (2 volume set)	210461
SYSTEMS QUALITY/RELIABILITY	231762

intel®

Intel the Microcomputer Company:

When Intel invented the microprocessor in 1971, it created the era of microcomputers. Whether used in embedded applications such as automobiles or microwave ovens, or as the CPU in personal computers or supercomputers, Intel's microcomputers have always offered leading-edge technology. Intel continues to strive for the highest standards in memory, microcomputer components, modules and systems to give its customers the best possible competitive advantages.

MICROCOMPUTER BOARDS AND SYSTEMS HANDBOOK

1990

Additional copies of this manual or other Intel literature may be obtained from:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

CUSTOMER SUPPORT

INTEL'S COMPLETE SUPPORT SOLUTION WORLDWIDE

Customer Support is Intel's complete support service that provides Intel customers with hardware support, software support, customer training, consulting services and network management services. For detailed information contact your local sales offices.

After a customer purchases any system hardware or software product, service and support become major factors in determining whether that product will continue to meet a customer's expectations. Such support requires an international support organization and a breadth of programs to meet a variety of customer needs. As you might expect, Intel's customer support is quite extensive. It can start with assistance during your development effort to network management. 100 Intel sales and service offices are located worldwide—in the U.S., Canada, Europe and the Far East. So wherever you're using Intel technology, our professional staff is within close reach.

HARDWARE SUPPORT SERVICES

Intel's hardware maintenance service, starting with complete on-site installation will boost your productivity from the start and keep you running at maximum efficiency. Support for system or board level products can be tailored to match your needs, from complete on-site repair and maintenance support to economical carry-in or mail-in factory service.

Intel can provide support service for not only Intel systems and emulators, but also support for equipment in your development lab or provide service on your product to your end-user/customer.

SOFTWARE SUPPORT SERVICES

Software products are supported by our Technical Information Service (TIPS) that has a special toll free number to provide you with direct, ready information on known, documented problems and deficiencies, as well as work-arounds, patches and other solutions.

Intel's software support consists of two levels of contracts. Standard support includes TIPS (Technical Information Phone Service), updates and subscription service (product-specific troubleshooting guides and; *COMMENTS Magazine*). Basic support consists of updates and the subscription service. Contracts are sold in environments which represent product groupings (e.g., iRMX® environment).

CONSULTING SERVICES

Intel provides field system engineering consulting services for any phase of your development or application effort. You can use our system engineers in a variety of ways ranging from assistance in using a new product, developing an application, personalizing training and customizing an Intel product to providing technical and management consulting. Systems Engineers are well versed in technical areas such as microcommunications, real-time applications, embedded microcontrollers, and network services. You know your application needs; we know our products. Working together we can help you get a successful product to market in the least possible time.

CUSTOMER TRAINING

Intel offers a wide range of instructional programs covering various aspects of system design and implementation. In just three to ten days a limited number of individuals learn more in a single workshop than in weeks of self-study. For optimum convenience, workshops are scheduled regularly at Training Centers worldwide or we can take our workshops to you for on-site instruction. Covering a wide variety of topics, Intel's major course categories include: architecture and assembly language, programming and operating systems, BITBUS™ and LAN applications.

NETWORK MANAGEMENT SERVICES

Today's networking products are powerful and extremely flexible. The return they can provide on your investment via increased productivity and reduced costs can be very substantial.

Intel offers complete network support, from definition of your network's physical and functional design, to implementation, installation and maintenance. Whether installing your first network or adding to an existing one, Intel's Networking Specialists can optimize network performance for you.

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Alphanumeric Index

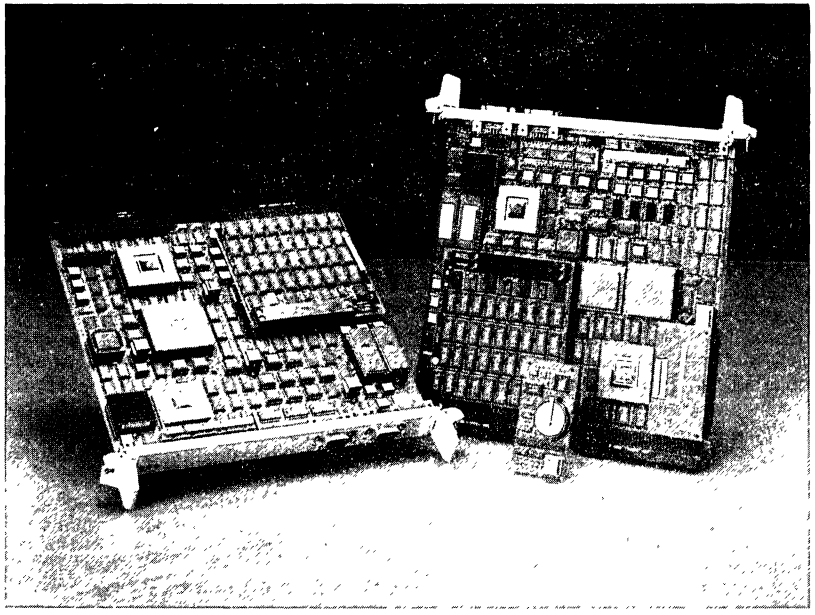
Alphanumeric Index (Continued)

Alphanumeric Index (Continued)

MULTIBUS® II
Single Board Computers

1

iSBC®486/125DU AND 386/133* SINGLE BOARD COMPUTERS



ADDING i486™ AND 386™ CPU POWER TO MULTIBUS® II

The iSBC® 486/125DU (Development Unit) and iSBC 386/133* Single Board Computers provide immediate access to the i486™ and 386™ 32-bit microprocessors on MULTIBUS II, the industry standard multiprocessing system bus. The iSBC 486/125DU board enables the system architect to prototype and validate an i486 processor-based system today. Boasting the 33MHz 386 processor and a feature set compatible with its predecessors, the iSBC 386/133 board offers an immediate performance boost for current MULTIBUS II designs. The iSBC 486/125DU and iSBC 386/133 boards are also fully compatible. Design and ship systems in volume today using iSBC 386/133 boards, and easily upgrade them to i486 CPU performance later.

FEATURES

iSBC® 486/125DU

- i486™ CPU operating at 25MHz with on-chip FPU and cache
- 8MB on-board DRAM with parity

iSBC® 386/133

- 386™ CPU operating at 33MHz
- 387™ Numeric Coprocessor
- 64KB SRAM zero wait-state cache
- 1-16MB on-board DRAM with parity

iSBC® 486/125DU AND iSBC® 386/133

- 82258 ADMA with 16-byte "blast" mode
- Two 32-pin JEDEC EPROM Sites with Built-in-self-test (BIST)
- iLBX™ II interface
- Two RS-232 asynchronous serial ports
- One iSBX™ connector
- 3 programmable interval timers, 15 levels of interrupt
- Full 32-bit MULTIBUS II (IEEE/ANSI 1296) Parallel System Bus interface
- Connector for on-board CSM option
- Full Operating Systems Support: iRMX® II and UNIX** System V/386 operating systems, and iRMK™ I real-time kernel

intel

* The iSBC 386/133 board is also manufactured under product code pSBC386/133 by Intel Puerto Rico, Inc

September, 1989
Order Number 281007-001

386™ MICROPROCESSOR SPEED AND PERFORMANCE

The iSBC 386/133 Single Board Computer features the highest speed 386™ microprocessor available today—33MHz. It also includes a 387™ floating-point coprocessor running at 33MHz. The 121-pin, extended math coprocessor socket could also house a WEITEK 3167 floating-point unit (FPU) instead of the 387 FPU.

HIGH-SPEED ON-BOARD MEMORY

For optimum CPU performance, the iSBC 386/133 board includes a 64K Byte cache memory resulting in zero wait-state read accesses on cache hits. The iSBC 386/133 board provides 1-16MB of parity protected, fast-page DRAM memory. The memory interface is designed to support up to 64M bytes, allowing for further memory expansion when 4 Mbit DRAM modules become available.

This board also includes two 32-pin JEDEC sites for up to 512KB of EPROM using 27020, 2 Mbit EPROM devices. These sites, as shipped, contain BIST (Built-In-Self-Test) and IDX (Initialization and Diagnostics eXecutive) power-up diagnostics residing in two pre-programmed 27010 EPROMs.

82258 ADMA COPROCESSOR WITH 16-BYTE "BLAST" MODE

The 82258 Advanced DMA coprocessor provides 4 DMA channels. Two channels are allocated to data transfers to/from the iSBX™ bus interface. The remaining two channels handle data transfers between the MPC and on-board memory. Special logic on the board allows the 82258 to transfer data to and from the MPC 4-bytes ("fly-by") or 16-bytes ("blast" mode) at a time; or at a sustained rate of 13.3M bytes/sec or 20.0M bytes/sec.

BALANCED SET OF ON-BOARD I/O: TWO SERIAL PORTS, iSBX™ CONNECTOR

Through extensive use of surface mount technology, the iSBC 386/133 board has increased the on-board I/O features over previous MULTIBUS II CPU boards. It provides two serial ports based on Intel's 82530 Serial Communications Controller and one iSBX connector, capable of supporting a single- or double-wide, 8- or 16-bit iSBX MULTIMODULE™ board.

iLBX™ II INTERFACE FOR MEMORY MAPPED I/O EXPANSION

The iLBX™ II interface on P2 provides expansion for 64M Bytes of off-board memory or memory mapped I/O. It operates at 8MHz and is completely compatible with the iLBX II interface on the iSBC 286/100A and the iSBC MEM 3xx memory modules.

ON-BOARD CSM (CENTRAL SERVICES MODULE) CAPABILITY

An iSBC CSM/002 connector on the iSBC 386/133 provides an on-board CSM option. The iSBC CSM/002 module performs all CSM functions required by the IEEE/ANSI 1296 Specification. It also provides a battery-backed time-of-day clock, periodic alarm function, and 28 bytes of non-volatile RAM. The iSBC 386/133 with CSM module is installed in slot zero of a MULTIBUS II chassis and requires only one slot.

FULLY COMPATIBLE WITH iSBC® 386/116,120 AND iSBC® 486/125DU

The iSBC 386/133 board is fully compatible with the INTEL386 family of MULTIBUS II boards, namely the iSBC 386/116, 386/120 and 486/125DU. Table 1 shows that the iSBC 386/133 feature set, with the exception of PSB access to on-board memory, is a super set of the iSBC 386/116, 120 feature set and matches that of the iSBC 486/125DU. Your 16MHz and 20MHz designs may be upgraded to 33MHz for an immediate system performance boost with a simple board swap.

PLUG-AND-PLAY WITH SYSTEM 520

Conforming to the MULTIBUS II Systems Architecture (MSA), this board integrates cleanly into the System 520. All you need is the "System Integration Toolkit" (SIT kit) that contains all the firmware necessary to operate in the System 520 enviroment. Install the firmware, plug the iSBC 386/133 into an empty slot, and start the system. It's that simple.

COMPREHENSIVE DEVELOPMENT AND OPERATING SYSTEM SUPPORT

Operating system support includes the iRMX® II Real-Time operating system and UNIX System V/386 operating system. The iRMK™ I real-time kernel is available for 32-bit embedded applications. All three include MULTIBUS II transport for full message passing support. To ease MULTIBUS II modules development, Intel offers both the iRMX and UNIX operating system versions of the System 520 Development System which can support on-target and/or cross-hosted software development in one chassis.

iSBC® 486/125DU FEATURES

DEVELOPMENT VEHICLE FOR FAST TIME TO MARKET WITH i486™ CPU

The iSBC 486/125DU Development Unit provides the system architect the opportunity to start designing today an i486™ processor-based system using an industry standard, off-the-shelf board. These units are currently available in limited sample quantities.

i486™ MICROPROCESSOR: THE HIGHEST PERFORMANCE COMPATIBILITY PROCESSOR

The heart of the iSBC 486/125DU board is the i486™ microprocessor, the newest and fastest member of the popular INTEL386™ 32-bit processor family. It is binary compatible with 386 microprocessors and offers two to three times the performance. The i486 microprocessor provides the highest level of performance through a state-of-the-art design containing a pipelined architecture, 8KB cache, and a high-performance local bus interface. Frequent instructions execute in one cycle. The performance is further enhanced by an on-chip floating-point unit (FPU) that is binary compatible with the 387 numerics coprocessor.

HIGH-SPEED MEMORY INTERFACE OPTIMIZES i486™ CPU PERFORMANCE

The iSBC 486/125DU board comes with 8 MBytes of byte-parity protected DRAM. On-board memory access is optimized via a two way interleaved memory design using fast page DRAMs. The iSBC 486/125DU board fits in a single MULTIBUS II slot.

COMPLETELY COMPATIBLE WITH THE iSBC® 386/133

As shown in Table 1, all the I/O and MULTIBUS II architecture support features of the iSBC 386/133 are also provided on the iSBC 486/125DU for full compatibility with the iSBC 386/133. Both boards are supported by the iRMX, iRMK and UNIX operating systems.

TABLE 1

INTEL386™ architecture based MULTIBUS II CPU Boards' Feature Set Comparison:

	CPU/Speed (MHz)	FPU/Speed	Cache	Max DRAM (MByte)	Two EPROM Sockets	ADMA	Serial Ports	iSBX™ Connector	iLBX™ II Interface	BIST and IDX	CSM Option	Local Access to PSB Memory & I/O	PSB Access to On-board Memory	iRMX®II, iRMK™ I, UNIX* support
iSBC® 486/125DU	486™/25	On chip	On chip	8	✓	✓	2	✓	✓	✓	✓	✓		✓
iSBC® 386/133	386™/33	387™/33	64KB	16	✓	✓	2	✓	✓	✓	✓	✓		✓
iSBC® 386/120	386™/20	387™/20	64KB	16	✓	✓	1(8751)	✓		✓		✓	✓	✓
iSBC® 386/116	386™/16	387™/16	64KB	16	✓	✓	1(8751)	✓		✓		✓	✓	✓

WORLD WIDE SERVICE AND SUPPORT

Should these or any Intel board ever need service, Intel maintains a world wide network of service and repair facilities to keep you and your customers up and running. For unique applications requiring customization of our products, the Intel Custom Board and Systems Group is available to modify, integrate and test Intel boards and system components to your requirements.

INTEL QUALITY—YOUR GUARANTEE

The iSBC 386/133 and iSBC 486/125DU boards are designed and manufactured to meet Intel's strict standards, assuring their reliability and high quality.

SPECIFICATIONS

CPU CLOCK RATE

iSBC 386/133:	386™ CPU @ 33.3MHz 387™ FPU @ 33.3MHz
iSBC 486/125DU:	i486™ CPU @ 25MHz

CACHE MEMORY

iSBC 386/133:	64K bytes 0 wait state on read hit
iSBC 486/125DU:	8K bytes on-chip 0 wait state on read hit

DMA CLOCK RATES/MAXIMUM BANDWIDTH*

82258 ADMA 10MHz/20 MB per second

*assumes transfer between local DRAM and MPC.

ON-BOARD DRAM MEMORY

Model	Supplied
iSBC 486/125 DU	8MB
iSBC 386/133 F01	1MB
iSBC 386/133 F02	2MB
iSBC 386/133 F04	4MB
iSBC 386/133 F08	8MB

- Single-bit parity error detection per byte
- iSBC 386/133 only: Memory expansion possible with one additional iSBC MM0xFP module. Maximum on-board memory capacity = 16MB (64MB w/ 4Mbit DRAMs)

NOTE: Model suffixes F02 and F08 require two MULTIBUS II card slots.
iSBC 486/125DU with 8MB fits in one MULTIBUS II slot.

EPROM MEMORY

Two JEDEC sites provide following capacity:

Devices	Capacity
27010	256K byte (supplied)
27020	512K byte

INTERFACES

- MULTIBUS II PSB: 32-bit Parallel System Bus (ANSI/IEEE 1296) interface with full message passing capability
- iSBX Bus: Compliance Level: D16/16 DMA
- iLBX II Bus: Compliance Level: PRQA
- Serial I/O: RS232C DTE ASYNC

SERIAL I/O PORT

- Channel A & B: RS232C compatible DTE Asynchronous interface
- 9-pin D-shell shielded connector
- Configurable baud rates: 300,600,1200,2400,4800,9600,19200, and 38400

PHYSICAL DIMENSIONS

Height:	233 mm (9.18 inches)
Depth:	220 mm (8.65 inches)
Front Panel Width:	19.2 mm (0.76 inches)

ELECTRICAL CHARACTERISTICS

DC Power Requirements (Typical): +5V, 13A
±12V, 200mA

NOTE: Does not include power for iSBX module, or added iSBC MM0x modules.

For the second iSBC MM0x module, add:
iSBC MM01-FP or iSBC MM04-FP +5V, 0.71A
iSBC MM02-FP or iSBC MM08-FP +5V, 0.96A

REFERENCE MANUAL

iSBC 386/133 Single Board Computer User's Guide (order number 457629-001)

iSBC 486/125DU Single Board Computer User's Guide (order number 459600-001)

ORDERING INFORMATION

Order Code	Description
SBC486125DU	25MHz 486 CPU-based Development Unit w/ 8MB DRAM
SBC386133F01	33MHz 386 CPU board w/ 1MB DRAM
SBC386133F02	33MHz 386 CPU board w/ 2MB DRAM
SBC386133F04	33MHz 386 CPU board w/ 4MB DRAM
SBC386133F08	33MHz 386 CPU board w/ 8MB DRAM
SBCMM01FP	1MB, 85ns memory expansion module
SBCMM02FP	2MB, 85ns memory expansion module
SBCMM04FP	4MB, 85ns memory expansion module
SBCMM08FP	8MB, 85ns memory expansion module
SBCCSM002	CSM Option Module
SIT133KIT	System 520 firmware for iSBC 386/133

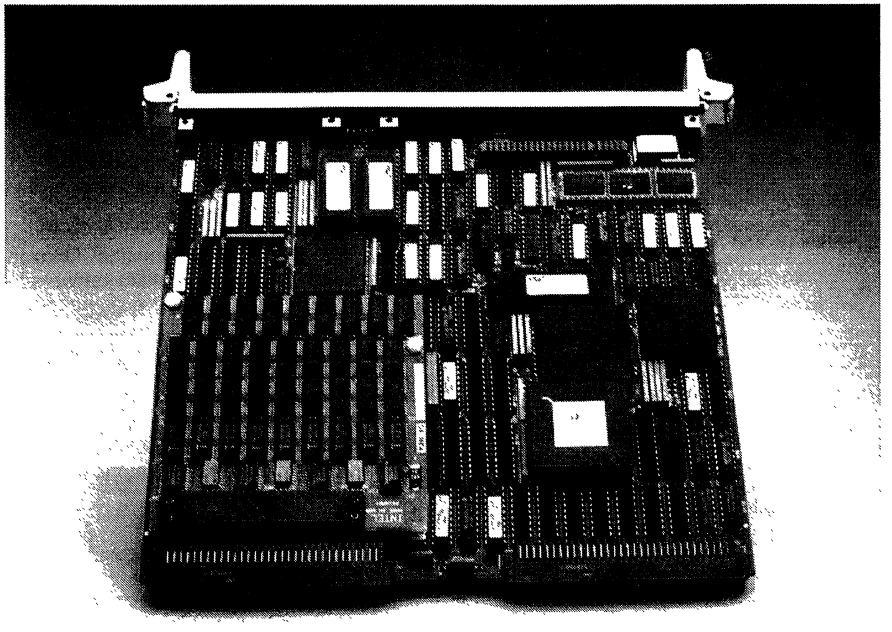
For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada).

iSBC® 386/116 AND 386/120* MULTIBUS® II SINGLE BOARD COMPUTERS

- **High Performance 32-bit 386™ Processor Operating at 16 MHz or 20 MHz**
- **80387 Numerics Co-Processor Providing IEEE 754 Floating Point Instruction Set, Operating at 16 MHz or 20 MHz**
- **64K byte Static RAM Cache Providing Zero Wait State Reads**
- **1, 2, 4 or 8M Bytes of On-Board Dual-Ported Dynamic RAM Memory with Parity Error Detection, Expandable to 16M Bytes**
- **One RS 232C Serial I/O Port**
- **82258 DMA Controller Providing 4 High Performance DMA Channels**
- **32-Bit MULTIBUS®II Parallel System Bus (IEEE 1296) Interface with Full Message Passing Capability**
- **8-, 16-Bit iSBX™ Bus (IEEE P959) Interface with DMA for I/O Expansion**
- **Resident Firmware to Support Built-In-Self-Test (BIST) Power-Up Diagnostics**
- **MULTIBUS II Interconnect Space for Software Configurability and Diagnostics**

The iSBC 386/116 and 120 MULTIBUS II Single Board Computers are based on Intel's 386 high performance 32-bit microprocessor. The 386 CPU maintains software compatibility with the entire 8086 microprocessor family and delivers new performance standards for microcomputer-based systems. Four versions of the iSBC 386/116 and 120 boards are offered: the M01, which contains 1M byte of DRAM; the M02, which contains 2M bytes of DRAM; the M04, which includes 4M bytes of DRAM; and the M08 which contains 8M bytes of DRAM. An optional memory expansion module can be added to expand the iSBC 386/116 or 120 board's resident memory to a maximum of 16M bytes.

The 64K byte static RAM cache enables the 386 CPU to execute at its full potential performance, while the MULTIBUS II bus provides an interface for reliable, high performance multiprocessing.



280631-1

The iSBC® 386/116 and iSBC® 386/120 are also manufactured under product code piSBC® 386/116 and iSBC® 386/120 by Intel Puerto Rico, Inc.

November 1989
Order Number: 280631-002

FUNCTIONAL DESCRIPTION

Overview

The iSBC 386/116 and 120 boards utilize Intel's 386 32-bit microprocessor. The advanced capabilities of the MULTIBUS II architecture coupled with the high performance and compatibility features of the 386 CPU, provide the designer with a superior 32-bit solution for multiprocessing applications. By using the MULTIBUS II architecture, multiprocessing systems are enhanced through advanced bus features including: 21-board distributed arbitration, virtual interrupts, hardware-assisted message passing, bus parity for high reliability, and software configurability using interconnect address space. The MULTIBUS II parallel system bus (iPSB) interface on the iSBC 386/116 and 120 boards support full message passing and dual-port architectures and is fully compatible with other SBCs based on the MULTIBUS II (IEEE 1296) bus specification.

The iSBC 386/116 and 120 boards are offered in four versions: M01, M02, M04 and M08 which contain 1, 2, 4 and 8M bytes of resident DRAM memory respectively. This memory is physically located on an expansion board, and can be accessed directly from the iSBC 386/116 or 120 board's local bus or by another CPU over the iPSB bus. This dual-port memory can be expanded to a maximum of 16M bytes though the addition of a second Intel iSBC MM01, MM02, MM04 or MM08 (1, 2, 4 or 8M byte) memory expansion module. Parity error detection is included on all resident DRAM memory.

Architecture

The iSBC 386/116 and 120 logic consists of eight resource modules and three interfaces connected together over an on-board local bus. The resources include the 386 CPU, the 80387 numeric co-processor, the 82258 DMA controller, the dual-port DRAM memory, the SRAM cache memory, the EPROM memory with BIST software, the programmable timers and the interrupt controllers. Interfaces included are the iPSB parallel system bus, the iSBX I/O bus and the RS 232C serial I/O interface. A block diagram of the iSBC 386/116, 120 board is shown in Figure 1. The following text describes each of the resources and interfaces.

386™ PROCESSOR

Intel's 386 CPU is the central processor for the iSBC 386/116 and 120 boards. This is the first 32-bit member of Intel's 8086 family of microprocessors. At 16 MHz and 20 MHz, the 386 is capable of executing at sustained rates of 4 and 5 million 32-bit instructions per second, respectively. This performance is made possible through a state-of-the-art design combining advanced VLSI semiconductor technology, a pipelined architecture, address translation caches and a high performance local bus interface.

The 386 processor provides a rich, generalized register and instruction set for manipulating 32-bit data and addresses. Features such as scaled indexing and a 64-bit barrel shifter ensure the efficient addressing and fast instruction processing. Special emphasis has been placed on providing optimized instructions for high-level languages and operating system functions. Advanced functions, such as hardware-supported multitasking and virtual memory support, provide the foundation necessary to build the most sophisticated multitasking and multiuser systems. Many operating system functions have been placed in hardware to enhance execution speed. The integrated memory management and protection mechanism translates virtual addresses to physical addresses and enforces the protection rules necessary for maintaining task integrity in a multiprocessing environment.

The 386 CPU provides access to the large base of software developed for the 8086 family of microprocessors. Binary code compatibility allows execution of existing 16-bit applications without recompilation or reassembly, directly in a virtual 8086 environment. Programs and even entire operating systems written for 8086 family processors can be run as tasks under 32-bit operating systems written for the 386 CPU.

80287 NUMERIC CO-PROCESSOR

The 80387 is a high-performance floating-point co-processor that takes numerics functions which would normally be performed in software by the 386 microprocessor and instead executes them in hardware. The instruction set executed by the 80387 is compatible with the IEEE 754 floating point standard, with high-precision 80-bit architectures and full support for single, double and extended precision operations. The 80387 executes floating point operations at a rate of 1.5M Whetstones per second at 16 MHz, and 1.86M Whetstones per second at 20 MHz.

82258 ADVANCED DMA CO-PROCESSOR

The 82258 is a high performance 4 channel DMA co-processor. Unlike other DMA devices, the 82258 has processing capabilities. Its command chaining feature and data manipulation capabilities (compare, verify, translate), allow the 82258 to execute simple

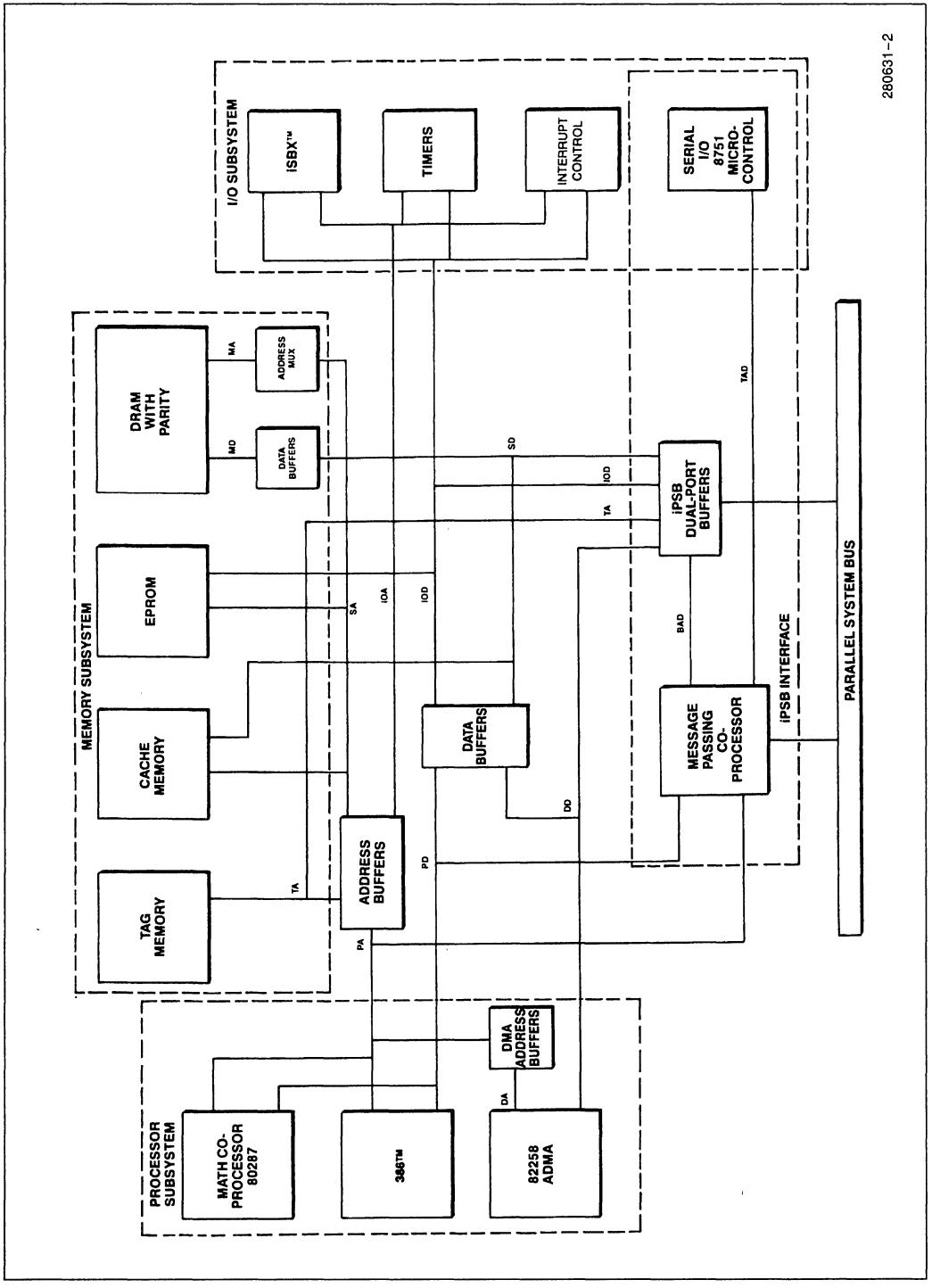


Figure 1. Block Diagram of the iSBC® 386/116, 386/120 Board

input/output programs without processor intervention. This allows the 386 CPU more time for general purpose processing, thus improving total system performance. The 82258 shares ownership of the on-board local bus via the 386 processor's HOLD, HOLD ACKNOWLEDGE co-processor protocol. The maximum percentage of on-board local bus utilization by the 82258 is user programmable.

The four 82258 channels are allocated to iSBC 386/116 or 120 on-board resources as shown in Table 1. Special logic on the boards allows the 82258 to transfer data to and from the message passing co-processor (MPC) 32-bits at a time using single cycle mode. Using this mode, the 82258 (which operates at 8 MHz on 386/116 and 10 MHz on 386/120) can load or unload an MPC solicited message (from or to resident DRAM) at a sustained rate of 10.7M bytes and 13.3 Mbytes per second, respectively.

Table 1. DMA Channel Allocation

Channel	Function
0	iSBX DMA support
1	iSBX DMA support
2	MPC Solicited Message Receive
3	MPC Solicited Message Transmit

DUAL-PORTED DYNAMIC RAM

The iSBC 386/116 and 120 boards include 1, 2, 4 or 8M bytes of DRAM depending upon the version. This memory can be extended to a maximum of 16M bytes through the addition of an Intel iSBC MM01, MM02, MM04 or MM08: 1, 2, 4 or 8M byte memory expansion module. The DRAM refresh control, dual-port control and parity generation/checking logic is physically located on the baseboard, while the actual DRAM components are located on low-profile surface mount expansion boards. Each iSBC 386/116 or 120 board is shipped with one expansion memory module installed and may be expanded to contain two total memory expansion modules. The memory expansion module mechanics are shown in Figure 2.*

***NOTE:**
Only one single-sided memory module (MM01 or MM04) installed onto the iSBC 386/116 or 120 board will fit within one MULTIBUS II slot. A double-sided module (MM02 or MM08) or any stack of two modules will require two MULTIBUS II slots.

Parity error detection is provided on a byte-by-byte basis. The parity logic normally generates and checks for odd parity with detected errors signaled via an on-board LED and a CPU interrupt. Even parity can be forced to generate a parity error for diagnostic purposes.

The DRAM is accessible from both the on-board local bus and the iPSB bus. The amount of memory accessible from the iPSB bus and the iPSB address aliasing values are dynamically configurable via interconnect space registers.

CACHE MEMORY

The cache memory on the iSBC 386/116 and 120 boards allow zero wait-state accesses to memory when the data requested is resident in the cache memory. The static RAM cache has 16,384 32-bit data entries with 8-bit "tag" fields. Each 32-bit DRAM memory location maps to one (and only one) cache data entry. The "tag" fields are used to determine which 32-bits of DRAM memory currently resides in each cache data entry. The combination of a direct mapped cache data array and a tag field ensures data integrity and accurate, high performance identification of cache "hits".

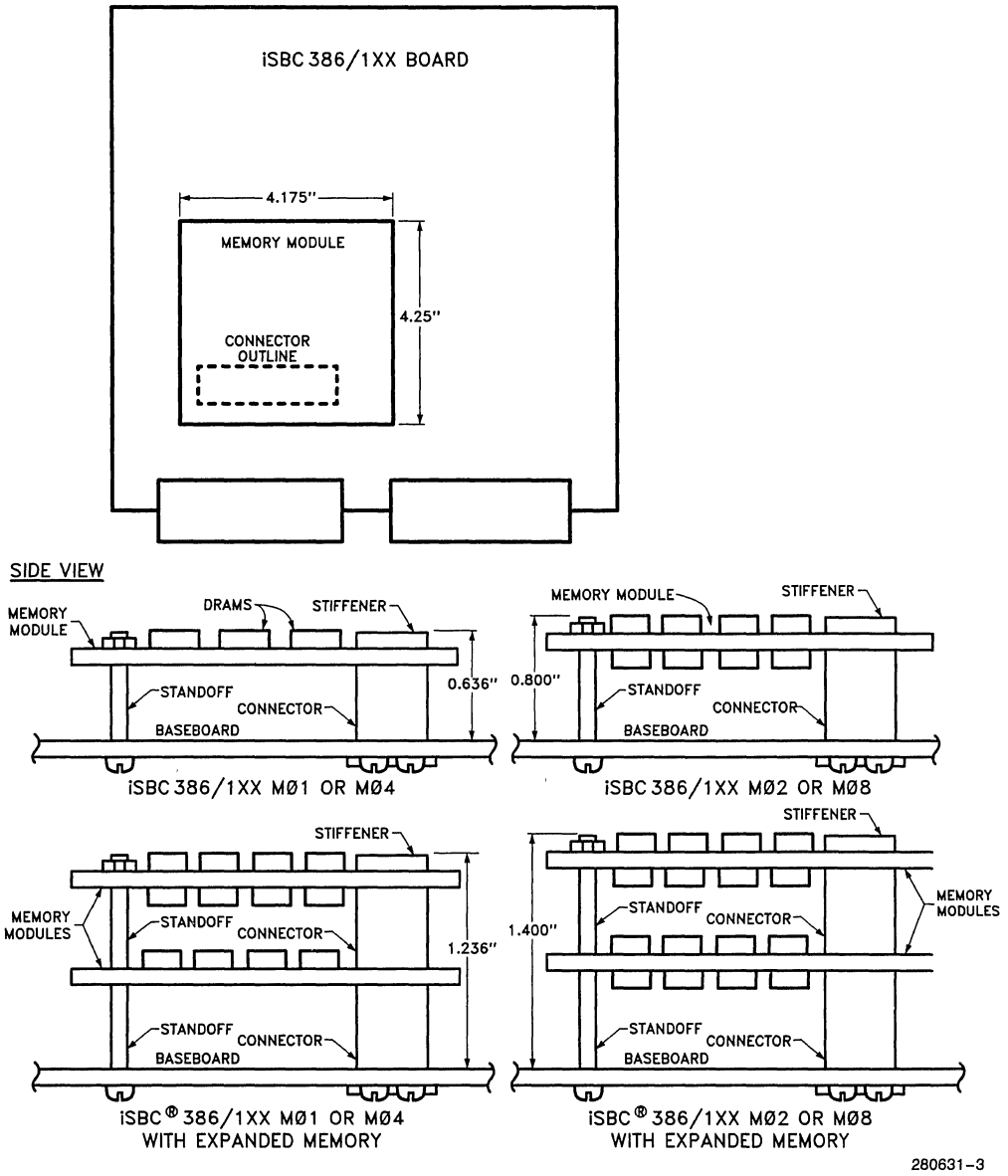
Data integrity is maintained for cache "misses" (DRAM memory READs not in the cache) and DRAM memory WRITEs through a simple, yet effective replacement algorithm. 386 CPU generated cache READ "misses" cause the data field of the cache entry corresponding to the addressed memory to be filled from the DRAM array and the tag field to be updated. All iPSB or ADMA READs are treated as cache "misses", except that the cache is not updated. All WRITE "hits", local and iPSB generated, cause the cache data field to be updated. WRITE "misses" do not update the cache. The cache memory size and replacement algorithm are designed to optimize both the probability of cache "hits" and local bus utilization.

EPROM MEMORY

Two 32-pin JEDEC EPROM sites capable of supporting up to 512K bytes of EPROM (using 27020 EPROMs) are supplied on the iSBC 386/116 and 120 boards. These sites, as shipped, contain built-in-self-test power-up diagnostics residing in two preprogrammed 27512 EPROMs. These EPROMs may be replaced by the user. Jumper configurations allow the use of 2764, 27128, 27256, 27512, 27010, and 27020 EPROMs.

8254 PROGRAMMABLE TIMERS

The iSBC 386/116 and 120 boards contain an Intel 8254 component which provides three independent programmable 16-bit interval timers. These may be used for real-time interrupts or time keeping operations. Outputs from these timers are routed to one of the two 8259A interrupt controllers to provide software programmable real-time interrupts.



NOTE:
1. The iSBC 386/1XX M01 or M04 fits within one MULTIBUS II slot. The iSBC 386/1XX M02 or M08, or any iSBC 386/1XX with a stack of two memory modules will require two MULTIBUS II slots.

Figure 2. iSBC® 386/116 and 120 Boards Memory Module Mechanics

INTERRUPT CONTROL

Two Intel 8259A programmable interrupt controllers on the iSBC 386/116 and 120 boards are used in a master-slave configuration for prioritizing up to 15 separate on-board interrupt sources. The devices and functions are listed in Table 2.

The MULTIBUS II iPSB bus utilizes virtual interrupts (called unsolicited messages) for board-to-board signaling. The bus interface component (MPC) queues-up incoming virtual interrupts from the iPSB bus and generates a single message interrupt (MINT) signal. This signal is connected into one of the 8259A interrupt controllers for prioritization and interruption of the host 386 CPU. Error conditions occurring on the iPSB bus will cause the MPC to generate an error interrupt (EINT) signal. This signal is connected to another 8259A interrupt input.

Other interrupt sources come from the 82258 DMA controller, the 8254 timers, the iSBX interface, the 8751 serial port, and the DRAM parity checker.

SERIAL I/O INTERFACE

One RS 232C compatible serial I/O port is provided via the Intel 8751 microcontroller. This port is configured as a data terminal equipment (DTE) asynchronous serial port. Mechanically, the serial port exits through the iSBC 386/116 or 120 board's front panel via a 9-pin D-shell connector.

iPSB BUS INTERFACE

The MULTIBUS II parallel system bus interface is implemented by Intel's MPC (message passing coprocessor) and a pre-programmed 8751 microcontroller. This interface supports full arbitration, transfer and error checking features as defined in the iPSB specifications. In addition, the interface supports advanced features of the iPSB bus including hardware message passing and autoconfiguration through geographic addressing.

The MPC component contains nine 32-byte buffers which are used to decouple iPSB bus traffic from iSBC 386/116 or 120 local bus traffic through the concept known as message passing. These nine buffers are utilized as follows: four buffers queue-up incoming unsolicited messages, one buffer stores an out-going unsolicited message, two buffers are used to double-buffer an out-going solicited message, and two buffers are used to double-buffer an incoming solicited message. These buffers are capable of transferring data packets over the iPSB bus at its maximum transfer rate. Unsolicited messages include address and type fields and 28 bytes of user-defined data, and are transferred over the iPSB bus in 900 ns. Solicited messages are automatically divided into small packets, with each packet containing address and type fields and 32 bytes of user-defined data. Each solicited message packet is transferred over the iPSB bus in 1000 ns.

Table 2. 8259A Interrupt Sources

Device	Function	Number of Interrupts
MPC-MINT	Signals arrival of virtual interrupt over iPSB bus, solicited input complete, transmit FIFO not full or transmit error	1
MPC-EINT	Signals error condition on the iPSB bus	1
82258 DMA	Transfer complete	1
8254 Timers	Timers 0, 1, 2 outputs, function determined by timer mode	3
8751 Serial Port	Serial diagnostic port requests	1
iSBX Interface	Function determined by iSBX bus multimodule board	4
DRAM Parity Checker	Signals parity error	1

The 8751 component implements the iPSB geographic addressing feature called Interconnect space. Read-only registers are used to hold information such as board type and revision level. Software configurable registers are used for auto-configurability, local or remote diagnostics and software controlled reset. In addition, the 386 CPU executes power-up built-in self tests of the various resources on the iSBC 386/116 and 120 boards. The results of these tests are reported via registers in interconnect space. After successfully completing its BIST routines, the 386 CPU must clear the reset-not-complete register. If, after 30 seconds, the reset-not-complete has not been cleared, the 8751 resets the local bus and holds it in a reset state. In this way, only a few components on the iSBC 386/116 or 120 board must be functional to allow the iPSB bus to operate.

iSBX™ BUS INTERFACE

One iSBX connector, capable of supporting one single- or double-wide, 8- or 16-bit iSBX MULTIMODULE board, is provided on the iSBC 386/116 and 120 boards for the addition of an optional I/O module. Two DMA channels from the 82258 can be used with iSBX modules which require DMA support.

SPECIFICATIONS

Word Size

Instruction — 8-, 16-, 24-, 32-, 40-bit
Data — 8-, 16-, 32-bit
Floating Point Data — 80-bit

Clock Rates

	386/116	386/120
386™ CPU	16 MHz	20 MHz
80387 Numeric Co-processor	16 MHz	20 MHz
82258 DMA	8 MHz	10 MHz

Dual-Port DRAM Memory

DEFAULT CAPACITY

iSBC 386/116 M01—1M byte
iSBC 386/116 M02—2M byte
iSBC 386/116 M04—4M byte
iSBC 386/116 M08—8M byte

iSBC 386/120 M01—1M byte
iSBC 386/120 M02—2M byte
iSBC 386/120 M04—4M byte
iSBC 386/120 M08—8M byte

EXPANSION MODULES

iSBC MM01—1M byte
iSBC MM02—2M byte
iSBC MM04—4M byte
iSBC MM08—8M byte

MAXIMUM CAPACITY—16M BYTES

EPROM Memory

Default — 128K byte using two pre-programmed 27512 EPROMs
Capacity — Two 24-, 28- or 32-pin JEDEC-compatible devices

EPROM	Memory Capacity
2764	16 KB
27128	32 KB
27256	64 KB
27512	128 KB
27010	256 KB
27020	512 KB

Timers

Capability — Three independently programmed 16-bit interval timers
Input Frequency — 1.25 MHz ±0.1%
Output Period — 1.6 μs to 52.4 ms

Interrupt Capability

Incoming Interrupts — 255 individual and 1 broadcast from iPSB bus 12 local sources (see Table 2)
Outgoing Interrupts — 255 individual and 1 broadcast to IPSB bus

Serial Port Interface

RS 232C Electrical
Asynchronous, DTE only
9-pin D-shell connector
Baud rates: 9600, 4800, 2400, 1200, 300, 110 bits/sec

iSBX Interface

Capability — One 8- or 16-bit, single- or double-wide iSBX module
Compliance Code — D16/16 DMA

iPSB Interface

Capability— Requesting and replying agent supporting 8-, 16-, 24- and 32-bit transfers, parity bit generation and checking, unsolicited and solicited message passing, and autoconfiguration through interconnect space.

Physical Dimensions

Length: 220 mm (8.6 in.)
Width: 233 mm (9.2 in.)
Front Panel Height: 19.2 mm (0.76 in.)

Power Requirements

5V: 11.14 Amps
12V: 0.046 Amps
−12V: 0.041 Amps
Voltage tolerance ±5%

Temperature Range and Airflow Requirements

Storage Temperature: −40°C to +70°C
Operating Temperature: 0°C to +55°C
Airflow: 200 LFM minimum

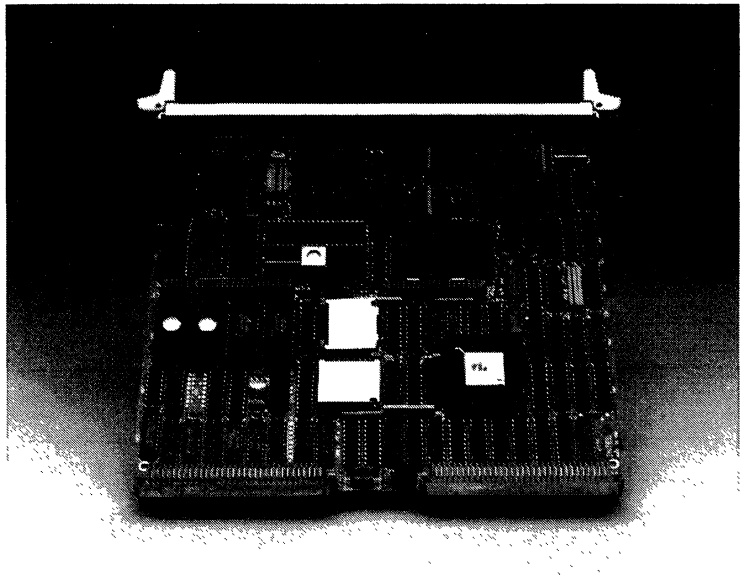
ORDERING INFORMATION

Part Number	Description
SBC386116M01	16 MHz 386 CPU-based MULTIBUS II CPU board with 1M byte memory
SBC386116M02	16 MHz 386 CPU-based MULTIBUS II CPU board with 2M byte memory
SBC386116M04	16 MHz 386 CPU-based MULTIBUS II CPU board with 4M byte memory
SBC386116M08	16 MHz 386 CPU-based MULTIBUS II CPU board with 8M byte memory
SBC386120M01	20 MHz 386 CPU-based MULTIBUS II CPU board with 1M byte memory
SBC386120M02	20 MHz 386 CPU-based MULTIBUS II CPU board with 2M byte memory
SBC386120M04	20 MHz 386 CPU-based MULTIBUS II CPU board with 4M byte memory
SBC386120M08	20 MHz 386 CPU-based MULTIBUS II CPU board with 8M byte memory
SBCMM01	1M byte memory expansion module
SBCMM02	2M byte memory expansion module
SBCMM04	4M byte memory expansion module
SBCMM08	8M byte memory expansion module
451833	iSBC 386/116 and 386/120 Single Board Computer Users Guide

iSBC® 286/100A* MULTIBUS®II SINGLE BOARD COMPUTER

- **8 MHz 80286 Microprocessor with Optional 80287 Numeric Data Co-Processor**
- **MULTIBUS® II PSB (Parallel System Bus) Interface with Full Message Passing Capabilities and up to 4 Gigabytes of Memory Addressability on the Bus**
- **High-Speed Memory Expansion with MULTIBUS II iLBX II (Local Bus Extension) Interface Addresses up to 16 MBytes of Local and/or Dual Port Memory**
- **Two iSBX Bus Interface Connectors for I/O Expansion Bus**
- **Four DMA Channels Supplied by the 82258 Advanced DMA Controller with 8 MBytes/sec Transfer Rate**
- **MULTIBUS® II Interconnect Space for Software Configurability and Self-Test Diagnostics**
- **Resident Firmware Supports Self-Test Power-Up Diagnostics and On-Command Extended Self-Test Diagnostics**
- **Two Programmable Serial Interfaces, one RS232C (DCE or DTE), the other RS232C or RE422A/RS449 Compatible**
- **Two 28-pin JEDEC Sites for up to 128 KBytes of Local Memory Using SRAM, NVRAM, EEPROM, and EPROM**
- **24 Programmable I/O Lines Configurable as SCSI Interface, Centronics Interface, or General Purpose I/O**

The iSBC 286/100A Single Board Computer is part of Intel's family of MULTIBUS II CPU boards that utilizes the advanced features of the MULTIBUS II System Architecture. It is ideally suited for a wide range of OEM applications. The combination of the 80286 CPU, the Message Passing Coprocessor (MPC), the MULTIBUS II Parallel System Bus (PSB bus), and the Local Bus Extension (iLBX II bus) makes the iSBC 286/100A board suited for high performance, multiprocessing system applications in a multimaster environment. The board is a complete microcomputer system on a 220mm x 233mm (8.7 x 9.2 inch) Eurocard form factor with pin and socket DIN connectors.



280076-1

*The iSBC® 286/100A is also manufactured under product code piSBC® 286/100A by Intel Puerto Rico, Inc.

October 1989
Order Number: 280076-005

Overview

The iSBC 286/100A Single Board Computer combines the 80286 microprocessor with the Message Passing Component (MPC) on a single board within the MULTIBUS II system architecture. This offers a message passing based high performance multiprocessing solution for system integrators and designers. Figure 1 shows a typical MULTIBUS II multiprocessing system configuration. Overall system performance is enhanced by the Local Bus Extension (iLBX II) which allows 0 wait state high speed memory execution.

Architecture

All features of the MULTIBUS II architecture are fully supported by the iSBC 286/100A board including the Parallel System Bus (PSB), interconnect space, Built-In-Self-Tests (BIST) diagnostics, and full message passing. These features are described in the following sections. In addition to taking advantage of the MULTIBUS II system architecture, the iSBC 286/100A board has complete single board computer capability including two iSBX bus expansion connectors, 80287 numeric data coprocessor option, advanced DMA control, JEDEC memory sites, SCSI configurable parallel interface, serial I/O, and programmable timers. Figure 2 shows the iSBC 286/100A board block diagram.

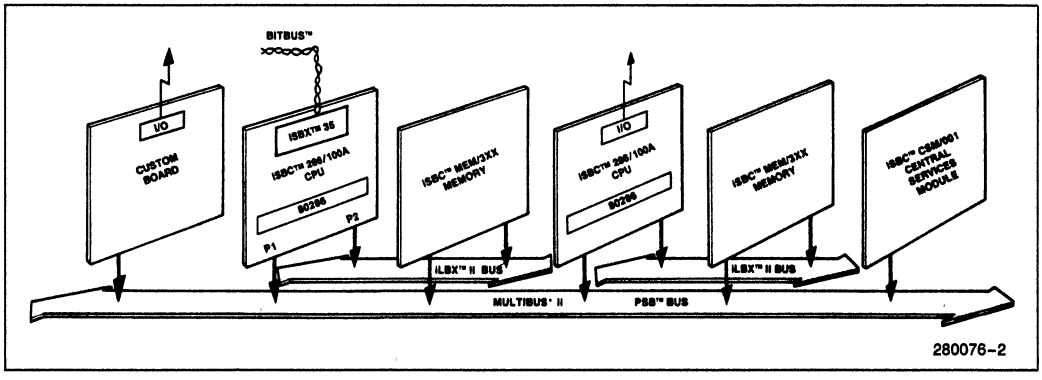


Figure 1. Typical MULTIBUS®II Multiprocessing System Configuration

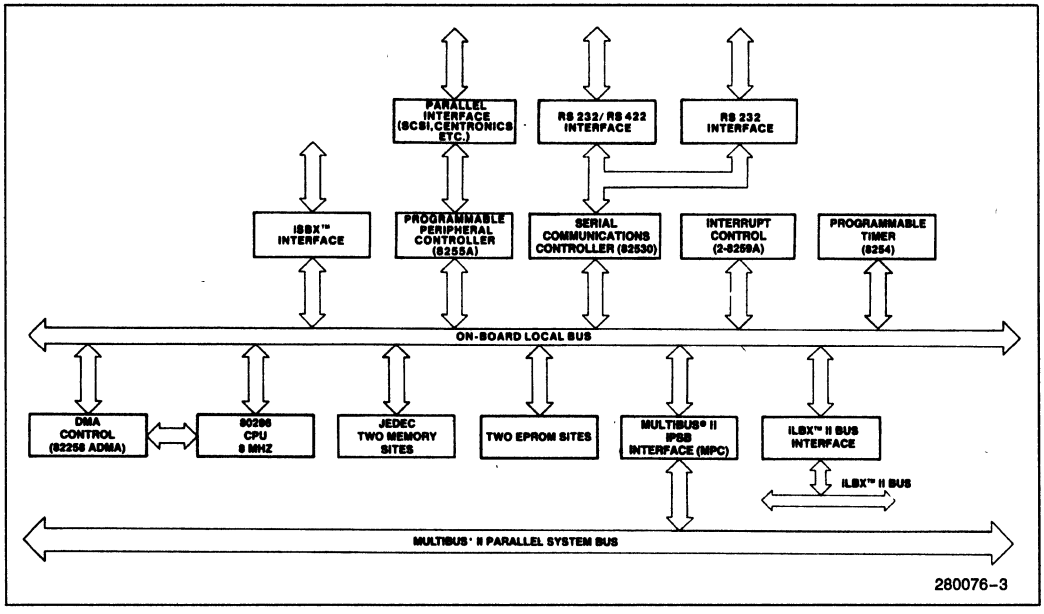


Figure 2. iSBC® 286/100A Board Block Diagram

Central Processing Unit

The central processing unit for the iSBC 286/100A board is the 80286 microprocessor operating at 8.0 MHz clock rate. The 80286 runs 8086 and 80186 code at substantially higher speeds (due to a parallel chip architecture) while maintaining software compatibility with Intel's 8086 and 80186 microprocessors. Numeric processing power may be enhanced with the 80287 numeric data coprocessor. The 80286 CPU operates in two modes: real address mode and protected virtual address mode. In real address mode, programs use real addressing with up to one megabyte of address space. In protected virtual address mode, the 80286 CPU automatically maps 1 gigabyte of virtual address per task into a 16 megabyte real address space. This mode also provides the hardware memory protection for the operating system. The operating mode is selected via CPU instructions.

PSB Bus Interface

The iSBC 286/100A board has a Message Passing Coprocessor (MPC) component on the base board that contains most of the logic required to operate the Parallel System Bus (PSB bus) interface. Some of the key functions provided by the MPC include bus arbitration, transfer control, parity generation and checking, and error detection and reporting.

Data transfers between processors via the PSB bus is defined in the MULTIBUS II architecture through a transfer protocol, a reserved address space, and an information/data block. This interprocessor communication convention is known as message passing. Operations occurring within the reserved address space are called message space operations.

Message passing allows PSB bus agents to transfer variable amounts of data at rates approaching the maximum bandwidth of the bus. Message passing permits a sustained transfer rate of 2.2 Mbytes per second, and a single message may transfer up to 16 Mbytes from one agent to another. The MPC fully supports message space operations, executes PSB bus arbitration and executes the message passing protocol independent of the host CPU, leaving the host free to process other tasks.

The MPC supports both solicited and unsolicited message passing capability across the PSB. An unsolicited message can be thought of as an intelligent interrupt from the perspective of the receiving agent because the arrival of an unsolicited message is unpredictable. Attached to an unsolicited message is one of 255 possible source addresses along with 28 bytes of data attached to the message data field. A solicited message moves large blocks of data between agents on the PSB bus. The arrival of a solicited message is negotiated between the sending and receiving agents. Data is sent in "packets" with each packet containing four bytes of control information and up to 28 bytes of data. There is no specific limit to the number of packets that may be sent in a single message, but the total message may not transfer more than 16 Mbytes.

The iSBC 286/100A also includes a feature called the PSB window register that allows the user to selectively access under software control any 256K byte block of memory within the 4 Gigabytes of memory space on the PSB bus interface.

INTERCONNECT SPACE SUPPORT

Interconnect space is one of four MULTIBUS II address spaces, the other three being memory space, I/O space, and message space. Interconnect space allows software to initialize, identify, configure, and diagnose the boards in a MULTIBUS II system. The Interconnect template consists of 8-bit registers, organized into functional groups called records. There are three types of records, the header record, function records, and the End of Template record.

The header record provides board and vendor ID information, general status and control information, and diagnostic control. The function records allow the user to configure and/or read the iSBC 286/100A board's hardware configuration via software. The End of Template record identifies the end of the interconnect template.

BUILT IN SELF TEST (BIST) DIAGNOSTICS

MULTIBUS II's Built in Self Test (BIST) diagnostics improve the reliability and error reporting and recovery capability of MULTIBUS II boards. These confidence tests and diagnostics not only improve reliability but also reduce manufacturing and maintenance costs for the OEM user. The yellow BIST LED (LED 1) on the front panel provides a visual indication of the power-up diagnostics status.

Error Reporting and Recovery

The MULTIBUS II Parallel System Bus and the iLBX II bus provides bus transmission and bus parity error detection signals. Error information is logged in the MPC and a bus error interrupt is generated. Information on the error source for reporting or recovery purposes is available to software through the iSBC 286/100A board interconnect space registers.

INTERRUPT CONTROL

In a MULTIBUS II system, external interrupts (interrupts originating off the CPU board) are messages over the bus rather than signals on individual lines. Message based interrupts are handled by the MPC. Two on-board 8259A Programmable Interrupt Controllers (PICs) are used for processing on-board interrupts. One is used as the master and the other as the slave. Table 1 includes a list of devices and functions supported by interrupts.

iSBX® BUS MULTIMODULE™ ON-BOARD EXPANSION

Two iSBX bus MULTIMODULE connectors are provided, one 16- or 8-bit and the other 8-bit. Through these connectors additional on-board I/O functions may be added. The iSBX bus MULTIMODULE boards optimally support functions provided by VLSI peripheral components such as additional parallel and serial I/O, analog I/O, and graphics control. The iSBX bus connectors on the iSBC 286/100A board provides all signals necessary to interface to the local on-board bus including 16 data lines and DMA for maximum data transfer rates. MULTIMODULE boards designed with 8-bit data paths and using the 8-bit iSBX bus connectors are also supported. A broad range of iSBX bus MULTIMODULE options are available from Intel. Custom iSBX bus MULTIMODULE boards designed for MULTIBUS or proprietary bus systems are also supported provided the IEEE P959 iSBX bus specification is followed.

NUMERIC DATA CO-PROCESSOR

The 80287 Numeric Data Co-Processor can be installed on the iSBC 286/100A board by the user. The 80287 Numeric Data Co-Processor is connected to dedicated processor signal lines which are pulled to their inactive state when the 80287 Numeric Data Co-Processor is not installed. This enables the user to detect via software that the 80287 socket is occupied. The 80287 Numeric Data Co-Processor runs asynchronously to the 80286 clock. The 80287 Numeric Data Co-Processor operates at 8 MHz and is driven by the 8284A clock generator.

Table 1. Interrupt Devices and Functions

Device	Function	Number of Interrupts
MULTIBUS® II Interface	Message-based Interrupt Request from the PSB Bus via 84120 Message Interrupt Controller	1 Interrupt from up to 256 sources
8751 Interconnect Controller	BIST Control Functions	1
82530 Serial Controller	Transmit Buffer Empty, Receive Buffer Full and Channel Errors	1 Interrupt from 10 Sources
8254 Timers	Timers 0, 1, 2 Outputs; Function Determined by Timer Mode	3
8255A Parallel I/O	Parallel Port Control	2
iLBX II Bus Interface	Indicates iLBX™ II Bus Error Condition	3
PSB Bus Interface	Indicates Transmission Error on PSB Bus	1
iSBX Bus Connector	Function Determined by iSBX Bus MULTIMODULE Board	2
Edge Sense Out	Converts Edge Triggered Interrupt to a Level	1
Bus Error	Indicates Last PSB Bus Operation Encountered an Error	1
Power-Fail	External/Power-Fail Interrupts	1

DMA CONTROL

Four DMA (Direct Memory Access) channels are supplied on the iSBC 286/100A board by the 82258. The 82258 is an advanced DMA controller designed especially for the 16-bit 80286 microprocessor. It has four DMA channels which can transfer data at rates up to 8 Megabytes per second (8 MHz clock) in an 80286 system. The large bandwidth allows the user to handle very fast data transfer or a large number of concurrent peripherals.

MEMORY CAPABILITIES

The local memory of the iSBC 286/100A board consists of two groups of byte-wide sites. The first group of two sites are reserved for EPROM or ROM and are used for the BIST power-up diagnostic firmware. The second group of two sites support JEDEC standard 28-pin devices.

PARALLEL PERIPHERAL INTERFACE

The iSBC 286/100A board includes a parallel peripheral interface that consists of three 8-bit parallel ports. As shipped, these ports are configured for general purpose I/O. Programmed PAL (Programmable Array Logic) devices and the octal transceiver 74LS640-1 are provided to make it easy to reconfigure the parallel interface to be compatible with the SCSI (Small Computer System Interconnect) peripheral interface. Alternatively, the parallel interface may be reconfigured as a Centronics compatible line printer by adding one PAL and reconfiguring jumpers. Both interfaces may use the 82258 DMA controllers for data transfers.

The SCSI interface allows multiple mass storage peripherals such as Winchester disk drives, floppy disk drives, and tape drives to be connected directly to the iSBC 286/100A board. A sample SCSI application is shown in Figure 3. The SCSI interface is compatible with SCSI controllers such as the Adaptek 4500, DTC 1410, Iomga Alpha 10, Shugart 1601 and 1610, Vermont Research 8403, and Xebec 1410.

The Centronics interface requires very little software overhead since a user-supplied PAL device is used to provide necessary handshake timing. Interrupts are generated for printer fault conditions and a DMA request is issued for every character.

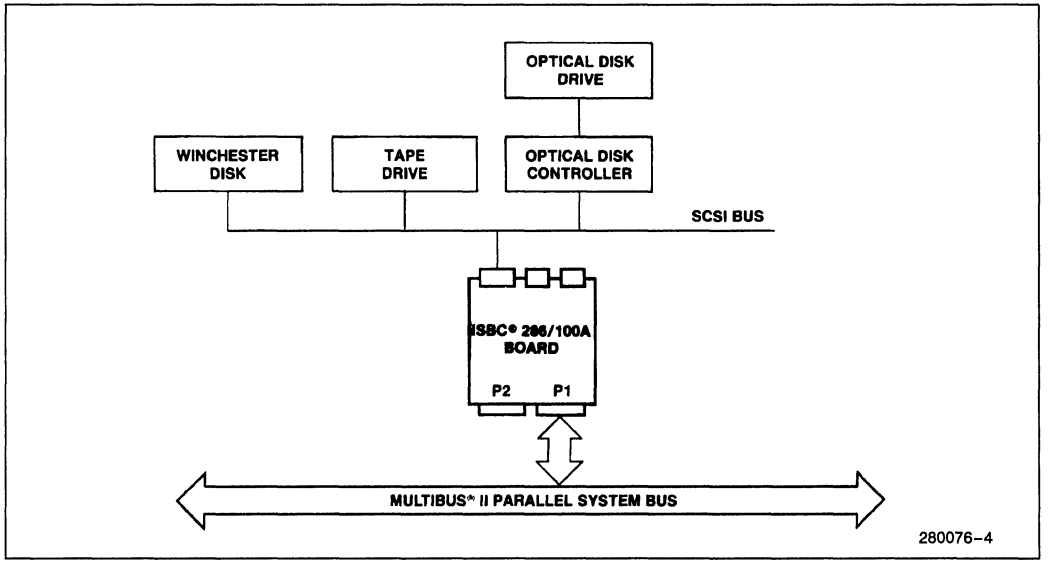


Figure 3. Sample SCSI Applications

SERIAL I/O

The 82530 Serial Communications Controller (SCC) is used to provide two channels of serial I/O. The SCC generates all baudrate clocks and provides loopback capability on both channels. Channel B is RS232C only and is configured as a DCE. Channel A is factory-default configured for DCE RS232C operation. Channel A may be reconfigured by the user for DTE or RS422 operation.

The 82258 ADMA can be programmed to support both channels A and B to perform movement of large bit streams or blocks of data.

PROGRAMMABLE TIMERS

The iSBC 286/100A board provides three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8254 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Three of these timers/counters are available to the system designer to generate accurate time intervals under software control. The outputs may be independently routed to the 8259A Programmable Interrupt Controller to count external events. The system software configures each timer independently to select the desired function. Seven functions are available as shown in Table 2. The contents of each counter may be read at any time during system operation.

SOFTWARE SUPPORT

The iRMX II Operating System software provides the ability to execute all configurable layers of the iRMX II software in the MULTIBUS II environment. The iRMX II Operating System also supports all 80286 component applications.

For on-target MULTIBUS II development, use the iSBX 218A or a SCSI controller and a floppy or Winchester drive, or port iRMX application software developed on the System 310, Series II/III, IV to MULTIBUS II hardware.

Language support for the iSBC 286/100A boards real address mode includes Intel's ASM 86, PL/M 86, PASCAL and FORTRAN as well as many third party 8086 languages. Language support for virtual address mode operation includes ASM 286, PL/M 286, PASCAL and C. Programs developed in these languages can be down-loaded from the Development System to the iSBC 286/100A board via the iSDM 286 System Debug Monitor. The iSBX 218A can be used to load iRMX software developed on a System 310. The iSDM 286 monitor also provides on-target program debugging support including breakpoint and memory examination features.

Table 2. Programmable Time Functions

Function	Operation
Interrupt on Terminal Count	When terminal count is reached, an interrupt request is generated. This function is extremely useful for generation of real-time clocks.
Programmable One-Shot	Output goes low upon request of an external trigger edge or software command and returns high when terminal count is reached. This function is retriggerable.
Rate Generator	Divide by N counter. The output will go low for one input clock cycle, and the period from one low going pulse to the next is N times the input clock period.
Square-Wave Rate Generator	Output will remain high until one-half the count has been completed, and go low for the other half of the count.
Software Triggered Strobe	Output remains high until software loads count (N). N counts after count is loaded, output goes low for one input clock period.
Hardware Triggered Strobe	Output goes low for one clock period N counts after rising edge counter trigger input. The counter is retriggerable.
Event Counter	On a jumper selectable basis, the clock input becomes an input from the external system. CPU may read the number of events occurring after the counter "window" has been enabled or an interrupt may be generated after N events occur in the system.

The MULTIBUS II Interconnect Space Registers allow the software to configure boards eliminating much of the need for jumpers and wire wraps. The iSDM 286 Monitor can initialize these registers at configuration time using user-defined variables. The monitor can also automatically configure memory boards, defining the addresses for each board sequentially in relation to the board's physical placement in the card cage. This feature allows for swapping, adding, and deleting of memory boards on a dynamic basis.

SPECIFICATIONS

WORD SIZE

Instruction— 8-, 16-, 24-, 32-, or 40-bits

Data — 8- or 16-bits

SYSTEM CLOCK

CPU — 8.0 MHz

Numeric Co-Processor — 8.0 MHz

CYCLE TIME

Basic Instruction: 8.0 MHz-375 ns; 250 ns (assumes instruction in queue)

NOTE:

Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles)

Memory Capacity (Maximum)

EPROM: 2732, 8K bytes; 2764, 16K bytes; 27128, 32K bytes; 27256, 64K bytes; 27512, 128K bytes

EEPROM: 2817A, 4K bytes

RAM: 2186, 16K bytes

NOTE:

Two local sites must contain BIST or user-supplied boot-up EPROM.

I/O CAPABILITY

Parallel: SCSI, Centronics, or general purpose I/O

Serial: Two programmable channels using one 82530 Serial Communications Controller

Timers: Three programmable timers using one 8254 Programmable Interrupt Controller

Expansion: One 8/16-bit iSBX MULTIMODULE connector and one 8-bit iSBX MULTIMODULE connector

INTERRUPT CAPABILITY

Potential Interrupt Sources— 255 individual and 1 broadcast

Interrupt Levels — 16 vectored requests using two 8259As and the 80286 NMI line

Serial Communications Characteristics

Asynchronous Modes:

- 5–8-bit character; odd, even, or parity; 1, 1.5, or 2 stop bits
- Independent transmit and receive clocks, 1X, 16X, 32X, or 64X programmable sampling rate
- Error Detection: Framing, Overrun and Parity
- Break detection and generation

Bit Synchronous Modes:

- SDLC/HDLC flag generation and recognition
- Automatic zero bit insertion bit and detection
- Automatic CRC generation and detection (CRC 16 or CCITT)
- Abort generation and detection
- I-field residue handling
- SDLC loop mode operation
- CCITT X.25 compatible

Byte Synchronous Modes:

- Internal or external character synchronization (1 or 2 characters)
- Automatic CRC generation and checking (CRC 16 or CCITT)
- IBM Bisync compatible

Common Baud Rates

Baud Rate	Synchronous (x1 Clock)	Asynchronous (x16 Clock)
	Time Constant	Time Constant
64 K	36	—
48 K	49	—
19.2 K	126	6
9600	254	14
4800	510	30
2400	1022	62
1800	1363	83
1200	2046	126
300	8190	510
110	—	1394

Timers

Input Frequencies: 1.23 MHz ±0.1% or 4 MHz ±0.1% (Jumper Selectable)

Output Frequencies/Timing Intervals

	Single Timer/Counter		Dual Timer/Counter (two timers cascaded)	
	Min	Max	Min	Max
Real-Time Interrupt	500 ns	53.1 ms	1.00 ms	57.9 min
Programmable One-Shot	500 ns	53.1 ms	1.00 ms	57.9 min
Rate Generator	18.8 Hz	2 MHz	0.000290 Hz	1 MHz
Square-Wave Rate Generator	18.8 Hz	2 MHz	0.000290 Hz	1 MHz
Software Triggered Strobe	500 ns	53.1 ms	1.00 ms	57.9 min
Hardware Triggered Strobe	500 ns	53.1 ms	1.00 ms	57.9 min
Event Counter	—	5.0 MHz	—	—

INTERFACES

PSB Bus: All signals TTL compatible

iLBX II Bus: All signals TTL compatible

iSBX Bus: All signals TTL compatible

SERIAL I/O

Channel A: RS232C/RS422 compatible, configurable as a data set or data terminal

Channel B: RS232C compatible, configured as a data set

Timer: All signals TTI compatible

Interrupt Requests: All signals TTL compatible

CONNECTORS

Location	Function	Part #
P1	PSB Bus	603-2-IEC-C096-F
P2	iLBX™ II Bus	603-2-IEC-C096-F

PHYSICAL DIMENSIONS

The iSBC 286/100A board meets all MULTIBUS II mechanical specifications as represented in IEEE 1296 specification.

Double-High Eurocard Form Factor:

Depth:	220 mm (8.7 in.)
Height:	233 mm (9.2 in.)
Front Panel Width:	20 mm (0.784 in.)
Weight:	653 g (1 lb. 7 oz.)

ENVIRONMENTAL REQUIREMENTS

Temperature: (Inlet air) at 200 LFM airflow over boards
Non-operating—−40°C to +70°C
Operating—0 to +55°C

Humidity: Non-operating—95% RH @ 55°C
Operating—90% RH @ 55°C

ELECTRICAL CHARACTERISTICS

The maximum power required per voltage is shown below. These numbers do not include the power required by the optional memory devices, SCSI PALs, or expansion modules.

Voltage (volts)	Max/Typical Current (amps)	Max Power (watts)	BTU	Gram-Calorie
+5	10.31/8.25A	54.39W	3.13	774.2
+12	50/40 mA	630 mW	0.04	9.0
−12	46/37 mA	580 mW	0.03	8.3

REFERENCE MANUALS

ISBC 286/100A Single Board Manual Computer User's Guide (#149093)

Manual may be ordered from any Sales Representative, Distribution Office, or from the Intel Literature Department, 3065 Bowers Ave., Santa Clara, CA 95051

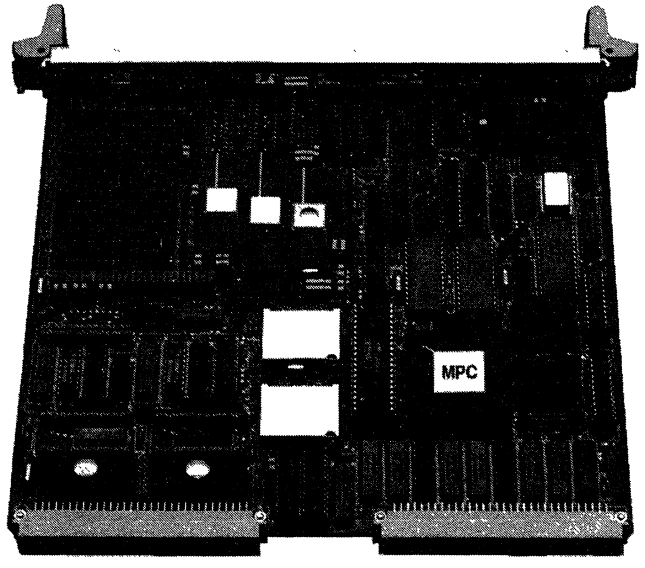
ORDERING INFORMATION

Order Code	Description
SBC286/100A	MULTIBUS II 80286 based Single Board Computer

iSBC® 186/100* MULTIBUS® II SINGLE BOARD COMPUTER

- **8.0 MHz 80186 Microprocessor with Optional High Speed 8087-1 Numeric Data Coprocessor**
- **Optional 82258 Advanced DMA Controller Providing Four Additional High Peformance DMA Channels**
- **On-Board 512K Bytes DRAM Configurable as Dual Port Memory**
- **MPC (Message Passing Coprocessor) Single Chip Interface to the Parallel System Bus with Full Message Passing Capability**
- **Four (Expandable to Eight) 28-Pin JEDEC Sites for PROM, EPROM, or EEPROM**
- **24 Programmable I/O Lines Configurable as SCSI Interface, Centronics Interface, or General Purpose I/O**
- **Two Programmable Serial Interfaces, One RS 232C and the Other RS 422A with Multidrop Capabilities**
- **Resident Firmware Supporting a Reset Operating System, a Program Table, and Build-In-Self-Test (BIST) Diagnostics Including Initialization and Power-Up Tests**
- **8- or 16-bit iSBX™ IEEE 959 Interface Connector with DMA Support for I/O Expansion**

The iSBC® 186/100 Single Board Computer is a member of Intel's family of microcomputer modules that utilizes the advanced features of the MULTIBUS® II system architecture. The 80186-based CPU board takes advantage of VLSI technology to provide economical, off-the-shelf, computer based solutions for OEM applications. All features of the iSBC 186/100 board, including the single chip bus interface (message passing coprocessor), reside on a 220mm x 233mm (8.7 inches x 9.2 inches) Eurocard printed circuit board and provide a complete microcomputer system. The iSBC 186/100 board takes full advantage of the MULTIBUS II bus architecture and can provide a high performance single CPU system or a powerful element for a highly integrated multi-processing application.



280263-1

*The iSBC® 186/100 is also manufactured under product code piSBC® 186/100 by Intel Puerto Rico, Inc.

October 1989
Order Number: 280263-002

FUNCTIONAL DESCRIPTION

Overview

The iSBC 186/100 MULTIBUS II Single Board Computer utilizes the 8 MHz 80186 microprocessor to provide a range of solutions for various low cost OEM and end-user applications. Intel's commitment to offering high performance at a cost effective level are evident in the design of the iSBC 186/100 Single Board Computer. The integration of the functions of a general purpose system (CPU, memory, I/O and peripheral control) into a single board computer imply that the total system's board count, power and space requirements, and costs are reduced. Combining these cost advantages with the advanced features of the MULTIBUS II system architecture, the iSBC 186/100 board is ideal for price sensitive MULTIBUS II multi-processing or single CPU applications. Some of the advanced featues of the MULTIBUS II architecture embodied in the iSBC 186/100 board are distributed arbitration, virtual interrupt capabilities, message passing, iPSB bus parity, and software configurability and diagnostics using interconnect address space.

Architecture

The iSBC 186/100 CPU board supports the PSB bus features of interconnect address space, Built-In-Self-Test (BIST) diagnostics, solicited and unsolicited message passing, and memory and I/O references. In addition to supporting the PSB bus architecture, other functions traditionally found on Intel single board computers are included in the iSBC 186/100 board. These traditional capabilities include iSBX bus expansion; high speed 8087-1 numeric coprocessor; advanced DMA control; JEDEC memory site expansion; SCSI; Centronics; or general purpose configurable parallel I/O interface; serial I/O; and programmable timers on the 808186 microprocessor. Figure 1 shows the iSBC 186/100 board block diagram.

Central Processing Unit and DMA

The 80186 is an 8.0 MHz 16-bit microprocessor combining several common system components onto a single chip (i.e., two Direct Memory Access lines, three Interval Timers, Clock Generator, and Programmable Interrupt Controller). The 80186 instruction set is a superset of the 8086 and maintains object code compatiblity while adding additional instructions.

In the basic configuration, Direct Memory Access (DMA) requests are available between the local memory and the bus interface (see Table 1). With the addition of an Advanced DMA (ADMA) 82258 controller, ADMA requests may be generated by either the iSBX interface, the SCSI interface, the bus interface controller, or the serial interface (see Table 2). The addition of the ADMA controller also allows the serial ports to be used in a full-or half-duplex multidrop application.

An additional high performance 8087-1 Numeric Data Coprocessor may be installed by the user to significantly improve the iSBC 186/100 board's numerical processing power. Depending on the application, the high speed 8087-1 will increase the performance of floating point calculations by 50 to 100 times.

Table 1. Basic DMA Configuration

80186	Local Bus
DMA Channel 0	Output DMA iPSB Bus Interface
DMA Channel 1	Input DMA iPSB Bus Interface

Memory Subsystem

The 1M byte memory space of the 80186 is divided into three main sections. The first section is the 512K bytes of installed DRAM, the second section is the window into the global 4G bytes memory space of the PSB bus (PSB memory window address space) which starts at 512K bytes and goes up to either 640K bytes or 768K bytes, and the third section is designated for local ROM going from the ending address of the PSB memory window address space up to, if desired, 1M byte (see Figure 2).

The iSBC 186/100 board comes with 512K bytes of DRAM installed on the board. This memory can be used as either on-board RAM or Dual Port RAM by loading the start and end addresses into the appropriate interconnect registers. The lower boundary address to the PSB memory window may begin at any 64K byte boundary and the upper boundary address may end at any 64K byte boundary. Refer to the iSBC 186/100 Single Board Computer User's Guide for specific information on programming address spaces into interconnect registers.

The memory subsystem supports 128K bytes or 256K bytes access to the PSB memory address space. The PSB memory window base address is fixed at address 512K. The position of the window in the iPSB memory address space is programmable and thus allows the CPU to access the complete 4G byte memory address space of the MULTIBUS II PSB bus.

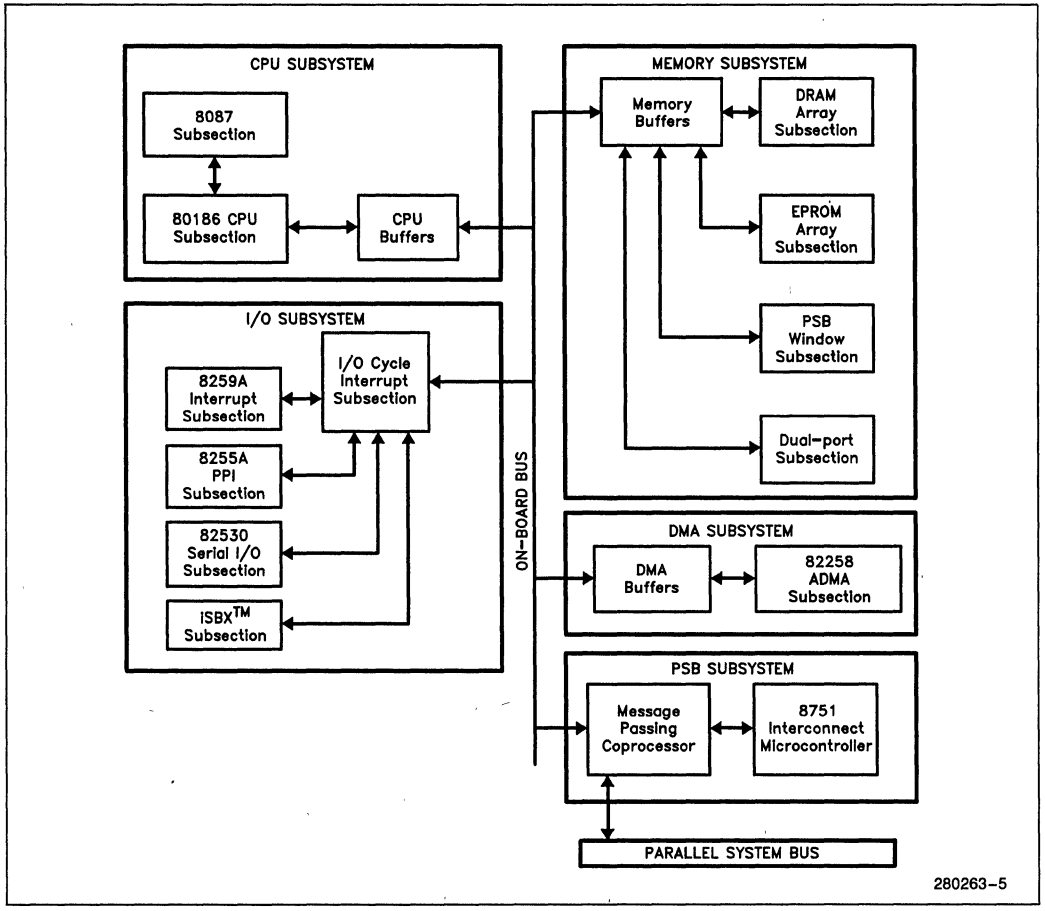


Figure 1. iSBC® 186/100 CPU Board Block Diagram

The ROM space consists of four 28-pin JEDEC sockets which take EPROMs, EEPROMs, or ROMs with 28-pin packages. An iSBC 341 28-pin MULTIMODULE™ EPROM board can be plugged into 2 of the JEDEC sockets and provide up to 512K bytes of ROM memory. Device capacities, which are jumper selectable, are supported from 8K x 8 up to 64K x 8. Once the device capacity is selected, the capacity is uniform for all sockets.

I/O access from the iSBC 186/100 CPU board across the PSB bus is accomplished by mapping 64K bytes of local I/O access one to one to the PSB I/O address space. However, only the upper 32K bytes are available to access the PSB I/O address space because the lower 32K bytes on the iSBC 186/100 board are reserved for local on-board I/O.

On-Board Local Functions

PROGRAMMABLE TIMERS AND INTERRUPT CONTROL

The 80186 microprocessor on the iSBC 186/100 board provides three independent, fully programmable 16-bit interval timers/event counters. In conjunction, two 8259A Programmable Interrupt Controllers (PIC) on the iSBC 186/100 board are used in a master/slave configuration for processing on-board interrupts. At shipment, the 80186 interrupt controller and one PIC are connected as slaves to the master PIC. The first timer on the 80186 microprocessor is routed to the master Programmable Interrupt Controller and the second CPU timer is routed to the slave PIC. This architecture thus supports software

Table 2. DMA Configuration with ADMA Option

80186	Local Bus
DMA Channel 0	Serial Channel B DMA
DMA Channel 1	Serial Channel B DMA or Parallel Port
ADMA 82258	
DMA Channel 0	Input DMA Bus Interface
DMA Channel 1	Output DMA Bus Interface
DMA Channel 2	Half-duplex Fast Serial Interconnect 1 Channel A or Interrupt 1 from iSBX Bus if Used with an iSBC 341 EPROM MULTIMODULE Board
DMA Channel 3	Full-duplex Fast Serial Interconnect 1 Channel A or iSBX Bus DMA Channel if Used with an iSBC 341 EPROM MULTIMODULE board.

NOTE:
When a MULTIMODULE™ expansion board is installed and DMA support is required, then an ADMA controller must also be installed. For additional optional configurations see the *iSBC 186/100 Single Board Computer User's Guide.*

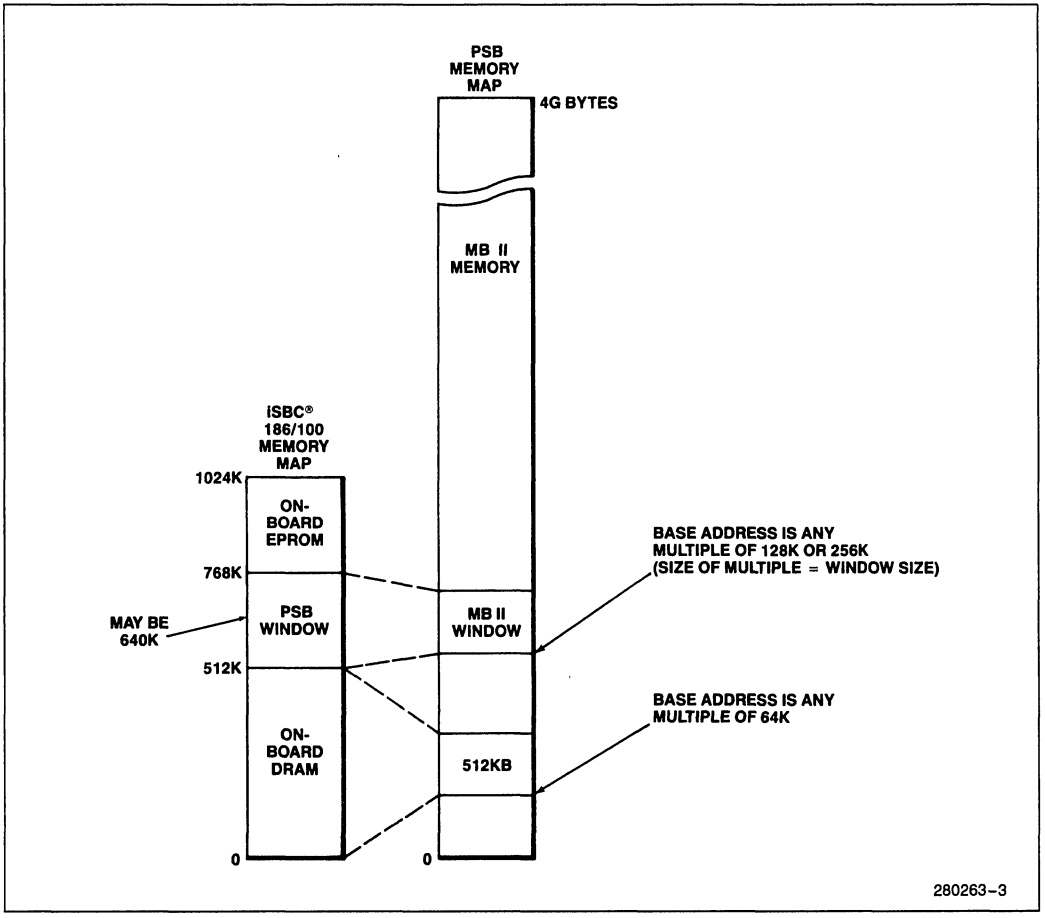


Figure 2. Memory Mapping Diagram

programmable timer interrupts. In addition, direct-vectored interrupt capability of the serial communication controller (SCC) may be used. Figure 3 depicts the interrupts in terms of their priorities.

Interrupt Services	Interrupt Priority
80186 Timer 0	Master Level 0
8087-1 Error Interrupt	1
Message Interrupt	2
iPSB Bus Error Interrupt	3
82530 SCC Interrupt	4
82258 ADMA Interrupt	5
80186 Slave PIC Interrupt	6
8259 Slave PIC Interrupt	7
PPI 0 Interrupt	Slave 0
iSBX Bus Interrupt 0	1
iSBX Bus Interrupt 1	2
Interconnect Space Interrupt	3
80186 Timer 1 Interrupt	4
PPI 1 Interrupt	5
Ground	6 & 7

Figure 3. iSBC® 186/100 Interrupt Priority Scheme

PARALLEL/SCSI PERIPHERAL INTERFACE

The iSBC 186/100 board includes an 8255A parallel peripheral interface that consists of three 8-bit parallel ports. As shipped, these ports are configured for general purpose I/O. Programmed PAL devices (Programmable Array Logic) and the bi-directional octal transceiver 74LS245 are provided to make it easy to reconfigure the parallel interface to be compatible with the SCSI (Small Computer System Interconnect) peripheral interface. Alternatively, the iSBC 186/100 board provides the jumper configuration facilities for operating the parallel interface as an interrupt driven interface for a Centronics compatible line printer by adding one PAL and reconfiguring jumpers. Both interfaces may use the 82258 DMA controller for data transfers if desired.

The SCSI interface allows multiple mass storage peripherals such as Winchester disk drives, floppy disk drives, and tape drives to be connected directly to the iSBC 186/100 board. A sample SCSI application is shown in Figure 4. The SCSI interface is compatible with SCSI controllers such as Adaptek 4500, DTC 1410, Iomega Alpha 10, Shugart 1601 and 1610, Vermont Research 8403, and Xebec 1410.

The Centronics interface requires very little software overhead since a user supplied PAL device is used to provide necessary handshake timing. Interrupts are generated for printer fault conditions and a DMA request is issued for every character.

SERIAL I/O LINES

The iSBC 186/100 board has one 82530 Serial Communciations Controller (SCC) to provide 2 channels of serial I/O. The SCC generates all baudrate clocks and provides loopback capability on both channels. Channel A is configured for RS 422A multidrop DTE application. Channel B is RS 232C only and is configured as DTE.

The multidrop configuration may either full-or half-duplex. A full-duplex multidrop configuration with a single master driving the output lines allow a slave to monitor the data line and to perform tasks in parallel with tasks performed on another slave. However, only the selected slave may transmit to the master. A half-duplex multidrop configuration is more strict in its protocol. Two data lines and a ground line are required between a master and all slaves in the system and although all units may listen to whomever is using the data line, the system software protocol must be designed to allow only one unit to transmit at any given instant.

BUILT-IN-SELF-TEST DIAGNOSTICS

On-board built-in-self-test (BIST) diagnostics are implemented using the 8751 microcontroller and the 80186 microprocessor. On-board tests include initialization tests on DRAM, EPROM, the 80186 microcontroller, and power-up tests. Additional activities performed include iDX, the Initialization and Diagnostics eXecutive which provides initialization at power-up and a program table which allows users to add custom code in EPROM while still maintaining full use of the factory supplied BISTs.

Immediately after power-up and the 8751 microcontroller is intialized, the 80186 microprocessor begins its own initialization and on-board diagnostics. Upon successful completion of these activities, the iDX invokes the user-defined program table. A check is made of the program table and the custom programs that the user has defined for his application will then execute sequentially.

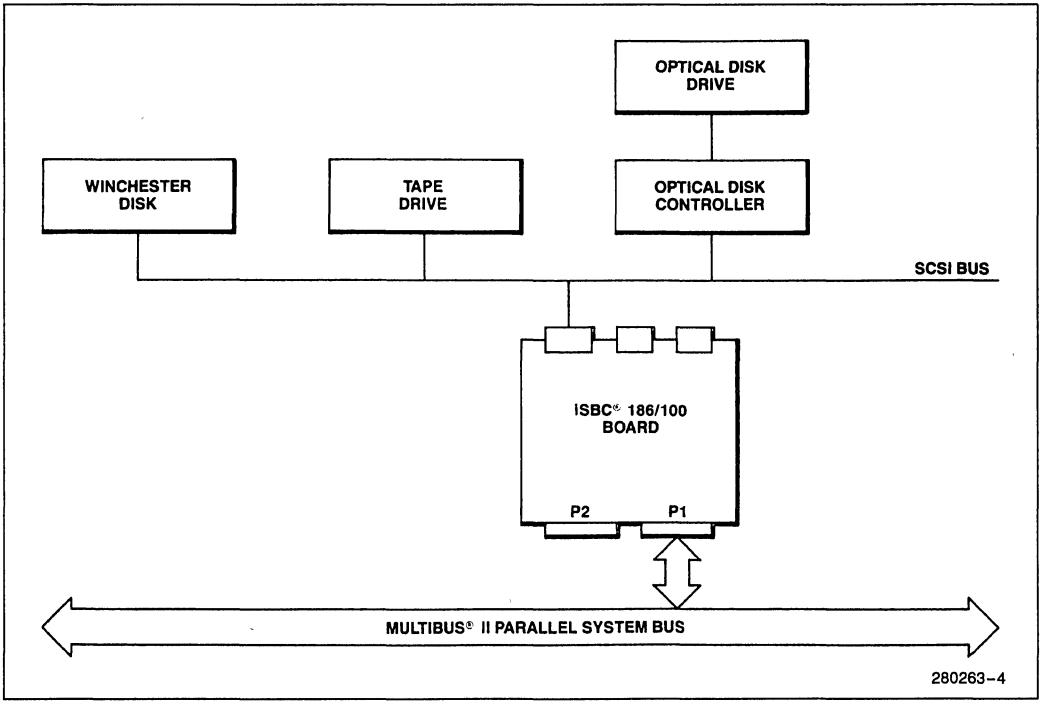


Figure 4. SCSI Application

BISTs improve the reliability, error reporting, and recovery capability of MULTIBUS II boards. In addition, these test and diagnostics reduce manufacturing and maintenance costs for the user. A yellow LED (labeled 'BIST') on the front panel indicates the status of the initialization checks and the power-up tests. It is illuminated if any of the initialization checks fail and remains off if the board successfully completes its tests. The LED also illuminates when the BIST tests start and stays on until the test complete successfully. The results of the BIST diagnostics are stored in the last 6 registers of the Header Record in Interconnect space.

iSBX™ BUS MULTIMODULE™ EXPANSION

One 8-or 16-bit iSBX bus MULTIMODULE connector is provided for I/O expansion. The iSBC 186/100 board supports both 8-bit and 16-bit iSBX modules through this connector. DMA is also supported to the iSBX connector and can be configured by programming the DMA multiplexor attached to the 82258 DMA component. The iSBX connector on the iSBC 186/100 board supports a wide variety of standard MULTIMODULE boards available from Intel and independent hardware vendors. Custom iSBX bus MULTIMODULE boards designed for MULTIBUS or proprietary bus systems are also supported as long as the IEEE 959 iSBX bus specification is followed.

PSB BUS INTERFACE SILICON

The MPC (message passing coprocessor) provides all necessary PSB bus interface logic on a single chip. Services provided by the MPC include memory and I/O access to the PSB by the 80186 processor, bus arbitration, exception cycle protocols, and transfers as well as full message passing support. Dual port architecture may be implemented using the message passing coprocessor.

Interconnect Subsystem

The interconnect subsystem is one of the four MULTIBUS II address spaces, the other three being memory space, I/O space, and message space. The purpose of interconnect space is to allow software to initialize, identify, configure, and diagnose the boards in a MULTIBUS II system. All Intel MULTIBUS II boards support interconnect space.

The interconnect space is organized into a group of 8-bit registers called a template. The interconnect registers are organized into functional groups called records. Each register belongs to only one record, and there are three basic types of interconnect records: a header record, a function record, and an End of Template (EOT) record. The 80186 on the SBC 186/100 board accesses its own template via the interconnect address space on the PSB bus.

The header record provides board and vendor ID information, general status and control information, and diagnostic status and control information. The function record contains parameters needed to perform specific functions for the board. For example, an PSB memory record contains registers that define the start and end address of memory for access across the PSB bus. The number of function records in a template is determined by the manufacturer. The EOT record simply indicates the end of the interconnect template.

There are two types of registers in the MULTIBUS II interconnect space, read-only and software configurable registers. Read-only registers are used to hold information such as board type, vendor, firmware level, etc. Software configurable registers allow read and write operations under software control and are used for auto-software configurability and remote/local diagnostics and testing. Software can be used to dynamically change bus memory sizes, disable or enable on-board resources such as PROM or JEDEC sites, read if an iSBX Board or PROM are installed as well as access the results of Built-In-Self-Tests or user installed diagnostics. Some of the interconnect registers on the iSBC 186/100 board perform functions traditionally done by jumper stakes. Interconnect space support is implemented with the 8751 microcontroller and iPSB bus interface logic.

SPECIFICATIONS

Word Size

INSTRUCTION: 8-, 16-, 24-, 32-, or 40-bits

DATA: 8-or 16-bits

System Clock

CPU: 8.0 MHz

NUMERIC COPROCESSOR: 8.0 MHz (part number 8087-1)

Cycle Time

BASIC INSTRUCTION: 8.0 MHz - 500 ns for minimum code read

Memory Capacity

LOCAL MEMORY

NUMBER OF SOCKETS: four 28-pin JEDEC sites

	Memory Capacity	Chip Example
EPROM	8K × 8	2764
EPROM	16K × 8	27128
EPROM	32K × 8	27256
EPROM	64K × 8	27512

ON-BOARD RAM

512K bytes 64K × 4 bit Dynamic RAM

I/O Capability

Serial:

— Two programmable channels using one 82530 Serial Communications Controller

— 19.2K baud rate maximum in full duplex in asynchronous mode or 1 megabit per second in full duplex in synchronous mode

— Channel A: RS 422A with DTE multidrop capability

— Channel B: RS 232C compatible, configured as DTE

— Parallel: SCSI, Centronics, or general purpose I/O

— Expansion: One 8-or 16-bit IEEE 959 iSBX MULTIMODULE board connector supporting DMA

Serial Communications Characteristics

ASYNCHRONOUS MODES:

- 19.2K baud rate maximum in full duplex
- 5-8-bit character; odd, even, or parity; 1, 1.5, or 2 stops bits
- Independent transmit and receive clocks, 1X, 16X, 32X, or 64X programmable sampling rate
- Error detection: Framing, Overrun, and Parity
- Break detection and generation

BIT SYNCHRONOUS MODES:

- 1 megabit per second maximum in full duplex
- SDLC/HDLC flag generation and recognition
- Automatic zero bit insertion and detection
- Automatic CRC generation and detection (CRC 16 or CCITT)
- Abort generation and detection
- I-field residue handling
- SDLC loop mode operation
- CCITT X.25 compatible

BYTE SYNCHRONOUS MODES:

- Internal or external character synchronization (1 or 2 characters)
- Automatic CRC generation and checking (CRC 16 or CCITT)
- IBM Bisync compatible

Timers

Three programmable timers on the 80186 microprocessor

INPUT FREQUENCIES:

Frequencies supplied by the internal 80186 16 MHz crystal

Serial chips: crystal driver at 9.8304 MHz divide by two

iSBX connector: 9.8304 crystal driven at 9.8304 MHz

Interrupt Capacity

POTENTIAL INTERRUPT SOURCES:

255 individual and 1 broadcast

INTERRUPT LEVELS:

12 vectored requests using two 8259As, 3 grounded inputs, and 1 input to the master PIC from the slave PIC

INTERRUPT REQUESTS:

All signals TTL compatible

Interfaces

PSB BUS:

As per IEEE/ANSI 1296 MULTIBUS II bus architecture specification

ISBX BUS:

As per IEEE 959 specification

CONNECTORS

Location	Function	Part #
P1	PSB Bus	603-2-IEC-C096-F

Physical Dimensions

The iSBC 186/100 board meets all MULTIBUS II mechanical specifications as presented in the MULTIBUS II specification (#146077)

DOUBLE-HIGH EUROCARD FORM FACTOR:

Depth: 220 mm (8.7 in.)
Height: 233 mm (9.2 in.)
Front Panel Width: 20 mm (0.784 in.)
Weight: 743 g (26 oz.)

Environmental Requirements

Temperature: Inlet air at 200 LFM airflow over all boards
Non-operating: −40° to +70°C
Operating: 0° to +55°C

Humidity: Non-operating: 95% RH @55°C, non-condensing
Operating: 90% RH @ 55°C, non-condensing

Electrical Characteristics

The maximum power required per voltage is shown below. These numbers do not include the power required by the optional memory devices, SCSI PALs, or expansion modules.

Voltage (Volts)	Max Current (Amps)	Max Power (Watts)
+5	6.5 mA	34.13W
+12	50 mA	0.06W
−12	50 mA	0.06W

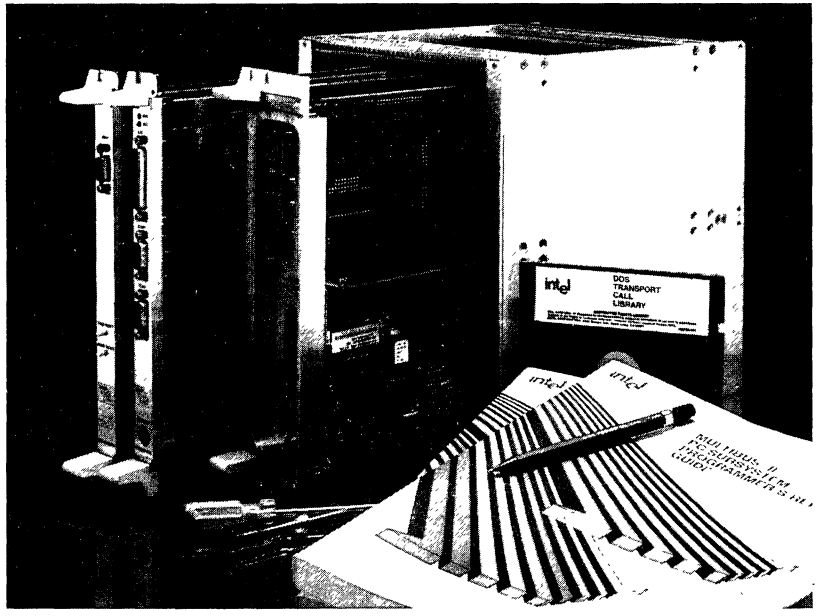
Reference Manuals

iSBC 186/100 Single Board Computer User's Guide (#148732)

Manuals may be ordered from any Sales Representative, Distribution Office, or from the Intel Literature Department, 3065 Bowers Avenue, Santa Clara, CA, 95051.

ORDERING INFORMATION

Part Number	Description
SBC186100	MULTIBUS II 80186-based Single Board Computer



PC/AT COMPATIBILITY COMES TO MULTIBUS®II SYSTEMS*

The Intel MULTIBUS®II PC Subsystem combines the power of the 386™ microprocessor, the multi-processing capabilities of the MULTIBUS II architecture and the large base of DOS compatible software into a high performance IBM PC/AT compatible two board set. When used with a standard PC/AT* compatible keyboard and VGA compatible monitor this subsystem provides an excellent foundation for a human interface with color graphics for MULTIBUS II systems. Running off-the-shelf software packages it is suitable for data acquisition or process monitoring applications, and can be easily customized using a variety of available PC compatible products.

FEATURES

CPU BOARD

- Fully IBM PC/AT compatible subsystem running at 16 Mhz 386 32-bit CPU.
- Includes socket for Intel 80387 or Weitek numeric co-processor chip, 64 K of high speed SRAM cache, 2 serial ports, 1 parallel port, keyboard and floppy drive controllers.
- Completely MULTIBUS II systems architecture compatible including ADMA, MPC and 8751 interconnect controller.

PERIPHERAL COMPANION BOARD

- ST-506/ST-412 compatible Hard Disk Controller
- VGA graphics controller, with VGA, CGA, EGA, and mono-graphics software compatibility
- Built-in CSM functionality

intel

September, 1989
Order Number 280673-002

PART OF THE MULTIBUS®II FAMILY

Now PC/AT* compatibility has come to a MULTIBUS II CPU. The MULTIBUS II Parallel System Bus is the bus of choice for Real Time multiprocessing. Its advanced bus architecture includes such features as a high speed (32 Mbytes/sec) Parallel Systems Bus (PSB) with message passing and bus parity detection, virtual interrupts, simplified systems configuration through interconnect space, and extensive power-up testing. Now our MULTIBUS II family is even more complete with DOS complementing iRMX®, iRMK™, and UNIX* operating systems, and bringing with it a complete human interface including keyboard controller and VGA graphics.

386 MICROPROCESSOR SPEED AND PERFORMANCE

The iSBC 386/PC16 CPU board features a 386™ CPU running at 16 Mhz and 64 K of) wait state (read hit) cache memory for 32-bit speed and performance. Performance can be even further enhanced by adding an Intel 80387 or Weitek math co-processor in the provided socket.

As much as 16 M-byte of DRAM can be provided on-board using memory expansion modules. For full IBM PC/AT software compatbility the iSBC 386/PC16 comes with an Award BIOS and runs either PC-DOS* or MS-DOS*. As a 386™ microprocessor-based PC platform, UNIX V/386 can also be easily ported to this board.

FULL COMPLEMENT OF PC PERIPHERALS

To minimize the need for add-in cards, the iSBC 386/PC16 CPU board includes 2 serial ports, 1 Centronics compatible parallel port, keyboard controller, and floppy disk controller.

The iSBC PCSYS/100 Peripheral Companion Board adds to that a hard disk controller, and a VGA graphics controller which is software compatible with EGA, CGA, and Hercules* monochrome graphics modes. In addition, it provides built-in MULTIBUS II Central Services Module Functionality.

INTEGRATES EASILY INTO A MULTIBUS®II SYSTEM

The ISBC 386/PC16 PCU board was designed to integrate easily into a MULTIBUS II system. Hardware support includes the MULTIBUS II Message Passing Co-processor (MPC), 8751 interconnect space controller, and 82258 ADMA controller to provide full message passing support. It can also access global memory and I/O on the Parallel Systems Bus.

Conforming too the MULTIBUS II Systems Architecture (MSA) the SBC 386/PC16 includes firmware support for BISTs (Built-In Self Tests), IDX (Initialization and Diagnostics eXecutive), and DOS MULTIBUS II Transport Protocol. A DOS Transport Call Library, provided on a floppy disk, allows user implementation of message passing based communication and data sharing with other MULTIBUS II CPUs and peripherals.

BACKPLANES AND ADAPTOR BOARD

Rounding out the complement of products in the Intel MULTIBUSS II PC Subsystem family are 2 and 4 slot backplanes for the Ps/aPC bus (the PC bus brought out on the MULTIBUS II P2 connector) and an Adaptor Board. Intended for development purposes, the iSBC PCSYS/900 Adaptor Board plugs into a MULTIBUS II card cage or chassis and accommodates either four "half size" PC/XT* add-on cards or two "half size" PC/XT and either two PC/AT "full size" or two PC/XT "full size" add-on boards.

HIGH RELIABILITY

Intel has designed the MULTIBUSS II PC Subsystem for high reliability. Extensive use of CMOS circuitry keeps the boards running cooler, and since excess heat can cause premature failure, running longer. DIN pin and socket connectors ensure reliable connectivity with the backplane, and parity error checking in the DRAM circuitry and on the Parallel Systems Bus improves overall system integrity. Furthermore the boards conform to Intel's strict design and manufacturing standards.

WORLD WIDE SERVICE AND SUPPORT

Should this or any Intel board ever need service, Intel maintains a world wide network of service and repair facilities to keep you and your customers up and running. In addition, should you need system level design support, our international Systems Engineering organization is available to integrate Intel boards and systems components into your products.

*UNIX is a trademark of AT&T
*PC-DOS, PC/XT, and PC/AT are trademarks of International Business Machines
*MS-DOS is a trademark of Microsoft
*Hercules is a trademark of Hercules Computer Technology, Inc.

MULTIBUS®II PC SUBSYSTEM CONFIGURATION GUIDE

When the iSBC®386/PC16F0x CPU is used with the iSBC PCSYS/100 Peripheral Companion Board and/or the iSBC PCSYS/900 Adaptor Board either an iSBC PCSYS/602 two-slot or iSBC PCSYS/604 four-slot backplane is required to bus the AT signals between the P2 connectors. Please use the following guide when ordering to select the correct backplane.

386/PC16F0x Only ..... None required, however the iSBC PCSYS/602 2-slot backplane provides a connector which facilitates connecting the floppy drive.

386/PC16F0x and PCSYS/100 Peripheral Companion Board ..... Order iSBC PCSYS/602 2-slot Backplane

386/PC16F01 or F04 and PCSYS/900 Adaptor Board ..... Order iSBC PCSYS/602 2-slot Backplane

386/PC16F02 or F08 and PCSYS/900 Adaptor Board ..... Order iSBC PCSYS/604 4-slot Backplane

386/PC16F0x, PCSYS/100 Peripheral Companion Board and PCSYS/900 Adaptor Board ..... Order iSBC PCSYS/604 4-slot Backplane

NOTE: If stacking multiple memory modules, order ISBC PCSYS/604 4-slot backplane.

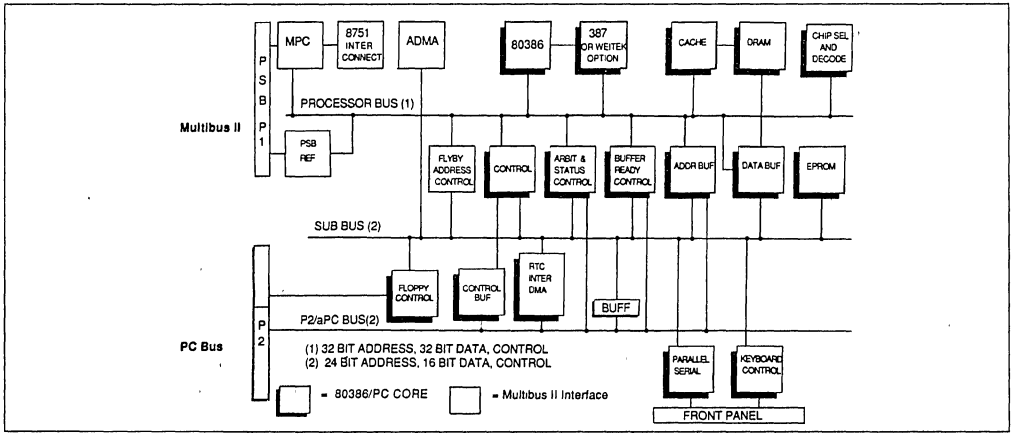


Figure 1: iSBC 386/PC16 Functional Block Diagram

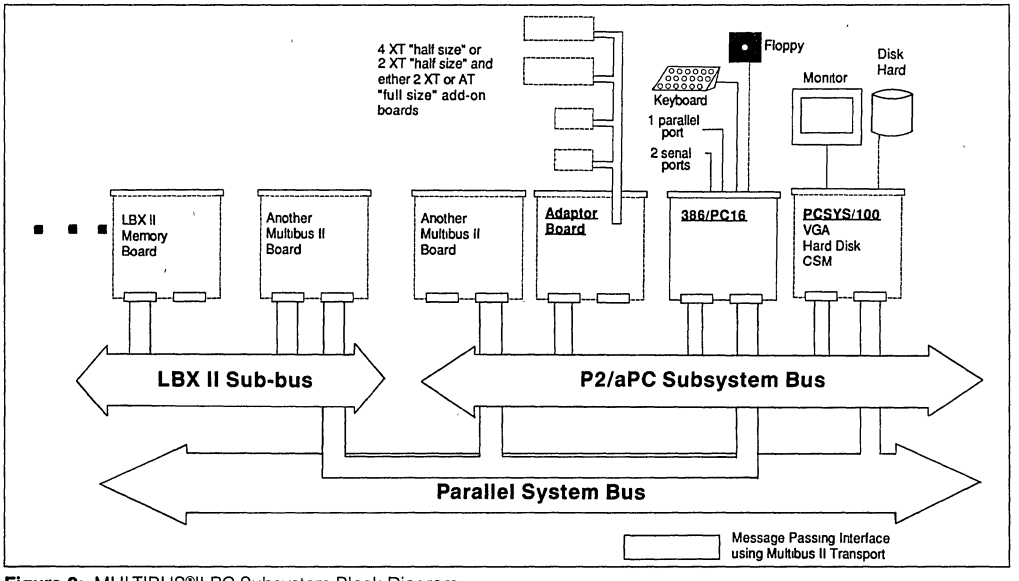


Figure 2: MULTIBUS®II PC Subsystem Block Diagram

CPU BOARD—SBC 386/PC16

CPU

386 microprocessor running at 16Mhz

DRAM Memory

32-bit parity protected memory:

Model	*Supplies*
SBC 386PC16 F01	1 Mb
SBC 386PC16 F02	2 Mb
SBC 386PC16 F04	4 Mb
SBC 386PC16 F08	8 Mb

Note: Model suffixes F02 and F08 require two MULTIBUS II card slots, Model suffixes F01 and F04 require only one MULTIBUS II card slot.

Memory expansion modules—one may be added to base models above

Model	*Supplies*
SBC MM01 FP	1 Mb
SBC MM02 FP	2 Mb
SBC MM04 FP	4 Mb
SBC MM08 FP	8 Mb

SRAM Cache

Capacity: 64K
Speed: 0 wait state on read hit
2 wait states on write
3 wait states on read miss

EPROM Memory

Two 32-pin JEDEC sites containing 256 K of EPROM memory with Awards BIOS and MSA firmware.

Two additional 32-pin JEDEC sites provided for user EPROM or EEPROM memory. Circuitry is provided to write as well as read EEPROM memory.

PERIPHERAL COMPANION BOARD—SBC PCSYS/100

Hard Disk Controller

- PC/AT Compatible Winchester Controller
- Supports up to two ST-506/ST-412 drives

Graphics

- Supports VGA, EGA, CGA, and Hercules Compatible graphics
- Four text mode resolutions: 40 x 25, 80 x 25, 132 x 25, 132 x 43
- Three graphics mode resolutions: 640 x 480 with 16 colors, 960 x 720 with 4 colors, and 1280 x 960 monochrome

CSM

- Assigns card slot and arbitration IDs at initialization
- Generates system clock for all agents on the PSB
- Provides system wide reset signals for power-up, warm reset, and power failure
- Detects bus timeouts

P2/aPC BACKPLANES—SBC PCSYS/602 AND SBC PCSYS/604

- Available in 2 and 4 slot versions

ADAPTOR BOARD—SBC PCSYS/900

- Fully accommodates a total of four half or 3/4 length PC/XT and PC/AT add-in cards in the following combinations: either four PC/XT or two PC/AT and two PC/XT add-in cards
- With restrictions, in some configurations two full size PC/AT or PC/XT add-in cards can be accommodated
- Adaptor board is 3 MULTIBUS II card slots wide

ENVIRONMENTAL REQUIREMENTS

Storage Temperature:	–40° to 70°C (0° to 158°F)
Operating Temperature:	0°C to 55°C (32° to 131°F)
Storage Humidity:	5%-95% non-condensing at 55°C
Operating Humidity:	8%-90% non-condensing at 55°C

ORDERING INFORMATION

SBC386PC16F01	386-based PC compatible CPU board with 1 Mb of DRAM
SBC386PC16F02	386-based PC compatible CPU board with 2 Mb of DRAM
SBC386PC16F04	386-based PC compatible CPU board with 4 Mb of DRAM
SBC386PC16F08	386-based PC compatible CPU board with 8 Mb of DRAM
SBCPCSYS100	Companion board with VGA graphics, HD controller and CSM functionality
S301K3	101-key enhanced AT-style keyboard
SBCPCSYS602	2-slot Backplane for the P2/aPC bus
SBCPCSYS604	4-slot Backplane for the P2/aPC bus
SBCPCSYS900	Adaptor Board
SBC MM01 FP	1 Mb Memory Expansion Module
SBC MM02 FP	2 Mb Memory Expansion Module
SBC MM04 FP	4 Mb Memory Expansion Module
SBC MM08 FP	8 Mb Memory Expansion Module

MULTIBUS® II
Memory Expansion Boards

2

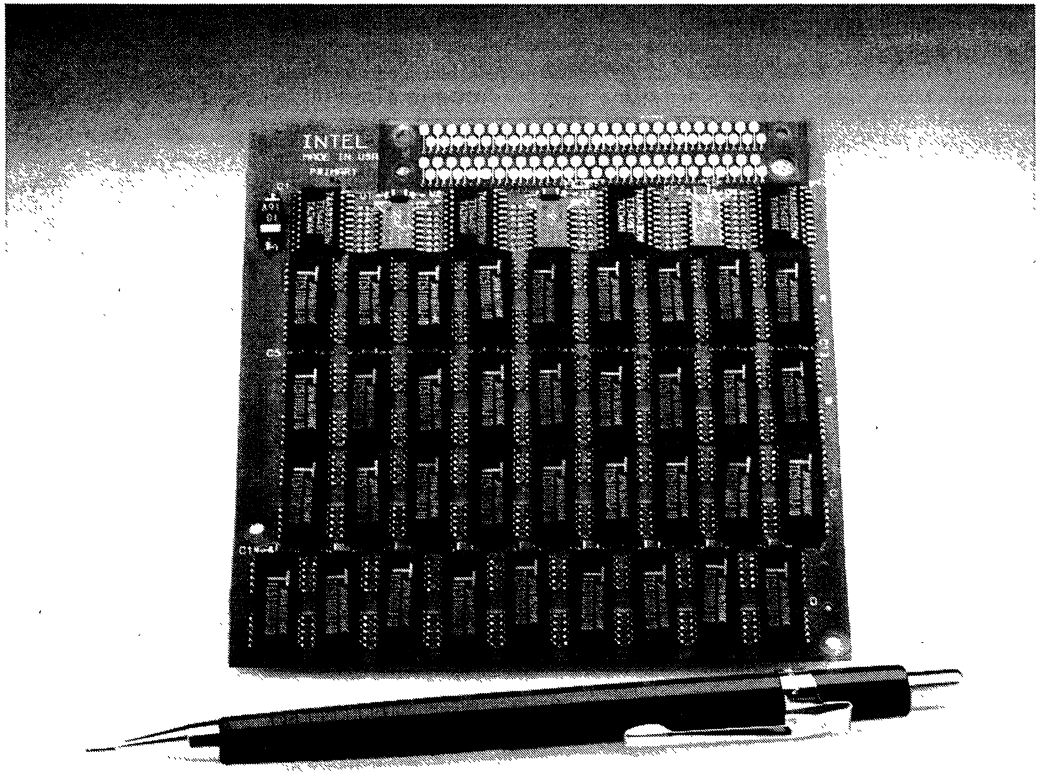
iSBC® MM01, MM02, MM04, MM08* HIGH PERFORMANCE MEMORY MODULES

- **Provides High Speed Parity Memory Expansion for Intel's iSBC® 386/2X, iSBC 386/3X and iSBC 386/1XX CPU Boards**
- **Available in 1M, 2M, 4M, and 8M Byte Sizes**
- **32 Bits Wide with Byte Parity**
- **Stackable to Provide up to 16M Bytes of High Speed Memory for MULTIBUS I and MULTIBUS II CPU Boards**
- **Supports 32-Bit, 16-Bit and 8-Bit Data Paths**
- **Supports Independent Read/Writes**
- **Easily Installed**

The iSBC MM01, iSBC MM02, iSBC MM04, and iSBC MM08 DRAM memory modules are members of Intel's complete line of iSBC memory and I/O expansion boards. The MM-Series of memory modules use a dedicated interface to maximize CPU/memory performance. The iSBC MM series of memory modules have been designed to provide both the on-board and expansion memory for the iSBC 386/2X, the iSBC 386/3X and the iSBC 386/1XX CPU Boards.

The modules contain (respectively) 1M byte, 2M, 4M, and 8M bytes of read/write memory using surface mounted DRAM components (see Figure 1).

Due to the high speed interface of the memory modules, they are ideally suited in applications where memory performance is critical.



280346-1

Figure 1. iSBC® MM08 Memory Module

*The iSBC® MM01, MM02, MM04, MM08 Memory Modules are also manufactured under product code piSBC® MM01, MM02, MM04, MM08 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280346-002

FUNCTIONAL DESCRIPTION

The iSBC MMxx memory modules provide high performance, 32-bit parity DRAM memory for the MULTIBUS I and MULTIBUS II CPU boards. These CPU boards come standard with one MMxx module installed, with memory expansion available through the addition of a second stackable iSBC MMxx module.

Memory Access Capabilities

The dynamic RAM memory of the memory modules is accessed through the dedicated memory module interface.

The MM memory module is designed for direct transfer of data between the CPU and the memory module without accessing the MULTIBUS interface.

MM01/MM02/MM04/MM08 Memory Size

The iSBC MM01, iSBC MM02, iSBC MM04, and iSBC MM08 modules can be stacked on the CPU baseboard in any combination.

Data Bus Structure

The MMxx-series memory modules use a 32-bit wide data path with storage for byte parity that can accommodate 8-bit byte, 16-bit or 32-bit word data transfers. In addition, the data path is capable of independent byte operations. This means that one byte can be written while the other three bytes (or any other combination) can be read.

Parity

One parity bit is provided for each of the four, 8-bit bytes in the 32-bit wide data path. For special applications, the parity bits can serve as data bits making possible 9-, 18-, or 36-bit data transfers.

Memory Function

The module protocol supports standard dynamic RAM READ, WRITE, RAS* only REFRESH cycles, and CAS* before RAS* REFRESH.

Installation

The iSBC MMxx memory modules are easily installed by the user. Each module includes all necessary connectors, screws, and other hardware for installation, either as a second stacked module or as a replacement for a module with less memory.

SPECIFICATIONS

Word Size Supported

8-, 16-, or 32-bits

Memory Size

iSBC MM01	1,048,576 bytes
iSBC MM02	2,097,152 bytes
iSBC MM04	4,194,304 bytes
iSBC MM08	8,388,608 bytes

Access Time (All Densities)

Read/Write — 107 ns (max)

The MMxx-series memory modules run with the iSBC 386/2X and iSBC 386/116 Boards at 16 MHz, and with the iSBC 386/3X and iSBC 386/120 Boards at 20 MHz. Wait state performance information with each of these CPU baseboards is contained in the Hardware Reference Manual for the specific CPU baseboard.

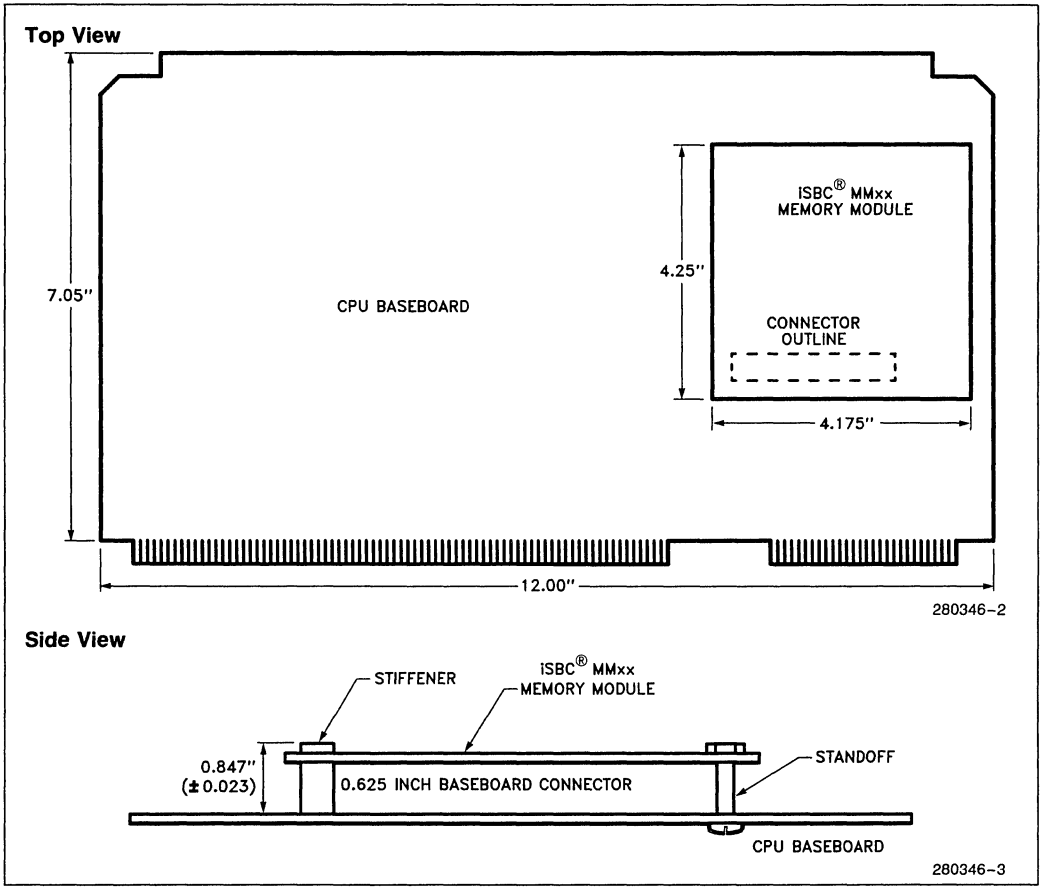
Cycle Time (All Densities)

Read/Write — 200 ns (min)

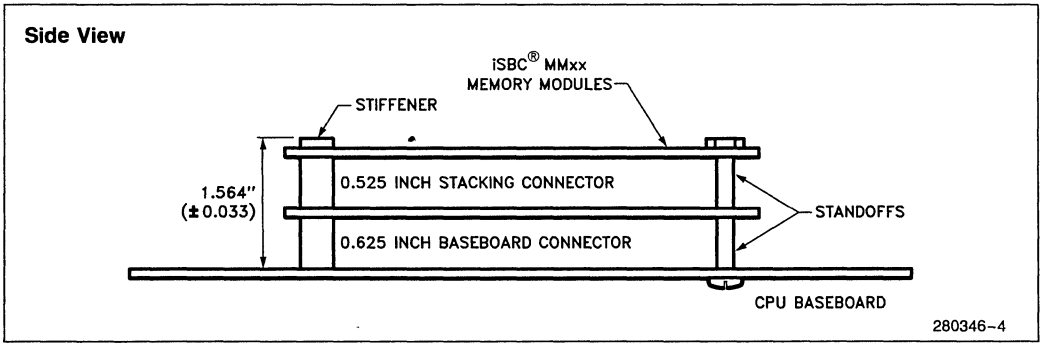
Power Requirements

Voltage −5 VDC ±5%

Memory addressing for the iSBC MMxx memory modules is controlled by the host CPU board over the memory module interface. The maximum system RAM size is 16M Bytes.



Single iSBC® MMxx Memory Module



Stacked iSBC® MMxx Memory Modules

Environmental Requirements

Operating Temperature — 0°C to 60°C

Storage Temperature — 40°C to +75°C

Cooling Requirement — 3 cubic feet per minute of airflow at an ambient temperature of 0°C to 60°C

Operating Humidity — To 95% relative humidity without condensation

Physical Dimensions

Module Alone:

Width — 4.250 inches (10,795 cm)

Length — 4.175 inches (10,604 cm)

Height — 0.500 inches (1,270 cm)

Weight — iSBC MM01/MM04: 2.5 ounces (70.0 gm)
iSBC MM02/MM08: 3.5 ounces (110.0 gm)

ORDERING INFORMATION

Part Number	Description
iSBC MM01	1M Byte RAM Memory Module
iSBC MM02	2M Byte RAM Memory Module
iSBC MM04	4M Byte RAM Memory Module
iSBC MM08	8M Byte RAM Memory Module

The Memory Modules ship with the required hardware (connectors, mounting screws, stand-offs, etc.) to stack a second module on the module already mounted on the base CPU board.

iSBC® MM01FP, MM02FP, MM04FP, MM08FP* HIGH PERFORMANCE MEMORY MODULES

- **Provides High Speed Parity Memory Expansion for Intel's iSBC® 386/2X, iSBC 386/3X and iSBC 386/1XX CPU Boards**
- **Available in 1M, 2M, 4M, and 8M Byte Sizes**
- **32 Bits Wide with Byte Parity**
- **Stackable to Provide up to 16M Bytes of High Speed Memory for MULTIBUS I and MULTIBUS II Boards**
- **Supports 32-Bit, 16-Bit and 8-Bit Data Paths**
- **Supports Independent Read/Writes**
- **Easily Installed**

The iSBC MM0X and iSBC MM0XFP DRAM memory modules are members of Intel's complete line of iSBC memory and I/O expansion boards. The MM-Series of memory modules use a dedicated memory interface to maximize CPU/memory performance.

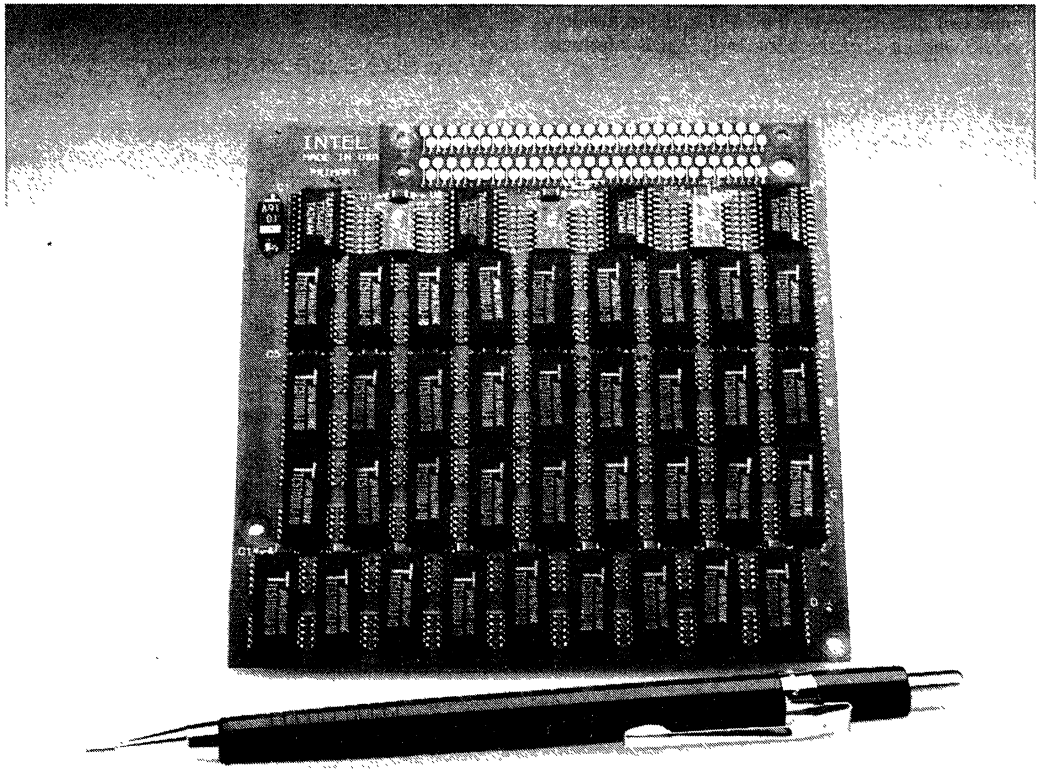


Figure 1. iSBC® MM08FP Memory Module 281010-1

The iSBC® MM01FP, MM02FP, MM04FP, MM08FP memory modules are also manufactured under product code piSBC® MM01FP, MM02FP, MM04FP, MM08FP by Intel Puerto Rico, Inc.

September 1989
Order Number: 281010-001

FUNCTIONAL DESCRIPTION

The iSBC MM-Series provide high performance, 32-bit parity DRAM memory for the MULTIBUS I and MULTIBUS II boards. These CPU boards come standard with one MM-Series module installed, with memory expansion available through the addition of a second stackable iSBC MM-Series module.

Memory Access Capabilities

The dynamic RAM memory of the memory modules is accessed through the dedicated memory module interface.

The MM memory module is designed for direct transfer of data between the CPU and the memory module without accessing the MULTIBUS interface.

MM01/MM02/MM04/MM08 Memory Size

The iSBC MM01, iSBC MM02, iSBC MM04, and iSBC MM08 modules can be stacked on the CPU baseboard in any combination.

Data Bus Structure

The MM-Series memory modules use a 32-bit wide data path with storage for byte parity that can accommodate 8-bit byte, 16-bit or 32-bit word data transfers. In addition, the data path is capable of independent byte operations. This means that one byte can be written while the other three bytes (or any other combination) can be read.

Parity

One parity bit is provided for each of the four, 8-bit bytes in the 32-bit wide data path. For special applications, the parity bits can serve as data bits making possible 9-, 18-, or 36-bit data transfers.

Memory Function

The module protocol supports standard dynamic RAM READ, WRITE, RAS* only REFRESH cycles, and CAS* before RAS* REFRESH.

Installation

The iSBC MM-Series memory modules are easily installed by the user. Each module includes all necessary connectors, screws, and other hardware for installation, either as a second stacked module or as a replacement for a module with less memory.

SPECIFICATIONS

Word Size Supported

8-, 16-, or 32-bits

Memory Size

iSBC MM01	1,048,576 bytes
iSBC MM02	2,097,152 bytes
iSBC MM04	4,194,304 bytes
iSBC MM08	8,388,608 bytes

Access Time (All Densities)

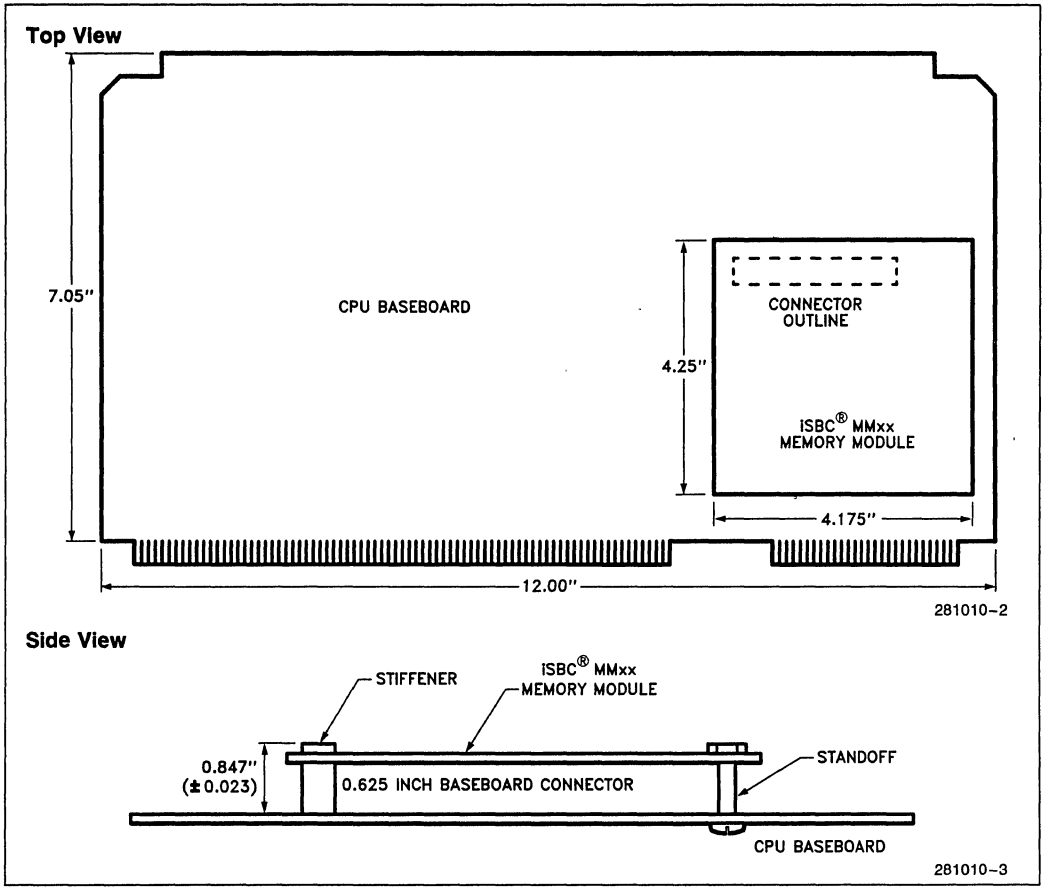
Read/Write — 107 ns (max)-MM0X

Read/Write — 88 ns (max)-MM0XFP

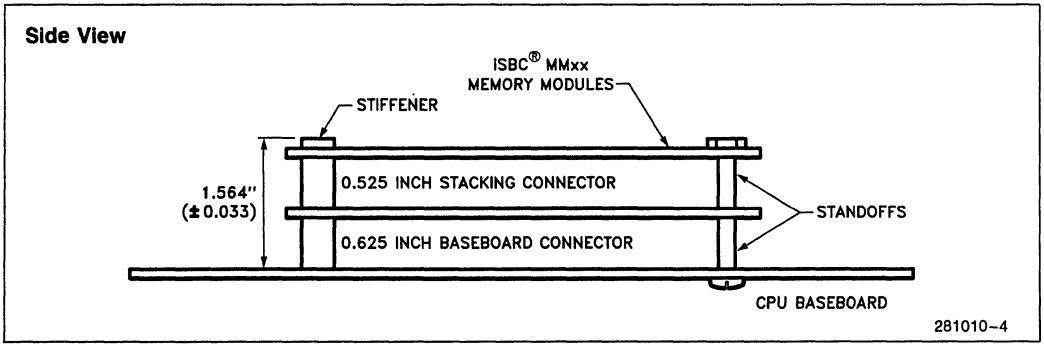
Power Requirements

Voltage −5 VDC ±5%

Memory addressing for the iSBC MM-Series memory modules is controlled by the host CPU board over the memory module interface. The maximum system RAM size is 16M Bytes.



Single iSBC® MMxx Memory Module



Stacked iSBC® MMxx Memory Modules

Environmental Requirements

Operating Temperature — 0°C to 60°C

Storage Temperature — 40°C to +75°C

Operating Humidity — To 95% relative humidity without condensation

Physical Dimensions

Module Alone:

Width — 4.250 inches (10,795 cm)

Length — 4.175 inches (10,604 cm)

Height — 0.500 inches (1,270 cm)

Weight — iSBC MM01/MM04: 2.5 ounces (70.0 gm)
iSBC MM02/MM08: 3.5 ounces (110.0 gm)

ORDERING INFORMATION

Part Number	Description
iSBC MM01FP	1M Byte Fast Page Memory Module
iSBC MM02FP	2M Byte Fast Page Memory Module
iSBC MM03FP	4M Byte Fast Page Memory Module
iSBC MM04FP	8M Byte Fast Page Memory Module

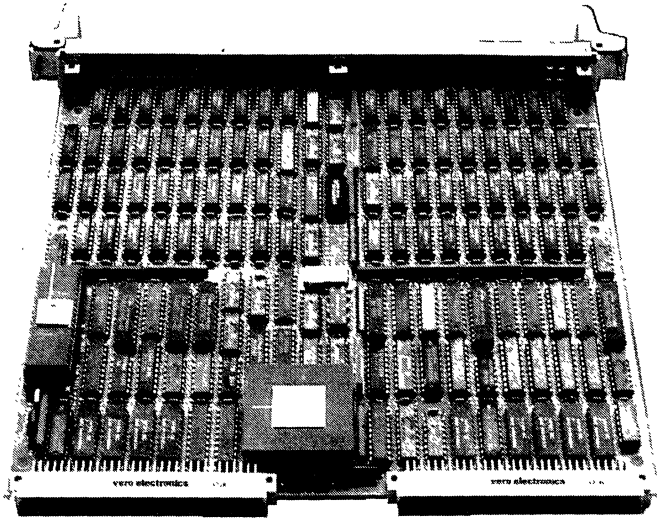
The Memory Modules ship with the required hardware (connectors, mounting screws, stand-offs, etc.) to stack a second module on the module already mounted on the base CPU board.

iSBC® MEM/320, 340*
CACHE-BASED MULTIBUS® II RAM BOARDS

- **32-bit MULTIBUS® II Parallel System Bus (PSB) and Local Bus Extension II (iLBX™ II Bus) Interface Support**
- **Zero Wait State Over iLBX™ on a Cache Hit, One Wait State for Cache Misses and Writes at 8 MHz**
- **Dual Port Memory with Four Versions Available:**
  - **iSBC MEM/320 2M Bytes**
  - **iSBC MEM/340 4M Bytes**
- **MULTIBUS II Interconnect Space for Dynamic Memory Configuration and Diagnostics**
- **Built-In-Self-Test (BIST) Diagnostics On-Board with Both LED Indicators and Software Access to Error Information**
- **Automatic Memory Initialization at Power-Up and at Power-Fail Recovery**
- **Byte Parity Error Detection**

The iSBC MEM/320, 340 are cache-based memory boards that support the MULTIBUS II architecture. They have 32-bit architecture throughout, supporting 8-, 16-, and 32-bit central processors. The iSBC MEM/3XX (generally refers to this family of boards) memory boards are dual-ported, with access to the interfaces of both the MULTIBUS II Parallel System Bus (PSB bus) and the iLBX™ II (Local Bus Extension).

In addition to the 32-bit memory transfer, the iSBC MEM/3XX high-speed cache control subsystem, standard on these boards, improves performance by allowing zero wait state read access over the iLBX II at 8 MHz when data requested is in the cache memory.



280071-1

*The iSBC® MEM/320, 340 is also manufactured under product code piSBC® MEM/320, 340 by Intel Puerto Rico, Inc.

October 1989
Order Number: 280071-003

FUNCTIONAL DESCRIPTION

General

The iSBC MEM/320, 340 high-speed cache-based memory boards are physically and electrically compatible with the MULTIBUS II PSB bus standard and the new iLBX II bus (Local Bus Extension) as outlined in the Intel MULTIBUS II specification. Figure 1 illustrates a typical multiprocessing MULTIBUS II system configuraton.

Architecture

The four main subsystems of the iSBC MEM/3XX boards are the cache controller subsystem, the cache memory subsystem, the DRAM memory subsystem, and the interconnect space subsystem (see Figure 2). The following sections describe these subsystems and their capabilities in more detail.

Cache Memory Capabilities

The cache memory system is designed around the 32-bit architecture of the main memory system and reduces read access timers. The 8K Bytes of 45 nsec SRAM allows zero wait state read accesses over the iLBX II bus when data requested is in the cache memory (cache hit). A cache hit takes only two iLBX II bus clocks (250 nsec at 8 MHz).

Each entry in the 8K Byte cache memory subsystem consists of a data field of 32-bits and a tag field of up to 9-bits (depending on board DRAM size). Each byte in the main memory DRAM array directly maps to one and only one entry on the cache array. This direct mapped cache array along with tag labels ensure data integrity and accurate identification of cache hits. The cache memory size and simple but effective replacement algorithm is designed to optimize both the probability of cache hits and the CPU bus utilization. On any miss or write access, the contents of one cache entry are updated to maintain consistency with the corresponding entry in the DRAM memory array.

Dual Port DRAM Capabilities

The MEM/320 and MEM/340 modules respectively contain 2M Bytes and 4M Bytes of read/write memory using 256K dynamic RAM components.

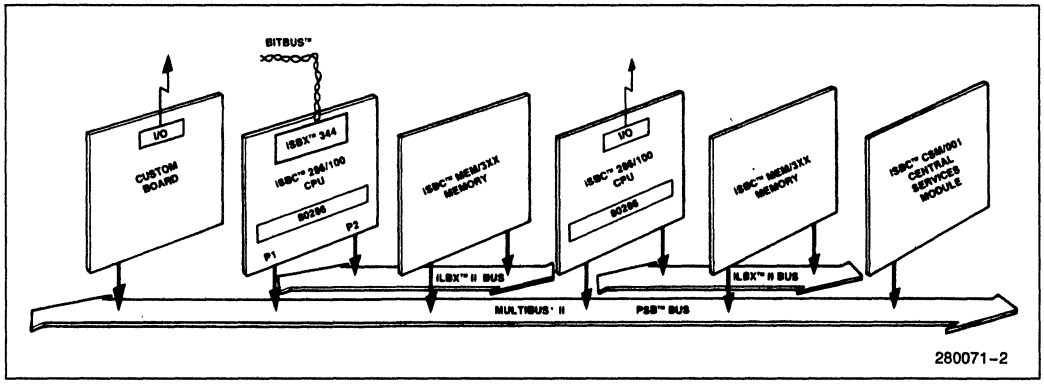


Figure 1. Typical MULTIBUS® II System Configuration

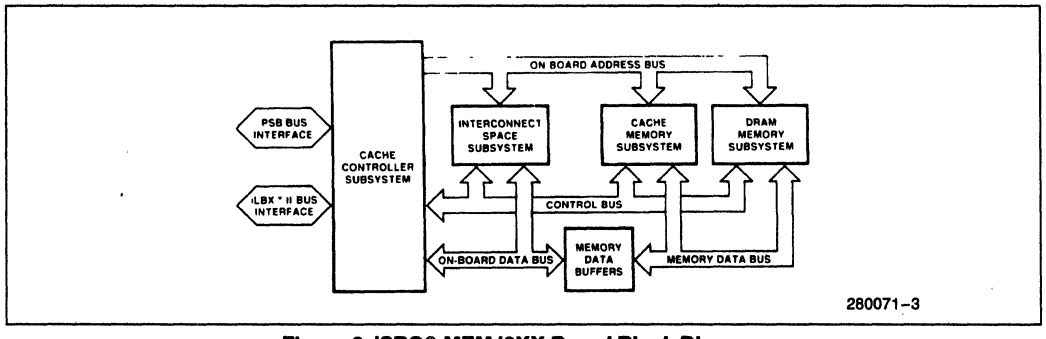


Figure 2. iSBC® MEM/3XX Board Block Diagram

The dual port capability of the iSBC MEM/3XX boards allows 32-bit access from either the PSB bus interface or the iLBX II bus interface (see Figure 1). Due to the simple arbitration nature of the iLBX II bus interface and the cache memory subsystem, the iSBC MEM/3XX family allows optimal access to 20M Bytes of DRAM on the iLBX II bus.

System Memory Size

Using this series of memory boards the maximum system memory capacity based on one CPU board and 19 memory boards is 76M Bytes on the PSB bus. The memory partitioning is independent for the PSB bus interface and the iLBX II bus interface.

The start address can be on any 64K Byte boundary on the PSB bus and any 64K Byte boundary on the iLBX II bus. Software configures the start and ending addresses through the interconnect space. No jumpers are needed.

Interconnect Space Capabilities

The iSBC MEM/3XX board module has a set of interconnect registers which allow the system software to dynamically configure and test the status of the memory board, replacing hardwired jumper functions. This interconnect subsystem also provides control and access to the Built-In-Self-Test (BIST) features. During power-up reset, the iSBC MEM/3XX board initializes the memory and cache, sets all interconnect registers to their default values and performs a self-test. Error information from both Built-In-Self-Test (BIST) and parity checking is indicated in front panel LEDs and recorded in interconnect space registers accessible to software.

Built-In-Self-Test (BIST)

Self-test/diagnostics have been built into the heart of the MULTIBUS II system. These confidence tests and diagnostics improve reliability and reduce manufacturing and maintenance costs. LED 1 (labelled BIST) is used to indicate the status of the Built-In Self Test. It is turned on when the BIST starts running and is turned off when the BIST completes successfully. The Built-In-Self-Test performed by the on-board microcontroller at power-up or at software command are:

1. **EPROM Checksum:**

   This test performs a checksum test on its internal EPROM to check operation of the 8751 microcontroller.

2. **Cache Data Test:**

   The microcontroller performs a sliding ones test on the cache memory in hit-only mode.

3. **Cache Address Test:**

   This test verifies that the cache address path is working properly.

4. **Refresh Check:**

   This test performs RAM test on a small portion of DRAM with an elapsed time between the write operation and the verification of the data.

5. **Dynamic RAM Address Test:**

   This test performs Address Rippled RAM test on the board memory (MISS ONLY operation mode).

6. **Dynamic RAM Data Test:**

   This test runs an AA-55 data pattern to check the DRAM data path.

7. **Parity Test:**

   This test injects parity errors in the DRAM array and then verifies that the board detects these errors.

These tests are described in detail in the User's Manual, Section 9–23.

Memory Initialization and Reset

Memory is initialized automatically during power-up. All bytes are set to 00.

Error Detection Using Byte Parity

Parity will detect all single bit parity errors on a byte parity basis and many multiple bit errors. LED 2 (labelled Parity) is used to indicate parity errors. LED 2 is turned on when a parity error is detected and turned off when the parity status register within interconnect space is cleared. This same LED turns on and off during power-up to verify operatoin of the LED.

Error information is recorded in interconnect space so it is accessible to software for error reporting.

SPECIFICATIONS

Word Size Supported

8-, 16-, 24-, and 32-bits

Memory Size

2 Megabytes (iSBC MEM/320) board
4 Megabytes (iSBC MEM/340) board

Access Times (All Densities)

MULTIBUS II Parallel System Bus—PSB (@ 10 MHz)

Read: 562 ns (avg.)
775 ns (max.)

Write: 662 ns (avg.)
775 ns (max.)

NOTE:
Average access times assume 80% cache hit rates

iLBX™ II Bus—Local Bus Extension (at 8 MHz)

Read: 250 ns (min.)
275 ns (avg.)
375 ns (max.)

Write: 375 ns (avg.)
375 ns (max.)

Base Address

PSB Bus—any 64K Bytes boundary
iLBX II Bus—any 64K Bytes boundary

Power Requirements

Voltage: 5V DC ±5%

Product	Current
iSBC MEM/320 Board	3.5 A (typ) 6.0 A (max)
iSBC MEM/340 Board	4.1 A (typ) 6.7 A (max)

ENVIRONMENTAL REQUIREMENTS

Temperature: (inlet air) at 200 LFM airflow over boards
Non-Operating: −40 to +70°C
Operating: 0 to +55°C

Humidity: Non-operating: 95% RH @ 55°C
Operating: 90% RH @ 55°C

Physical Dimensions

The iSBC MEM/3XX boards meet all MULTIBUS II mechanical specifications as presented in the ANSI/IEEE 1296 MULTIBUS II specification.

Double High Eurocard Form Factor:

Depth: 220 mm (8.6 in.)
Height: 233 mm (9.2 in.)
Front Panel Width: 20 mm (0.784 in.)

Weight:
iSBC MEM/320 board: 6720 gm (24 oz.)
iSBC MEM/340 board: 10080 gm (36 oz.)

Reference Manuals

iSBC MEM/3XX Board Manual (#146707)

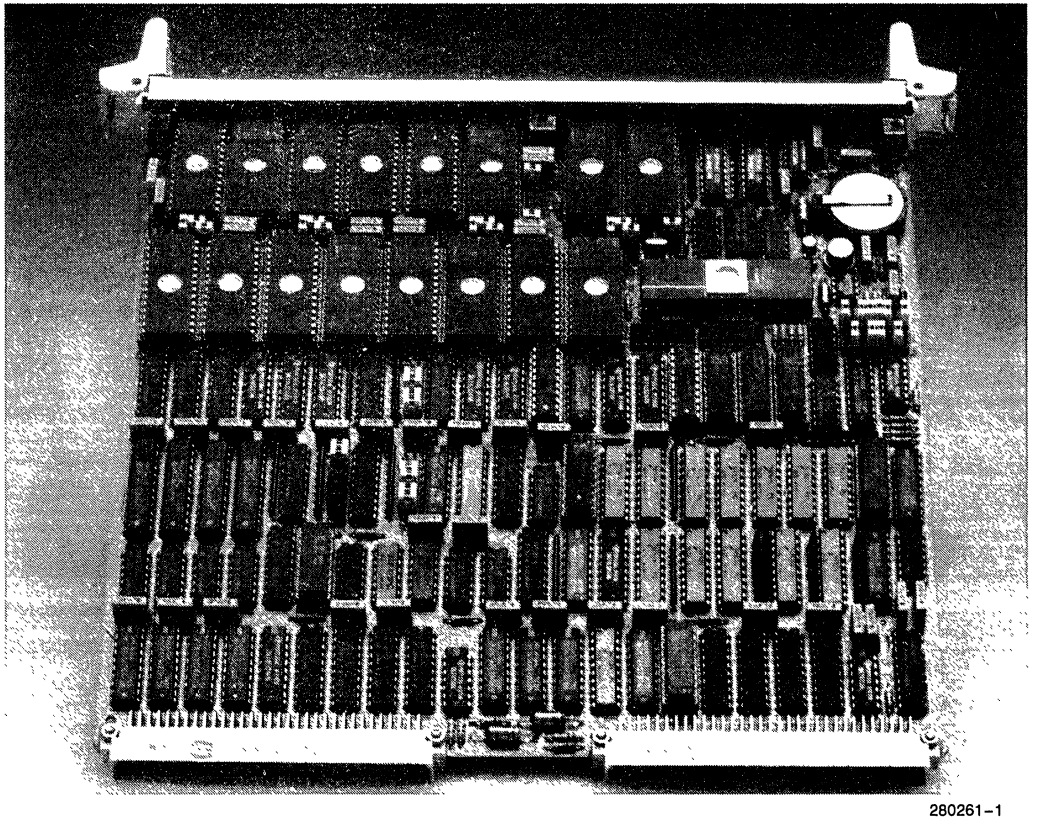
Ordering Information

Part Number	Description
SBCMEM320	2M Byte Cache Based MULTIBUS II RAM Board
SBCMEM340	4M Byte Cache Based MULTIBUS II RAM Board

iSBC® MEM/601 MULTIBUS® II UNIVERSAL SITE MEMORY EXPANSION BOARD

- **Supports EPROM, ROM, EEPROM, SRAM, and NVRAM**
- **Sixteen Sites Configured as Two Banks of Eight 28-Pin JEDEC Sockets**
- **Start Addresses for Each Bank Independently Assignable Anywhere on 64K Byte Boundaries Within the 4G Byte PSB Memory Address Space**
- **Automatic Memory Initialization at Power-Up**
- **Optional On-Board Support for Lithium Battery Backup Memory Protect**
- **MULTIBUS® II Software Interconnect Support for Dynamic Memory Configuration and Diagnositics**
- **Fully Supports Either MULTIBUS II 32-Bit Parallel System Bus (PSB) or 32-Bit Local Bus Extension (iLBX™ II) Bus**

The iSBC® MEM/601 MULTIBUS II Universal Site Memory Board is a member of Intel's line of product offerings that utilize the advanced features of the MULTIBUS II system architecture. The iSBC MULTIBUS II Universal Site Memory Board expands system memory capacity and interfaces across either the MULTIBUS II Parallel System Bus (PSB) or the high speed Local Bus Extension bus (iLBX II).



280261-1

October 1989
Order Number: 280261-001

FUNCTIONAL DESCRIPTION

General

The iSBC MEM/601 board contains two banks of eight standard 28-pin 600 mil DIP sockets. Either 28- or 24-pin devices may be inserted on the board. The actual capacity of the board is determined by the type and quantity of components installed by the user. The iSBC MEM/601 board is completely compatible with four different types and densities of devices (see Table 1). In addition, the board can be accessed by either the MULTIBUS II Parallel System Bus (PSB) or Local Extension Bus (iLBX II).

Memory Array

The sixteen universal memory sites on the iSBC MEM/601 board are partitioned into two banks of 8 sites each. Within each bank the 8 sites are further partitioned into 2 groups of 4 sites each (see Figure 1). Each group of 4 sites can support the device types described in Table 1 and is configurable via an arrangement of push-in jumpers dedicated to each of the four groupings of 4 sites. Devices of the same density and speed must reside within each bank and devices of the same type must reside within each group.

Memory Address Decoding

The memory array is divided into two separate addressable banks. The addressing for each bank is independently software-configurable through MULTIBUS II interconnect space and is on 64K byte boundaries. Software must insure that the address space of one bank does not overlap the address space of the other bank otherwise memory errors would result.

Built-In-Self-Test and Interconnect Subsystem

Self test and diagnostics have been built into the heart of the MULTIBUS II system. These confidence

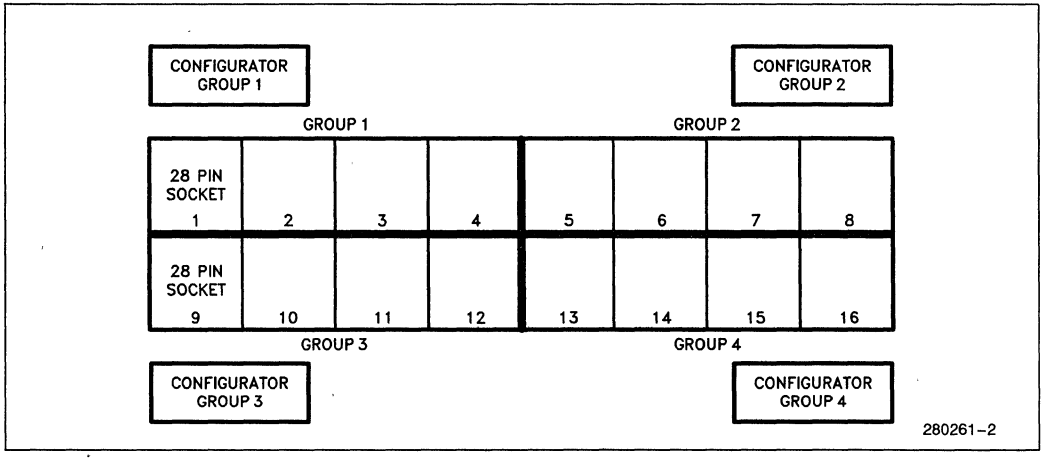


Figure 1. iSBC® MEM/601 Sixteen, 28-Pin Universal Site Memory Array

Table 1. Memory Devices Supported by the iSBC® MEM/601 Board

Type	2K x 8	4K x 8	8K x 8	16K x 8	32K x 8	64K x 8	
EPROM	2716	2732A	2764	27128	27256	27512	
ROM	Yes	Yes	Yes	Yes	Yes	Yes	
EEPROM	2817A	Yes	2864A	Yes	Yes	Yes	+5V Only
SRAM	TC 5516	Yes	TC 5565	Yes	TC 55257	Yes	NMOS and CMOS
Maximum Memory Capacity	32 KB	64 KB	128 KB	256 KB	512 KB	1 MB	

tests and diagnostics improve reliability and reduce manufacturing and maintenance costs. LED 1 (labeled BIST), is used to indicate the status of the built in self test. It is turned on when the BISTs start running and is turned off when the BISTs have successfully executed. Error information from the BISTs is recorded in the interconnect registers accessible to software. The built in self tests are peformed by the on-board microcontroller at power-up or on command.

The iSBC MEM/601 board interconnect subsystem consists of an 8751 microcontroller for Built-In-Self-Test (BIST), program storage, status, control registers, and interconnect control logic. The interconnect subsystem receives requests to interconnect space across either the PSB bus or the iLBX II bus depending on which interface is enabled. The interconnect subsystem is used by the software to configure the hardware.

Battery Backup

The iSBC MEM/601 board supports jumper selectable on-board or off board battery backup operation for CMOS SRAMs. Memory protection for the two memory banks can be supported with +5V from an off board power source or from the optional on board lithium battery. The memory content of the CMOS RAMs is protected during power-up and power-down by the protect signals from the PSB bus.

Parallel System Bus Interface

The PSB bus interface supports memory space and interconnect space and provides the capability of 8-, 16-, 24-, and 32-bit transfers. The PSB interface can be dynamically activated through the status register of the interconnect space under software control or can be jumper selectable. After a cold reset the PSB is enabled and the Local Bus Extension (iLBX II) bus is disabled.

Local Bus Extension Interface

The iSBC MEM/601 board provides 8-, 16-, 24-, and 32-bit transfers across the Local Bus Extension (iLBX II) interface. The iLBX II bus interface is enabled by the status register of the interconnect space and can therefore be dynamically changed through software. It is also jumper selectable. After a cold reset, the iLBX II interface is disabled. The PSB bus interface is always disabled when the iLBX II bus is enabled.

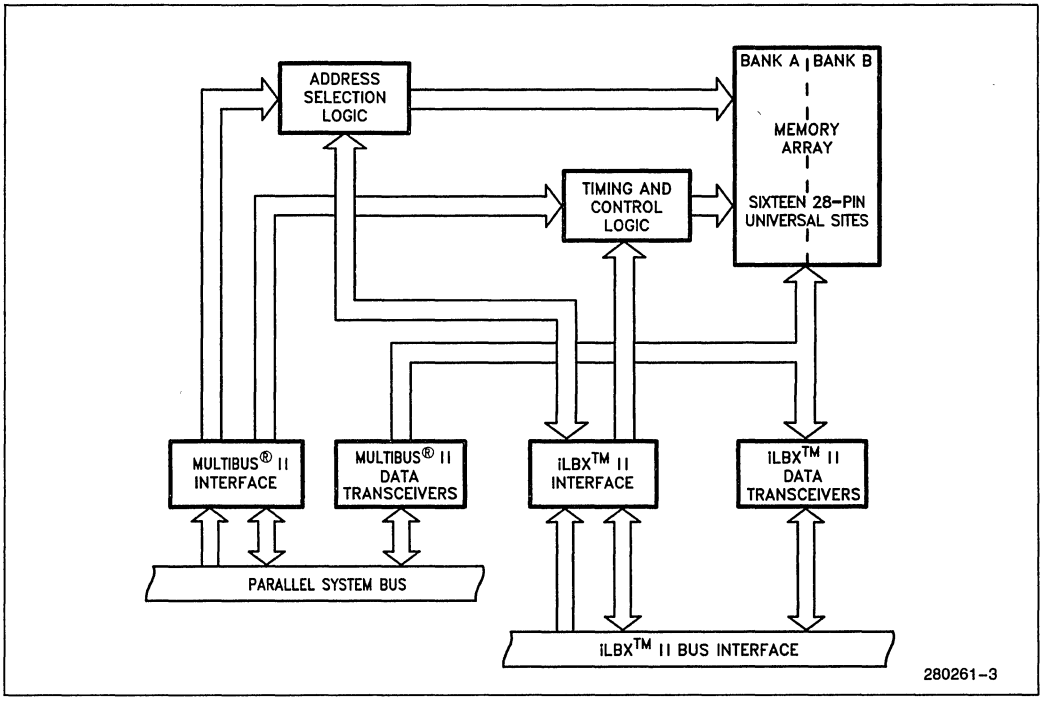


Figure 2. iSBC® MEM/601 Block Diagram

SPECIFICATIONS

Word Size 8-, 16-, 24-, and 32-bits

Memory Size

Sockets are provided for up to sixteen JEDEC compatible 28-pin devices which can provide up to 1.0M Byte of EPROM/ROM/SRAM memory.

Access Times

	PSB Bus	iLBX™ II Bus*
Read Cycle Without Replier Busy	300 ns	250 ns
Write Cycle Without Replier Busy	300 ns	250 ns
Read/Write with Agent Error	100 ns	10 ms

NOTES:
Access times are calculated without device speed included. True access times across either bus must include device access time and must be in 100 ns increments for the PSB bus. Above calculations assume 1 bus cycle. Refer to the iSBC MEM/601 Memory Board User's Guide for exact formula to determine access times for specific operating configurations.

*Access times across the iLBX II bus assumes an 8.0 MHz bus clock. The actual formula is as follows:

T = 2(C) + D where: T is iLBXII Bus access time
C is 1/f, f = iLBX II Bus clock speed
D is Device access time

Power Requirements

Current with 2764A EPROMs installed @ +5V: 4.5A

Current with 2864A EEPROMs installed @ +5V: 5.5A

At 3V and 300 mA hours lithium battery rating, the expected retention time for standard CMOS SRAM memories will be approximately 24–36 hours.

ENVIRONMENTAL REQUIREMENTS

Temperture: Inlet air at 200 LFM airflow over boards
Non-operating: −40°C to +70°C
Operating: 0°C to +55°C

Humidity:
Non-operating: 95% RH @ 55°C
Operating: 90% RH @ 55°C

Physical Dimensions

The iSBC MEM/601 board meets all MULTIBUS II mechanical specifications as presented in the MULTIBUS II specification (IEEE/ANSI 1296).

Double High Eurocard Form Factor

Depth: 220 mm (8.6 in.)
Height: 233 mm (9.2 in.)
Front Panel Width: 20 mm (0.784 in.)
Weight as shipped from factory: 543g (19 oz.)

Reference Manuals

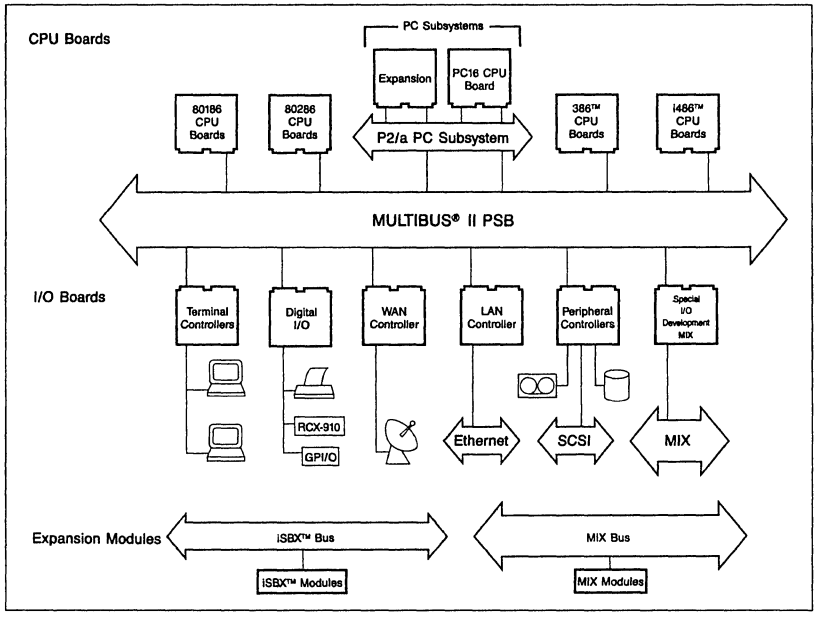
#149149—iSBC MEM/601 Memory Board User's Guide

Manuals may be ordered from any Intel Sales Representative, Distributor Office, or from the Intel Literature Department, 3065 Bowers Ave., Santa Clara, CA., 95051.

Ordering Information

Part Number	Descripton
SBCMEM601	MULTIBUS II Universal Site Memory Expansion Board

MULTIBUS® II I/O Products 3



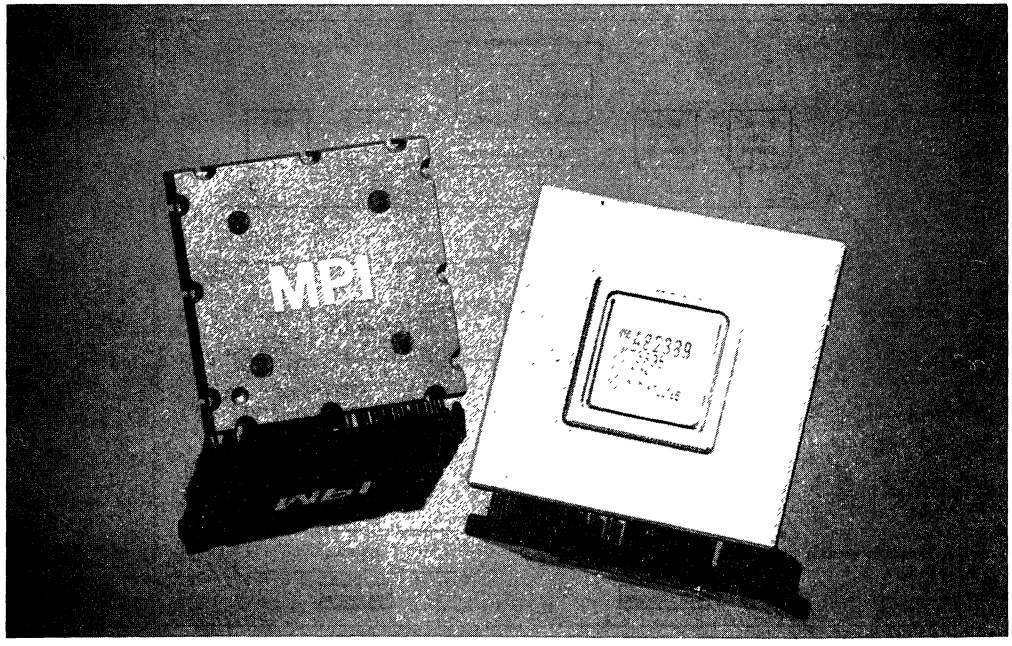
A SPECTRUM OF I/O FOR MULTIBUS®II DESIGNS

Intel's wide range of MULTIBUS®II I/O products is designed to help you easily complete your application. These boards include a variety of standard I/O products, such as terminal controllers, wide area network controllers, Ethernet controllers, SCSI peripheral controllers and Digital I/O boards. Intel also offers a choice of development methods for designing custom I/O boards. Now you can design low-cost, non-intelligent I/O boards based on the MULTIBUS II Peripheral Interface (MPI) silicon, or you can quickly and easily design high-performance, 386™ CPU-based I/O boards based on the Modular Interface eXtension (MIX) architecture. These products are described on the following pages.

CONTENTS

CONTENTS

September, 1989
Order Number 281009-001



MULTIBUS®II SILICON PRODUCTS

Intel's MULTIBUS II bus interface device product family has been created to aid the designer in interfacing application modules to the MULTIBUS II Parallel System Bus (PSB). The IEEE/ANSI 1296 specification defines a set of synchronous state machines which are clocked by a central bus clock. Adherence to this type of specification is simple and proveably correct. This architectural discipline has resulted in unprecedented compatibility between MULTIBUS II products from all vendors. Intel offers a silicon implementation of the IEEE/ANSI 1296 state machines with the 82389 Message Passing Coprocessor (MPC) component to provide a high capability interface targetted for intelligent board products. The MULTIBUS II Peripheral Interface (MPI) omits the block data transfer capabilities of the MPC and is targeted for non-intelligent, lower cost board products.

The MULTIBUS II Peripheral Interface (MPI) is a MULTIBUS II bus interface device providing PSB interface for non-intelligent I/O applications. The MPI is a replier in I/O and Interconnect space and can be implemented with a minimum of additional logic. The MPI supports the standardized signalling methods of the MULTIBUS II architecture with the ability to send and receive unsolicited messages (without data) as interrupts. An on-board CPU or microcontroller (such as an 8751) is not required for applications using the MPI. Interconnect space may be implemented using a single PAL or PROM.

The 82389 Message Passing Coprocessor (MPC) is the premier MULTIBUS II bus interface device for intelligent applications. It provides a complete, full function interface to the PSB, including arbitration, dual port memory recognition and the standardized signalling and data transfer methods of the MULTIBUS II architecture. The MPC component requires the support of an 8751-type microcontroller, and a DMA device is recommended for high performance data transfers. The MPC, in combination with the iSBC CSM/002 module or additional on-board logic, provides complete Central Services Module support for use in a slot 0 system location.

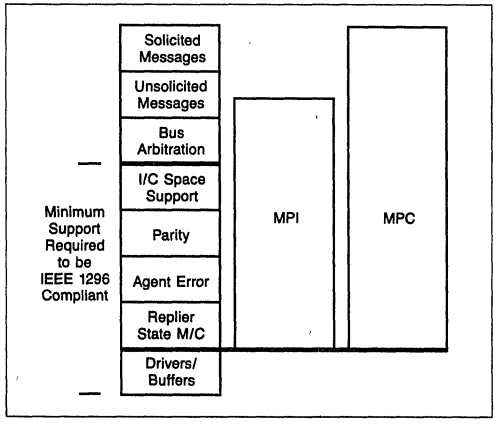
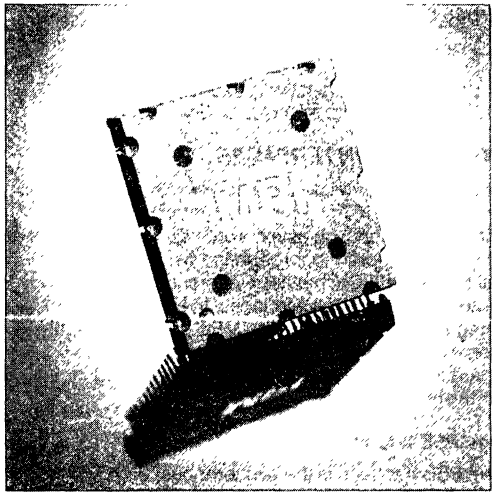


Figure 1: Comparing MPI and MPC Capability



MPI—MULTIBUS®II PERIPHERAL INTERFACE

The MULTIBUS II Peripheral Interface, MPI, is a single chip, "replier only" Parallel System Bus interface device. The MPI implements a IEEE/ANSI 1296 Replier State Machine, seen in Figure 2. All error conditions are monitored and generated if appropriate.

MPI FEATURES

- Replier in I/O Spac
  - 2 KBytes address on each board
  - 8/16 bus data width agents
- No application CPU required
- No support microcontroller required
- Supports up to 8 local interrupt sources
- Sends/Receives broadcast messages
- Sends/Receives unsolicited messages (without data)
  - Complete arbitration protocol
- Fair and High Priority modes are supported
- ANSI/IEEE 1296 compliant
- 124-pin plastic PGA package.

DESCRIPTION

The MPI component is a 16-bit integrated CMOS interface component compliant with the IEEE/ANSI 1296 standard and is compatible with other board products using the 82389 Message Passing Coprocessor. It supports data transfer in I/O space, as defined by the IEEE Specification. It is particularly suited to the design of low cost, non-intelligent I/O boards. Since the MPI component incorporates all the interface logic, except for five high current buffer drivers, it simplifies and accelerates I/O board design. The local interface is designed to provide a simple interface to I/O board components. The MPI also includes configuration registers which are programmed from the PSB to suit a variety of applications.

Interconnect Space

The MPI component supports IEEE/ANSI 1296 compliant interconnect space. Its registers are only accessed via the Parallel System Bus. The MPI includes the interface logic to support an external local memory device or PAL to implement most of the interconnect registers. Registers 34 through 38 are internal to the MPI.

I/O Space

The MPI component enables an I/O board to act as a replier in I/O space (as seen in figure 2). Board address space is programmed through interconnect space which allows multiple MPI-based boards to be used in a MULTIBUS II system with no jumpers. The width of the local I/O data bus can be 8 or 16 bits. Addresses and data for the local I/O is provided on a multiplexed bus.

Message Space

The MPI component enables an I/O board to send and receive interrupt packets, either in standard or Broadcast mode, without data. (Data transfer is carried out in I/O space.) Up to eight local interrupts may generate an interrupt packet onto the PSB; the highest priority interrupt level is encoded into this interrupt packet. The MPI entirely controls the access arbitration procedure for the PSB bus and the interrupt packet transfer. The MPI can receive interrupt packets from the PSB and uses them to generate a local interrupt signal.

The MPI interface supports parity signals when required and is capable of processing all error signals present on the bus.

MPI SPECIFICATIONS

Power Supply Voltage: 0 – 5 V
Operating Temperature: 0 – 70°C
Storage Temperature: −65 to +150°C
V_{CC} = 5.0 V + 10%

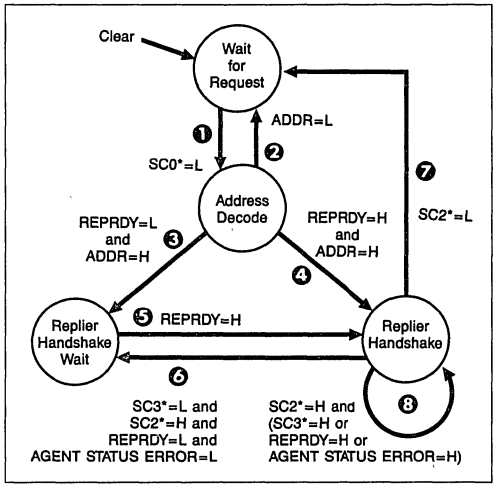
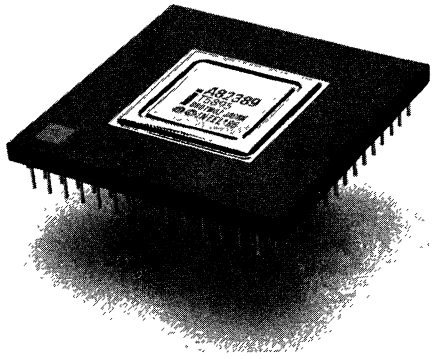


Figure 2: State-Flow Diagram for Replying Agents, from IEEE/ANSI 1296 Specification



82389—MULTIBUS®II MESSAGE PASSING COPROCESSOR

The 82389 MPC is a highly integrated VLSI CMOS device that maximizes the performance of a MULTIBUS II based multiprocessor system. The MPC implements the full message passing protocol as well as the functions (arbitration, transfer and exception cycle protocols) of the PSB bus interface control as defined in the IEEE/ANSI Standard 1296.

The 82389 MPC is designed to interface with an 32-, 16- or 8-bit processor. It provides support for message passing, interconnect space, memory, and I/O references on the PSB. In addition, the 82389 MPC component is designed to simplify implementation of dual port memory functions for those designs which will co-exist with the message passing communications protocol.

MPC FEATURES

- Single Chip Interface for the Parallel System Bus
- 1.5 u CMOS Technology
- 149-pin Ceramic PGA Package (15 x 15 Grid)
- Optimized for Real-Time Response (Maximum 900 ns for 32-byte Interrupt Packet)
- Processor Independent Interface to the Parallel System Bus
- Supports co-existance of dual port and message passing communication protocols
- Dual Buffer Input and Output DMA capabilities

MPC 82389 INTERFACES

The three primary interfaces to the MPC (PSB, Interface Host, CPU Interface and Interconnect Interface) all function asynchronously to one another. This is accomplished through the use of internal latches and FIFOs that allow references to occur simultaneously on all interfaces. In addition to the three primary interfaces, the MPC contains a Dual-Port interface which provides compatibility with shared memory system implementations and software.

The PSB Interface

The PSB Interface is the synchronous, communications pathway in a MULTIBUS II system. The PSB is a full 32-bit interface to other boards in the MULTIBUS II chassis. The PSB interface supports PSB arbitration, data transfer and error handling.

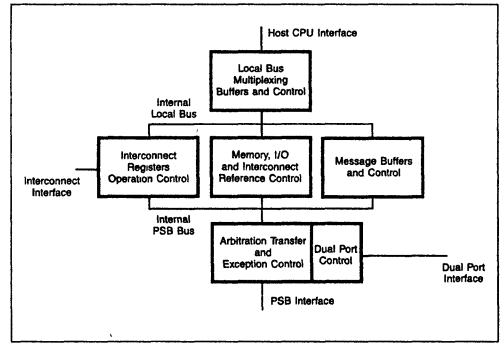


Figure 3: MPC Bus Interfaces

The Host CPU Interface

The Host CPU Interface is a set of addressable registers and ports that is the private pathway for the local microprocessor on the MULTIBUS II board. The Host CPU interface connects a 32-, 16- or 8-bit processor to the MPC. The Host CPU Interface supports direct references to memory, I/O, and interconnect address space on the PSB. The Host CPU Interface also supports DMA operations. The MULTIBUS II PSB and the MPC are defined to be processor independent.

The Interconnect Interface

The Interconnect Interface provides a path for added board functionality that is independent from the host CPU. The Interconnect Interface is an 8-bit communication interface which requires the MPC to be connected to a microcontroller or a simple state machine. A microcontroller will perform tasks such as board configuration at start-up and local diagnostics. All interconnect bus signals are asynchronous to the bus clock and to the local bus signals.

The Interconnect space of an agent is the only required bus space by the IEEE/ANSI 1296 specification and has a 512-byte register range.

The Dual Port Interface

The Dual Port interface supports shared memory accesses between agents on the PSB. The MPC contains programmable address recognizers and PSB cycle control. In order to fully implement dual-port memory, some additional dual-port memory controller logic is required.

MAJOR MPC OPERATIONS

The MPC standardizes the signalling and data transfer between multiple intelligent agents within a MULTIBUS II system. The traditional address spaces of memory and I/O were considered inadequate to accomplish this standardization task, so a new address space, called message space, was added. The movement of information in message space is called message passing. The MPC supports two types of messages: solicited and unsolicited.

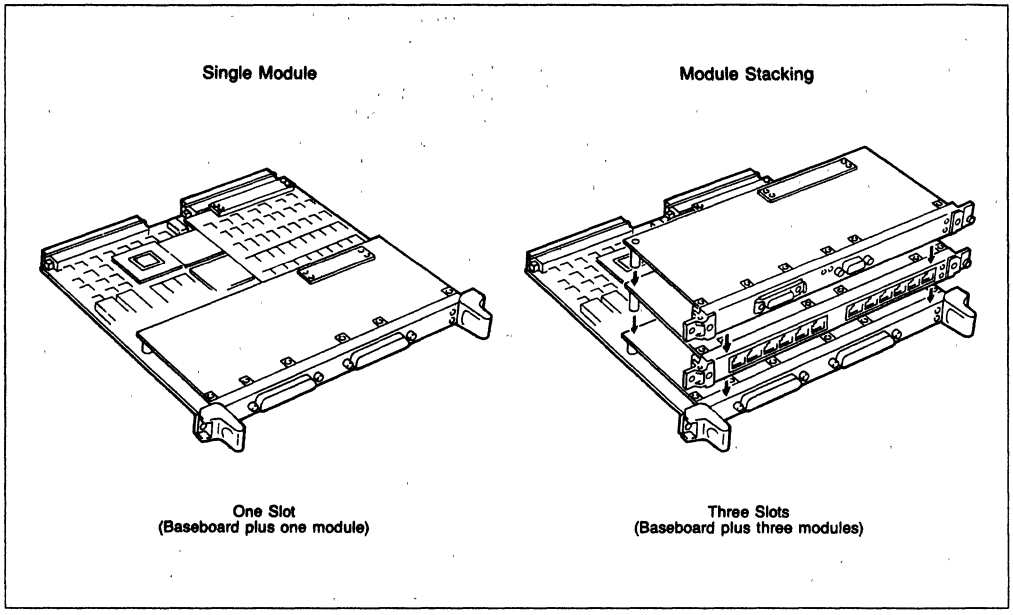
Solicited messages are used to transfer large amounts of data. Up to 16 MBytes (less 1 byte) of data can be transferred in a single solicited message transmission sequence. Solicited message transfers require the receiving agent to explicitly allocate a buffer. Data is packetized and reconstructed by the MPC to optimize PSB utilitzation and maintain deterministic performance. Buffer negotiation between sending and receiving agents is handled using unsolicited messages.

Unsolicited messages are short, fixed-length messages that can arrive unexpectedly. Unsolicited messages can be transmitted without explicit buffer allocation and without the cooperation of sending and receiving agents on the PSB. Unsolicited messages are often referred to as intelligent or virtual interrrupts, since they are used as a signaling mechanism between boards, replacing traditional system (hard-wired) interrupts and freeing the CPU from having to poll for information. In addition to interrupt generation, unsolicited messages allow for up to 28 bytes of user data.

MPC Specifications

Operating Temperature (under Bias). .10°C to +85°C
Storage Temperature. . . . . .65°C to +150°C
Voltage on Any Pin . . . . . .0.5V to V_{CC} +0.5V
Power Dissipation . . . . . .2.5 W

D.C. and A.C. Specifications are available in the 82389, Message Passing Coprocessor Datasheet. (See Ordering Information)



MIX HIGH-PERFORMANCE ARCHITECTURE FOR BUILDING TAILORED MULTIBUS®II I/O SOLUTIONS

Intel's Modular Interface eXtension (MIX) architecture provides a 32-bit asychronous bus technology designed for high-performance on-board I/O expansion. It is optimized for the 386™ microprocessor family and the MULTIBUS II system architecture. The MIX bus is implemented using a 130-contact surface mount connector and supports stacking of from one to three MIX I/O modules. The I/O module interface to the MIX bus is open, with specifications and documentation available from Intel for the development and implementation of MIX I/O modules.

The MIX architecture lets system designers make tradeoffs between the level of I/O performance, and the number and types of I/O functions. Using MIX modules, a range of I/O solutions can be implemented: from a single-module single-slot high performance I/O controller to a three-module three-slot I/O server subsystem. Now the system designer can select the right combination of I/O and CPU horsepower to effectively manage the system I/O requirements.

MIX ARCHITECTURE FEATURES

- Full compatibility with the MULTIBUS II (IEEE/ANSI 1296) Systems Architecture (MSA).
- Support for stacking of 1, 2, or 3 modules on the MIX baseboard. A MIX baseboard with a single MIX module fits into a single MULTIBUS II card slot.
- MIX bus data width of 32, 16, or 8 bits.
- Partitioning of I/O SBC architecture into a CPU & MULTIBUS II core and I/O module.
- Multimaster bus ownership for support of intelligent or non-intelligent MIX I/O modules.
- Support for like or unlike MIX module stacking with dynamic Built-in Self Test (BIST) and interconnect capabilities.
- Complete documentation available for building MIX I/O modules.

MULTIBUS®II COMPATIBILITY

The MULTIBUS II Systems Architecture is optimized for efficiently interconnecting multiple intelligent microprocessor-based subsystems. The MULTIBUS II architecture also accommodates many types of local bus extensions for solving local communication requirements within subsystems.

The MULTIBUS II Parallel System Bus (PSB) provides the main communication backbone for the total system, while other elements of the architecture solve the system integration issues such as initialization, diagnostics, and standardized subsystem to subsystem signalling and data transfer.

The MIX architecture adds a subsystem bus technology that provides a solution for high performance I/O and brings the capability of a high performance I/O server subsystem to the MULTIBUS II architecture.

PHYSICAL DECOUPLING OF CPU FROM I/O

MIX uses a baseboard plus modules approach to physically decouple the CPU technology of the baseboard from the I/O technology of the module. This decoupling has two benefits for I/O design. First, it allows the CPU and I/O technology to evolve independently so that new technology can be more easily incorporated into system designs. Second, it allows a baseboard to change personality by adding or substituting I/O modules. This provides I/O flexibility while preserving the software investment.

CONNECTING CPU AND I/O VIA A HIGH PERFORMANCE BUS INTERFACE

The MIX architecture provides the high performance bus interface for coupling the CPU baseboard with the I/O modules. Elements of the MIX architecture and bus interface include:

Signal Set

The MIX bus consists of 130 signal, power, and ground connections. There are two types of signals: dedicated and bussed. Dedicated signals belong to specific modules in the MIX stack, while bussed signals are shared by all modules.

Address Capability

The MIX bus supports the full 4 gigabyte physical addressing capability of the 386™ microprocessor and other compatible microprocessors.

Data Paths

MIX supports 8, 16, and 32 bit physical data paths on MIX modules. The MIX baseboard data path is 32 bits.

MIX Bus Transfers

The baseboard can perform memory, I/O, and DMA transfers on the MIX bus. The baseboard can also perform a bus vectored interrupt transfer cycle. Bus master modules can perform memory transfers with the baseboard memory.

Arbitration

The MIX bus uses a simple round robin arbitration scheme between the baseboard and master modules to insure that all modules and the baseboard have guaranteed access times to shared baseboard memory and have a guaranteed percentage of the shared memory bandwidth.

Interrupts

Each MIX module has one dedicated interrupt line. Each module also has an option line that can be used as an interrupt line.

DMA

The MIX bus supports DMA transfers between modules and the baseboard memory. Both single-cycle (fly-by) and two-cycle DMA transfers are supported.

Configuration Support

MIX configuration support has been designed to be compatible with the MULTIBUS II interconnect space architecture. MIX modules are viewed as baseboard functions by agents on the parallel system bus. The baseboard microcontroller reads the interconnect information stored in the EEPROM of each module present in the MIX stack to build the function record in baseboard interconnect space.

Built-in Self Test (BIST) Support

MIX provides the capability for BIST code resident in module EPROM to be downloaded and executed as an extension of the baseboard BIST.

OPEN INTERFACE FOR I/O MODULE DEVELOPMENT

MIX provides an excellent platform for building MULTIBUS II I/O solutions. A complete set of manuals, design specifications and design examples for building MIX I/O modules is available from Intel.

MIX BUS INTERFACE SPECIFICATIONS

General

Bus Type:	Asynchronous
Theoretical Bandwidth:	22 MByte/s
Typical Bandwidth:	10 MByte/s
Bus Overhead:	7% Xchange/Refresh
Interrupt Sources:	Any Module
Bus Vector Support:	Yes
Arbitration Scheme:	Fairness (Rd-Robin)
Module Maximum:	3 (master or slave)
Length of Bus Hold:	8 microsec (typical)
Flag Byte Support:	Yes

Lines

130 total signal, power and ground lines:

Number	**Functional Group**
37	Address
32	Data
7	Transfer Control
6	Arbitration
3	Interrupt
6	DMA
3	Option
7	Configuration
9	+5 VDC
13	GND
2	+12 VDC
2	−12 VDC
3	Reserved

Baseboard Address Range

Memory	4 Gigabytes
I/O	64 Kilobytes

Module Address Range

Memory	256 Megabytes
I/O	1 Kilobyte

Data Path

8, 16, and 32 bit

MIX Connector:

Connector Type:	Surface Mount
Connector Pads:	130

MIX Expansion Module:

Module Height:	8.9 inches
Module Depth:	3.75 inches
Module Area:	33 square inches
Max Configuration:	1 Baseboard 3 Modules

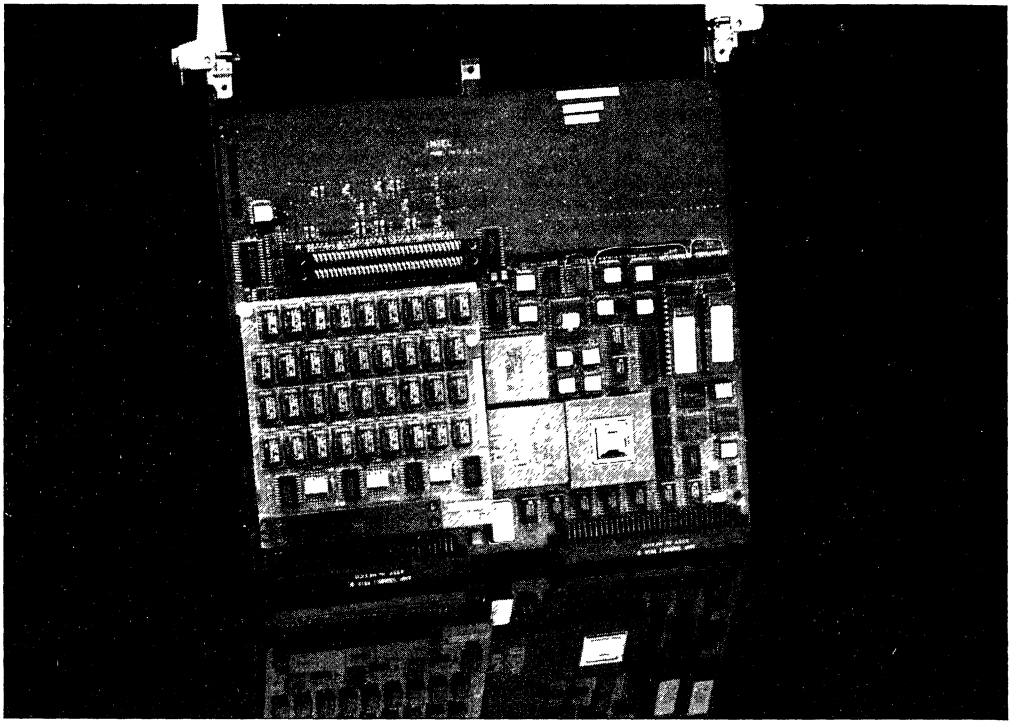
MIX Bus Power Limits

Voltage (VDC)	**Total Current (Amps)**
+ 5 (+5%, −2%)	9.0
+12 (+5%, −5%)	1.5
−12 (+5%, −5%)	1.5

Module Power Limits

Nominal Voltage (VDC)	**Max Current per Module (Amps)**
+ 5	3.0
+12	0.5
−12	0.5

Thermal Limit (all sources):
20 Watts max per module



A 386™ CPU-BASED CORE FOR BUILDING INTELLIGENT MULTIBUS®II I/O SOLUTIONS

The Intel Modular Interface eXtension (MIX) I/O Platform Family provides the 386™ microprocessor core, hardware development modules, and documentation needed to build high performance custom I/O solutions for MULTIBUS II systems

The MIX 386/020 Baseboard combines a 386™ microprocessor, Advanced Direct Memory Access (ADMA) controller, and Message Passing Coprocessor (MPC) to provide a significant amount of silicon muscle for handling I/O processing. In addition, with 1 megabyte of on-board DRAM and up to 16 megabytes of DRAM expansion, the baseboard provides enough memory to accommodate on-board execution of large amounts of I/O software. Add to that the I/O expansion capabilities of the MIX interface and you have a versatile, high-performance engine for handling I/O processing.

MIX 386/020 BASEBOARD FEATURES

- 386™ microprocessor operating at 20 MHz.
- 1 megabyte of on-board fast page mode DRAM with parity checking.
- Memory expansion up to an additional 16 megabytes of fast page mode DRAM with parity checking.
- 82258 ADMA for handling data transfers between the baseboard DRAM and the MPC and also between the baseboard DRAM and the MIX modules. Two cycle, fly-by, and burst mode transfers are supported.
- Full 32-bit MULTIBUS II Parallel System Bus interface provided by the 82389 Message Passing Coprocessor (MPC)
- MULTIBUS II systems architecture compatible firmware including Built-In Self-Test (BIST) code for the baseboard plus the capability to download and execute BIST code for the attached MIX modules.
- MIX bus interface capable of supporting one, two or three attached MIX I/O modules.

386™ MICROPROCESSOR CORE

The MIX 386/020 Baseboard obtains its I/O processing power from a 386™ microprocessor operating at 20 MHz. All the programming features of the 386™ microprocessor are supported.

The Protected Virtual Address Mode (PVAM) of the 386™ microprocessor provides the MIX 386/020 Baseboard with a full 4 gigabytes of addressability. The top gigabyte is used for baseboard EPROM and MIX Module memory access. The lower 3 gigabytes are divided between baseboard DRAM and Parallel System Bus access. PVAM operation also provides support for the 386™ microprocessor protection, virtual memory and paging mechanisms.

In addition, the 386™ microprocessor has a self-test capability which is utilized in the board's power up BIST testing. This function can be disabled via a board jumper option.

FROM 1 TO 17 MEGABYTES OF MEMORY

The DRAM block of the baseboard consists of an asynchronous fast page mode DRAM controller, address multiplexor, data transceivers with parity detection and generation, 1 megabyte of baseboard DRAM, and DRAM expansion using the MMxx interface. The MIX 386/020 is designed to accept one or two MMxx DRAM modules. A total of 17 megabytes of DRAM memory is obtained with the installation of two iSBC MM08FP memory modules. Baseboard DRAM (both on-board and on MMxx modules) is directly accessible to bus masters on the MIX bus.

Byte parity protection is used for DRAM error checking on the board. The transceivers generate parity for memory write cycles and check parity for memory read cycles.

The CPU block of the baseboard requires fast page mode DRAMS, which provide zero wait state performance for code prefetching by the 386™ microprocessor, and one wait state performance for all other DRAM accesses.

ADMA FOR FAST MEMORY TRANSFERS

The MIX 386/020 Baseboard uses the 82258 ADMA operating at 8 MHz, DMA Address Generator (DAG) gate array for 32-bit, 4 gigabyte addressing, and fast page mode DRAM control logic to handle high speed data transfers between baseboard DRAM and both the MPC and MIX modules.

The 82258 ADMA provides four independent channels for DMA service; two channels are used to service the MIX stack and two are used to service the Message Passing Coprocessor (MPC). Three transfer modes are supported: burst, single-cycle (fly-by), and two cycle. To the MIX bus, burst mode transfers look like single cycle transfers. Burst mode transfers between the baseboard DRAM and the MPC or MIX modules have a maximum transfer rate of 14.2 megabytes per second.

MULTIBUS®II SYSTEMS ARCHITECTURE SUPPORT

The MIX 386/020 Baseboard utilizes the 82389 Message Passing Coprocessor (MPC) to provide a full 32 bit interface to the MULTIBUS II Parallel System Bus. Firmware is also provided that contains baseboard Built-In Self Tests (BIST) and Initialization and Diagnostics eXecutive (IDX) code. Also included is the capability to download BIST code from MIX module EPROM and execute the code to test the modules in the MIX stack.

A POWERFUL ENGINE FOR I/O PROCESSING

The MIX 386/020 supports a stack of 1, 2 or 3 MIX I/O modules. The MIX bus supports 32, 16 and 8 bit data transfers and allows MIX modules to be either masters or slaves on the MIX bus.

A custom gate array device on the baseboard controls the baseboard interface to the MIX bus and implements the MIX bus arbitration logic. The gate array implements the standard MIX round-robin arbitration algorithm which provides guaranteed access to the MIX bus by the baseboard and module bus masters. The capability to modify certain arbitration parameters is also provided.

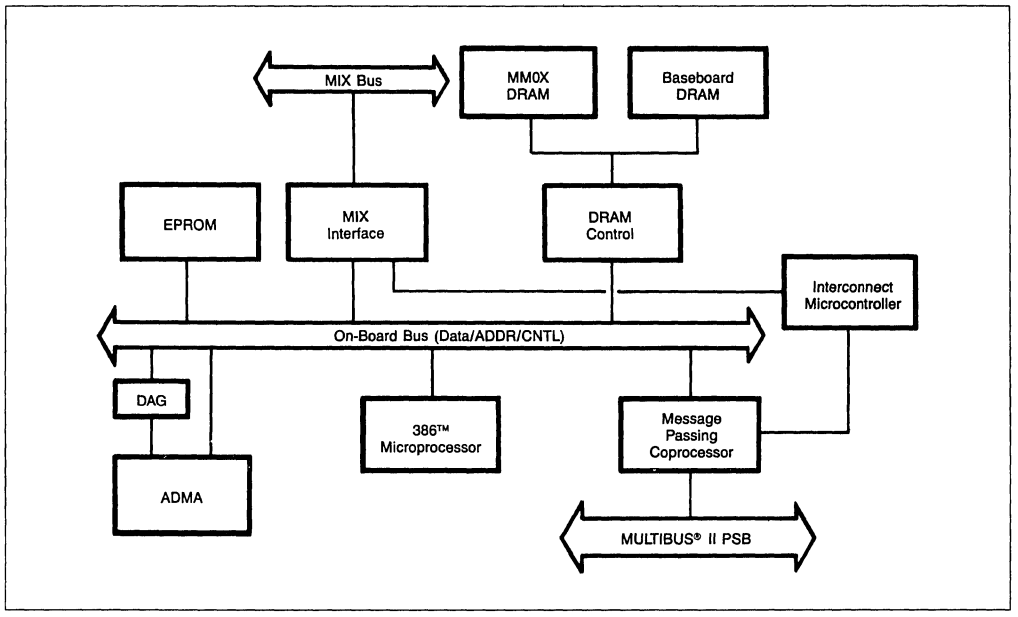


Figure 4: Block Diagram of MIX 386/020 Baseboard

MIX 386/020 BASEBOARD SPECIFICATIONS

Clock Rates

386™DX Microprocessor 20 MHz
82258 ADMA 10 MHz
8751 Microcontroller 12 MHz
82C54 Timer (Programmable)

EPROM Memory

Two 32-pin Sites

DRAM Memory

1M byte installed on the baseboard.
Memory may be increased by installing up to two iSBC MM0XFP Memory Expansion Modules, up to a total of 17 MB. Separate versions are orderable with 1, 2 or 5M bytes already installed.

Interrupt Capabilities

14 programmable interrupts

Interfaces

- P1, Full PSB.
- P2, power only.
- MMxx local memory expansion.
- MIX

Device Drivers

Check the latest release of the following operating systems for details:
iRMX II Operating System
UNIX* System V/386 Operating System

Physical Characteristics

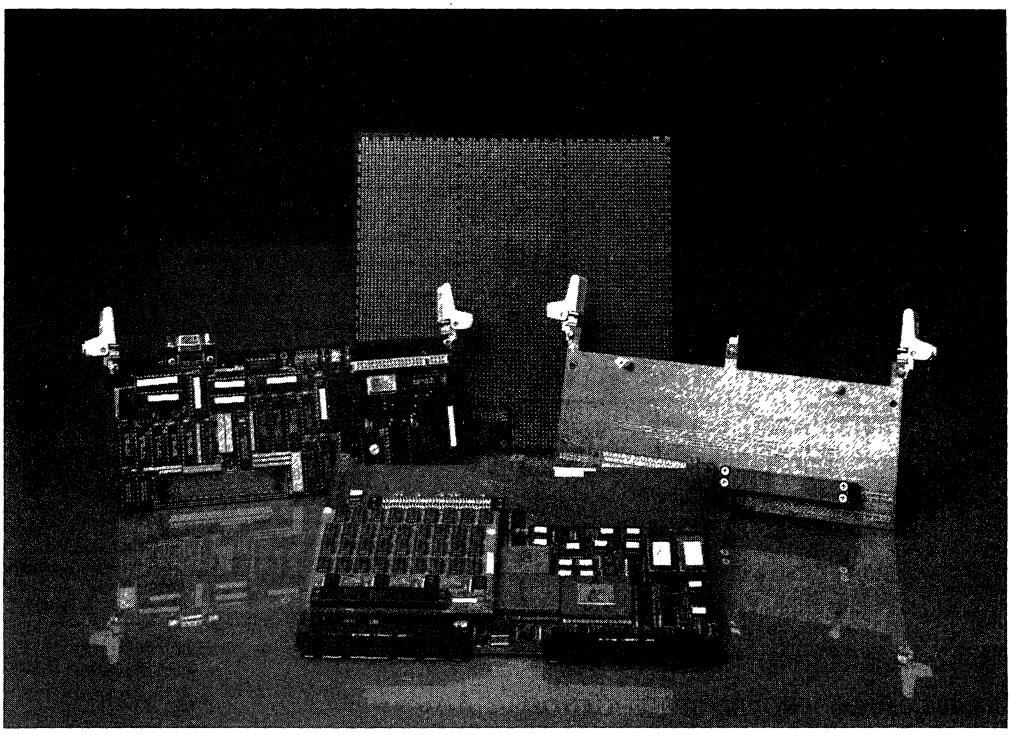
Standard MULTIBUS II board.

Power Requirements

Maximum values are at nominal voltage plus 5% and at an ambient temperature of 0 degrees C.

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+ 5	6.0	31.5
+12	0.0	0.0
−12	0.0	0.0

Note: Power requirements do not include installed MIX I/O modules.



A KIT FOR DEVELOPING MIX I/O MODULES

A requirement for many MULTIBUS II designs is the incorporation of special I/O. This is I/O that may be unique to the application or require an I/O controller not readily available in the market. To date, such a requirement typically would have necessitated the non-trivial task of designing a dedicated MULTIBUS II I/O controller. With the Modular Interface eXtension (MIX) interface and the MIX Module Development Kit, the task of developing a MULTIBUS II I/O solution is simplified.

The MIX bus is a straight-forward, well documented interface for developing MIX I/O modules. In addition, the MIX Module Development Kit provides all the hardware, documentation and optional consulting support needed to easily develop your own MIX I/O module that stacks on the MIX baseboard. This combination greatly simplifies the task of developing your complete I/O solution for MULTIBUS II systems.

MIX MODULE DEVELOPMENT KIT FEATURES

- MIX 386/020 Baseboard used as both the development vehicle and the platform for the final production module
- A set of MIX development modules to facilitate the MIX module development process, including a test module, breadboard module, and debug module.
- Complete documentation set providing all the information needed to develop a MIX module
- Optional Intel Field Systems Engineer consulting to help you better focus your development team and save development time.

MIX MODULE DEVELOPMENT KIT CONTENTS:

MIX 386/020 Baseboard

Developing custom I/O modules using the MIX 386/020 baseboard, provides an implementation method that offers both quick time-to-market and reduced risk. Time-to-market is faster because two thirds of the total I/O controller design is already provided by the baseboard. Risk is reduced, because with the MIX 386/020 baseboard, Intel has already solved the problems of designing the CPU core and providing the interface to the MULTIBUS II Parallel System Bus.

Documentation Package

A complete set of manuals, specifications and design examples for building a MIX module. This documentation package provides all the information you need to successfully develop a MIX module.

Optional Field Systems Engineer Consulting

Intel Systems Engineers have the experience and engineering expertise that can save you valuable development time. Consulting support for MIX module hardware, software, or firmware development is available as an optional component of the Module Development Kit.

MIX Development Modules

All the MIX development modules are designed to stack on top of the MIX baseboard or another MIX module.

- MIX MOD1 Test Module
  Used for testing of MIX module hardware and software designs. The test module contains a serial interface and an iSBX™ connector for communicating with the MIX baseboard.

- MIX MOD2 Breadboard Module
  Used for wire-wrapping and building a prototype module design. The breadboard module provides three separate wire-wrap areas, each surrounded by power and ground connections. In addition, stake pins are provided which give access to the signals from the MIX interface.

- MIX MOD3 Debug Module
  Used to mount a MIX module with its component side up (that is, flipped over from its normal mounting orientation). This allows access to the module's components for easy probe connection and debug.

The MIX Module Development Kit Makes MULTIBUS®II I/O Development Easy

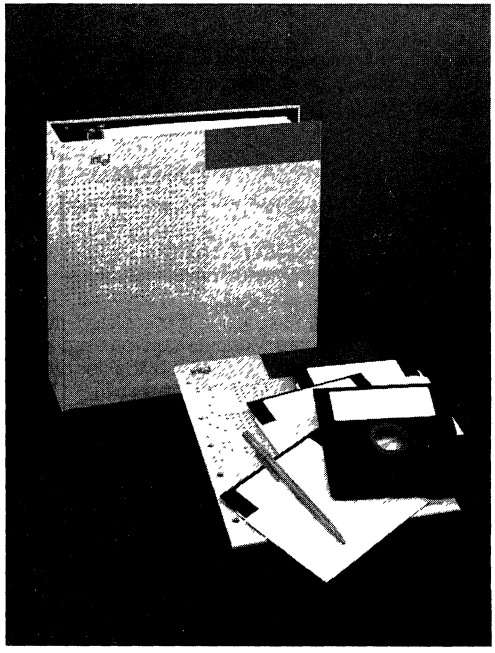
The elements of the MIX Module Development Kit help to facilitate your engineering teams development process.

To develop a MIX module, your engineers would start out as usual with the hardware team designing the hardware logic on paper or on a CAD system and the software team developing the preliminary design for the module firmware and application software. Once the preliminary hardware design is completed, the MIX MOD2 Breadboard Module would be used to build a prototype wire-wrap version of the design. Since the Breadboard Module mounts on the baseboard MIX connector, module interaction with the baseboard can be easily checked out.

With a wire-wrap version of the module, the software team can use the MIX MOD1 Test Module to test out the early application software interaction with the baseboard and the wire-wrap design. Because of the stacking capability of MIX, the Test Module and Breadboard Module can both be mounted to the baseboard in a stacked configuration.

Once the breadboard design has been checked out, the engineering team would build the production version of the module. Using the MIX MOD3 Debug Module, the final module can be mounted on the baseboard with its components side up for easy probe access and hardware debug.

Finally, using the Test Module and the Debug Module in a stacked configuration, the engineering team can debug both the final hardware and software to complete the design.



FIRMWARE DEVELOPMENT PACKAGE

The MULTIBUS II Firmware Development Package (FDP) makes the benefits of MSA (MULTIBUS II Systems Architecture) available at a fraction of the cost of developing a proprietary implementation using the MULTIBUS II specifications. FDP enables developers to fully realize a multiple processor design by providing a standard solution for system initialization and bootload. In addition, a standard Built In Self Test (BIST) architecture is provided to offer several levels of diagnostic compatibility. Developing MSA via FDP will reduce implementation time and simplify adherence to the MSA specification.

FIRMWARE DEVELOPMENT PACKAGE FEATURES:

- Source Code and Binary files for: Master Test Handler, Console Controller, Bootstrap Loader, Initialization & Diagnostic Executive, and Core Function Set.
- Complete Documentation: Overview manual, Specifications, BIST Writer's Guide
- DOS generation environment
- Available in C language
- Distributed via DOS diskettes, including make files compatible with PolyMake*
- Designed to facilitate customization

*PolyMake is a trademark of Polytron Corp.

FIRMWARE DEVELOPMENT PACKAGE FEATURES

OVERVIEW MANUAL

The FDP Overview Manual details the architecture of MSA firmware, explaining the rationale for module partitioning as well as the capabilities and limitations of the various modules.

SOURCE FILES

FDP is a source product. All files necessary to duplicate the MSA firmware functions for diagnostics, initialization, and booting are included in the package.

BINARY FILES

Binary files of each module are included and may make the generation of unmodified modules unnecessary. These object modules also serve as references in validating the development environment.

BUILT IN SELF TEST (BIST) EXAMPLES

Examples of actual BIST code commonly used on Intel MULTIBUS II hardware are included for reference.

BIST WRITER'S GUIDE

A BIST Writer's Guide is included to aid the process of learning the BIST interfaces and to show the typical organization of BIST code on Intel hardware. Using the guide, the first time BIST writer will quickly come up to speed. Master Test Handler (MTH), Local Test Handler (LTH), and power-up Test Handler (PTH) interfaces to the Initialization and Diagnostics Executive (IDX) are covered.

SPECIFICATIONS PACKAGE

Detailed external specifications for each module are included. These documents are suitable for implementation purposes and were actually used in the development of base FDP firmware.

AVAILABLE IN C LANGUAGE

Recognizing the need for portability and the popularity of C, the FDP software is available in the C language.

DISTRIBUTED VIA DOS DISKETTES

Covering the most popular development environment, the distribution media affords easy portability.

INCLUDES MAKE FILES FOR GENERATION

FDP includes a file that may be used with the PolyMake* utility to greatly ease the regeneration process. Only files having modifications are recompiled, which eases the generation process and significantly reduces the time required.

DESIGNED TO FACILITATE CUSTOMIZATION

FDP is an open product, which is partitioned and organized to facilitate any changes and extensions necessary to support your hardware.

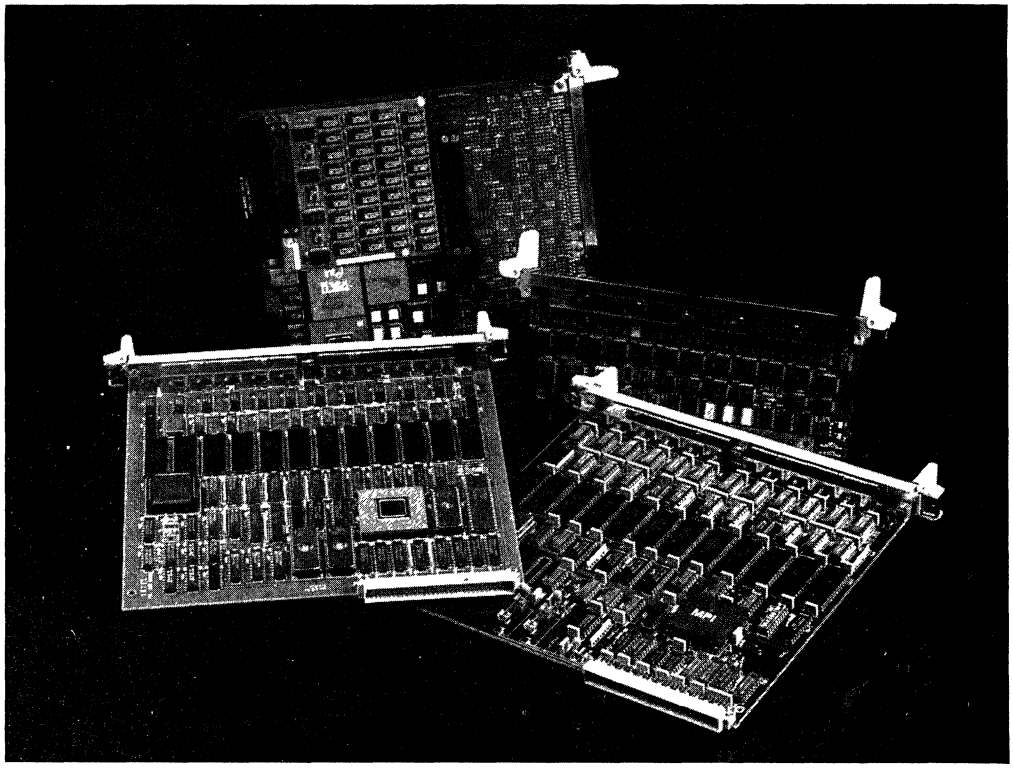
HARDWARE REQUIREMENTS FOR DEVELOPMENT

Any open system that supports MSA, e.g., the Intel System 520, will provide an adequate hardware environment for firmware validation.

PACKAGE CONTENTS

The product package contains source and object for the modules called out above on DOS diskette, external specifications, and an overview manual.

*PolyMake is a trademark of Polytron Corp



FAMILY OF MULTIBUS®II TERMINAL CONTROLLERS

Asynchronous Terminal Controllers must address such application requirements as terminal access, remote modem access and computer to computer communication while meeting price and performance criteria. The Intel MULTIBUS II terminal controller family addresses these application needs with three boards: iSBC MPI/450, iSBC 186/450 and MIX 386/450. These boards are application compatible, offering a range of price/performance options. The iSBC MPI/450 is a non-intelligent I/O board that provides asynchronous serial I/O port extensions to a host CPU board. The iSBC 186/450 is an intelligent dedicated terminal controller that efficiently performs terminal functions within the system. The MIX 386/450 is a high performance terminal controller, utilizing the power of 386™ CPU performance and the flexibility of Modular Interface eXtension (MIX) stacking.

All three boards have been designed stressing compatibility across the product line. In respect to hardware, all physical connections, eg. cabling and connectors, are interchangeable between the boards. The result is that the three boards follow the same front panel design with connectors, each clearly marked with the individual product name. The use of common components result in protocol compatibility. This compatibility is extended into the software structure as well. Standard software support for all three boards maintains a consistent application interface. By adhering to these hardware and software standards, the boards achieve a high degree of interoperability.

STANDARD TERMINAL CONTROLLER FEATURES

- Full duplex asynchronous transmission using the 82510 UART
- 12 ports per board, RS232C compatible
- 8 signal support, RJ45 (Phone Jack Style) shielded connectors
- Performance ranging from 110 baud to 19.2k baud
- Asynchronous Terminal Control Software (ATCS) for interrupt processing, character handling and modem support

MULTI-TERMINAL ACCESS

Twelve serial I/O ports are provided by each board. Each port is based on the 82510 USART component and supports full duplex asynchronous transmissions An on-chip baud rate generator allows for independent baud rates on each channel For applications requiring more than 12 ports, the number of ports can be expanded in three ways by adding more MPI boards (iSBC MPI/450), by adding more intelligent boards (iSBC 186/450), or by mounting additional MIX450 modules onto the MIX baseboard. The customer may choose their configuration based on price and performance requirements of their application.

EASE OF CABLING

Intel's Terminal Controllers utilize the RJ-45 "phone jack" style connector, which provides shielding and lock in mating. Individual connectors are directly plugged into the front panel mount, hence a break out box is not required. Changing terminal configurations is quick and easy with this versatile connection and the identical front panel mounts on the boards. Cables are available through commercial vendors in both shielded and unshielded specifications. Intel recommends shielded cables for application use.

MODEM SUPPORT

Each channel provides for 8 signal support. Software handshaking (DTR,RTS and CTS), Carrier Detect (DCD), Ring Indicator (RI), Data Relay (RXD and TXD) and Signal Ground (SG) are supported. This support allows for access to remote dial up modems and computer to computer communications.

CONSISTENT APPLICATION INTERFACE

Asynchronous Terminal Control Software (ATCS) provides a standard application interface for all Intel Terminal Controllers ATCS optimizes the interrupt processing and character handling on an intelligent terminal controller board, offloading this task from the application CPU. ATCS supports full duplex and provides such features as support for multiple hosts, dynamic line switching, and modem support. ATCS achieves these features and high performance by providing input and output buffers of 2K per line (port). These buffers increase serial data throughput on output and allow input bursts to be absorbed. Dependent on the CPU, this capability can result in simultaneous input and output up to 19.2k baud rates. ATCS code is designed to support up to 36 channels per server and multiple ATCS servers may reside in the system

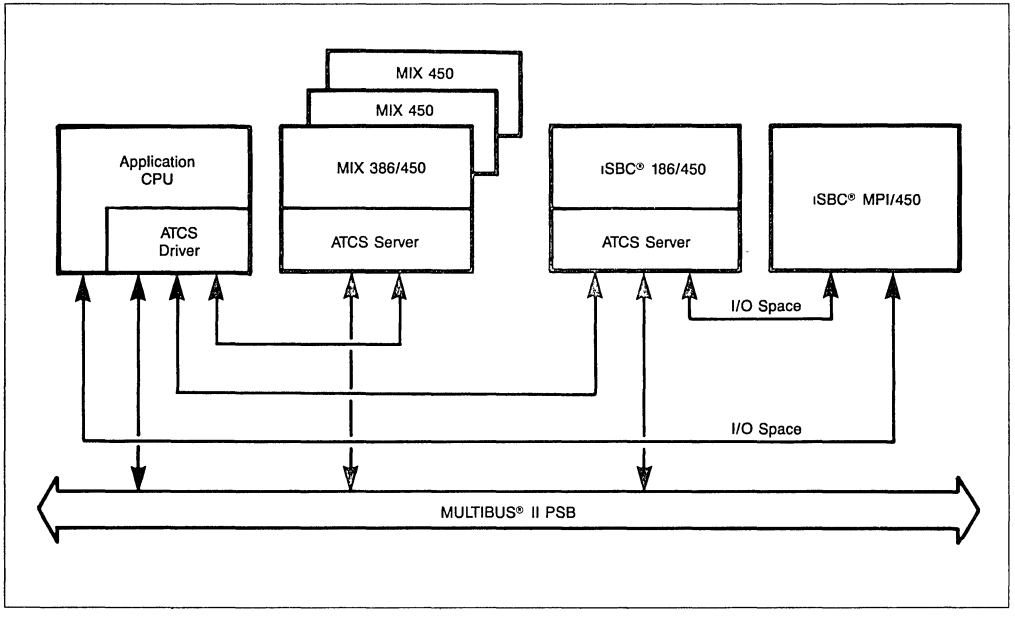
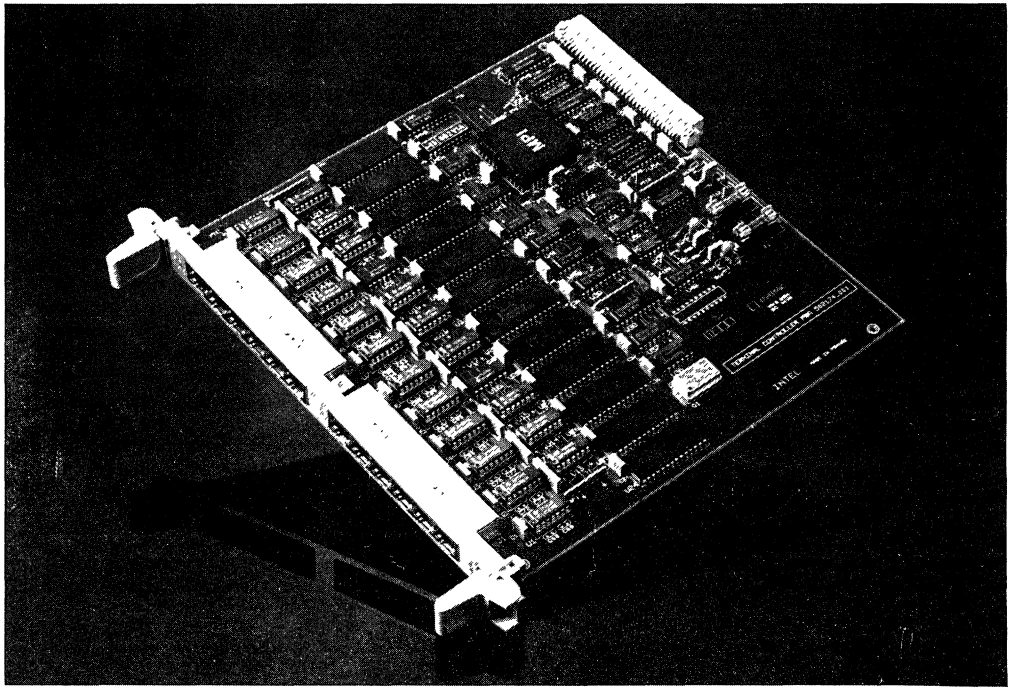


Figure 5: Terminal Controller Configuration Example.



iSBC®MPI/450 TERMINAL CONTROLLER

The iSBC MPI/450 is a non-intelligent 12 channel, RS 232-C compatible, asynchronous terminal controller. The iSBC MPI/450 utilizes the MULTIBUS II Peripheral Interface (MPI) component to add additional I/O capability to an application CPU.

iSBC®MPI/450 FEATURES

- Extension of application CPU by providing offboard asynchronous ports using the 82510 UART
- Slave MULTIBUS II Parallel System Interface provided by the MULTIBUS II Peripheral Interface (MPI)
- 12 ports per board, RS232C compatible
- 8 signal support, RJ45 (Phone Jack Style) shielded connectors
- Performance dependent upon application CPU bandwidth

EXTENSION OF APPLICATION CPU

The iSBC MPI/450 provides 12 offboard asynchronous channels to the application CPU, allowing low cost ports to be easily added to the system. The application CPU accesses the MPI ports via the PSB I/O space, therefore any intelligent board may host the iSBC MPI/450. The number of MPI boards that can be supported is dependent upon the host CPU bandwidth and application requirements.

MPI FEATURES

The MPI component provides the iSBC MPI/450 with the capability to generate unsolicited messages without data. This feature allows the host CPU boards to interact with the iSBC MPI/450 when prompted by a message rather than requiring continually polling. An interrupt register is provided for servicing the 82510 USARTs, whether a polling or message technique is used.

iSBC®MPI/450 CONTROLLER SPECIFICATIONS

Interfaces

P1	Slave PSB
Serial	12 channels, RS232C, 8-pin RJ-45 connectors, 82510 Controller

Device Drivers

Check the latest release of the following operating systems for details:
iRMX II Operating System
UNIX* System V/386 Operating System

Physical Characteristics

Standard MULTIBUS II board

Power Requirements

(Excluding user-installed memory devices)

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+ 5	1.55	7.75
+12	.15	1.8
−12	.15	1.8

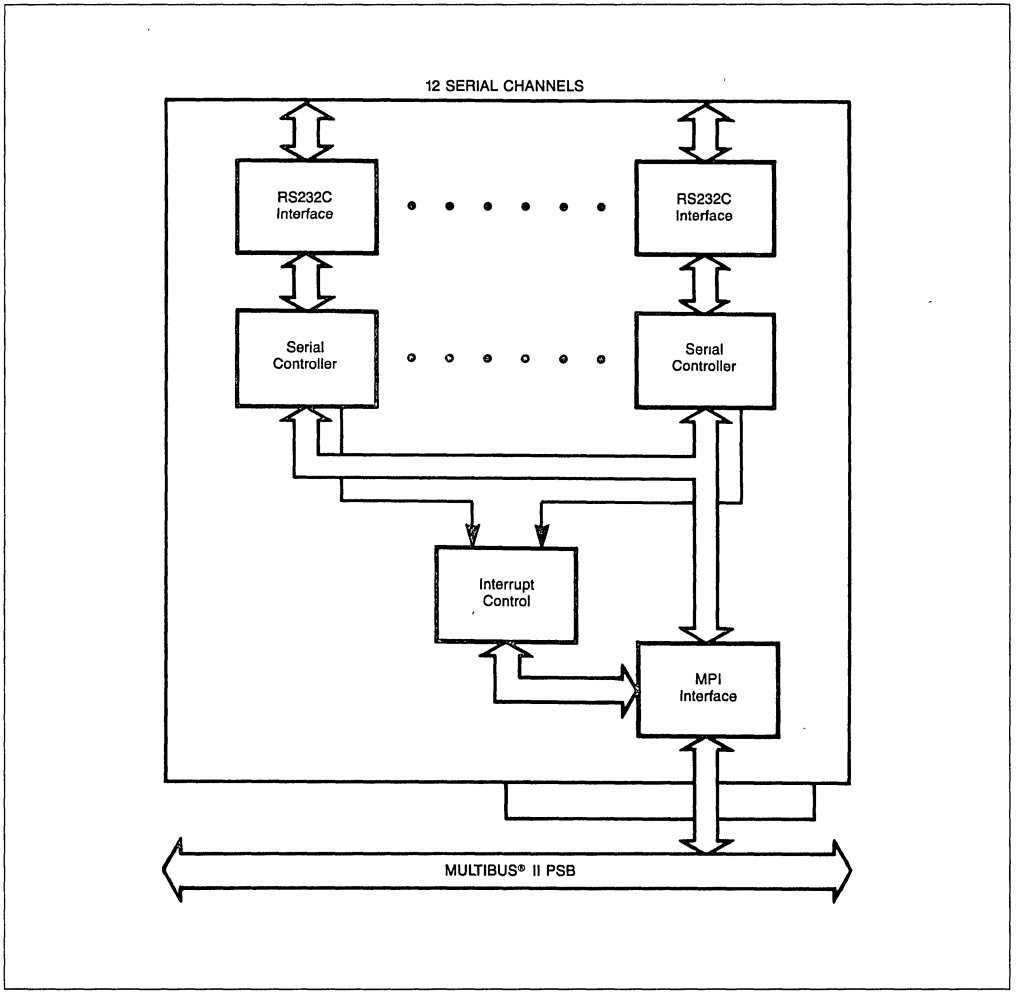
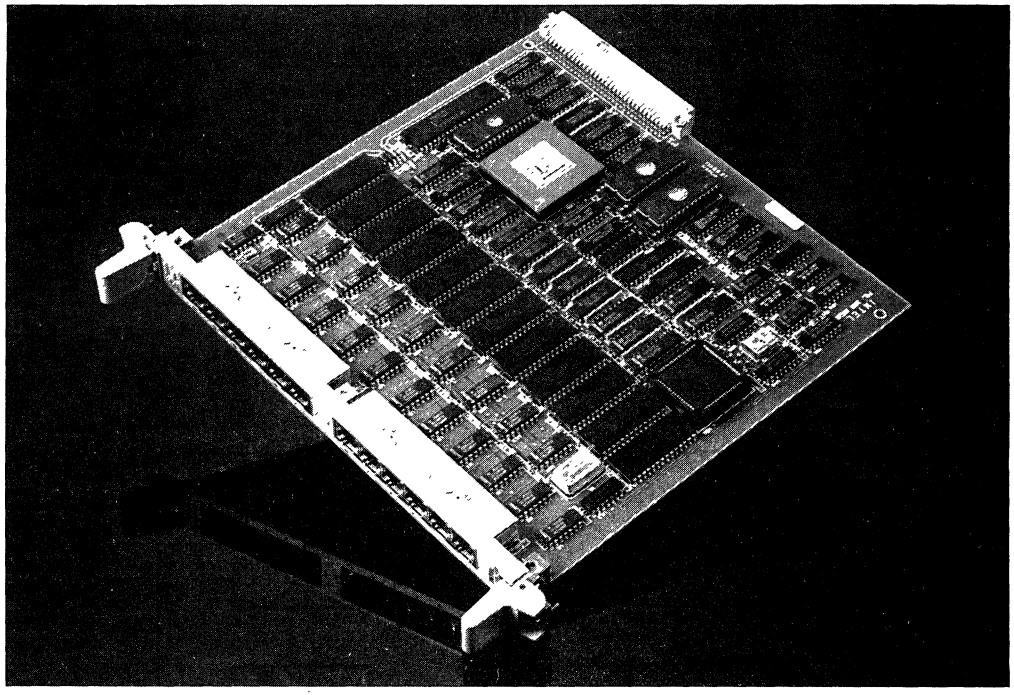


Figure 6: Block Diagram of iSBC®MPI/450 Terminal Controller



iSBC®186/450 TERMINAL CONTROLLER

The iSBC 186/450 is a high performance intelligent terminal controller. The 80C186 CPU, 512K RAM and ATCS served software allow this board to offload the terminal I/O processing from the MULTIBUS II application CPUs.

iSBC®186/450 FEATURES

- 80C186-based microprocessor operating at 12.5 MHz.
- 512K RAM, 128K or 256K EPROM
- Full MULTIBUS II Parallel System Bus interface provided by the Message Passing Coprocessor (MPC)
- 12 ports, RS232C compatible
- 8 signal support, RJ45 (Phone Jack Style) shielded connectors
- Asynchronous Terminal Control Software (ATCS) for interrupt processing, character handling and modem support
- Multiple host support including dynamic line switching
- Resident firmware to support Built-In-Self-Tests (BIST), host-to-controller software download

TERMINAL CONTROLLER SUPPORT

The iSBC 186/450 takes the role of dedicated terminal controller in the system by offloading the application CPU of the task of handling terminal interrupts and character processing. This controller can also be the server for the non-intelligent iSBC MPI/450 boards, using the ATCS software resident on the iSBC 186/450 to drive the MPI-based I/O ports

PERFORMANCE

The performance for the 12 channels of the iSBC 186/450 with the ATCS software can be measured at 19.2k baud sustained output, and 19.2k baud input in 2k byte bursts for all channels in a full duplex mode. If additional channels are required, they may be added by introducing additional iSBC 186/450's as additional standalone terminal controllers or by adding additional iSBC MPI/450 boards into the system and utilizing the iSBC 186/450 as a server.

SUPPORT FOR MULTIPLE HOSTS

The ATCS software has the ability to service multiple hosts. The same terminal may be connected to multiple clients and dynamic line switching is supported by the ATCS software.

FIRMWARE

The iSBC 186/450 contains two 32 pin EPROM sites with firmware that includes Built-In-Self-Tests (BISTS) and host-to-controller download code for soft-loading the ATCS software onto the board.

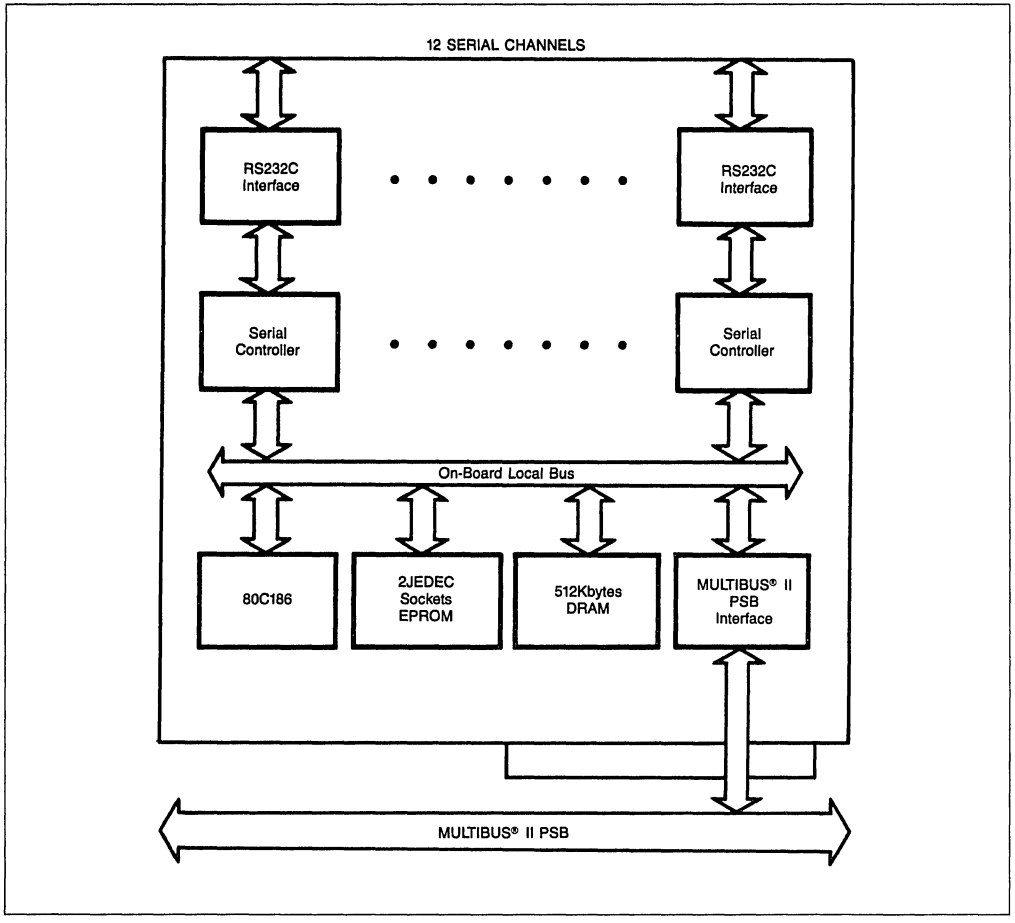


Figure 7: Block Diagram for iSBC® 186/450 Terminal Controller

iSBC®186/450 CONTROLLER SPECIFICATIONS

Clock Rate

80C186 Microprocessor 12.5 MHz

EPROM Memory

Two 32-pin sites.
Supports either 128K bytes or 256K bytes

DRAM Memory

512K bytes installed on the board

Interrupt Capabilities

5 levels with 5 on-board sources

Interfaces

P1	Full PSB
Serial	12 channels, RS232C, 8-pin RJ-45 connectors, 82510 Controller

Device Drivers

Check the latest release of the following operating systems for details:
iRMX II Operating System
UNIX* System V/386 Operating System

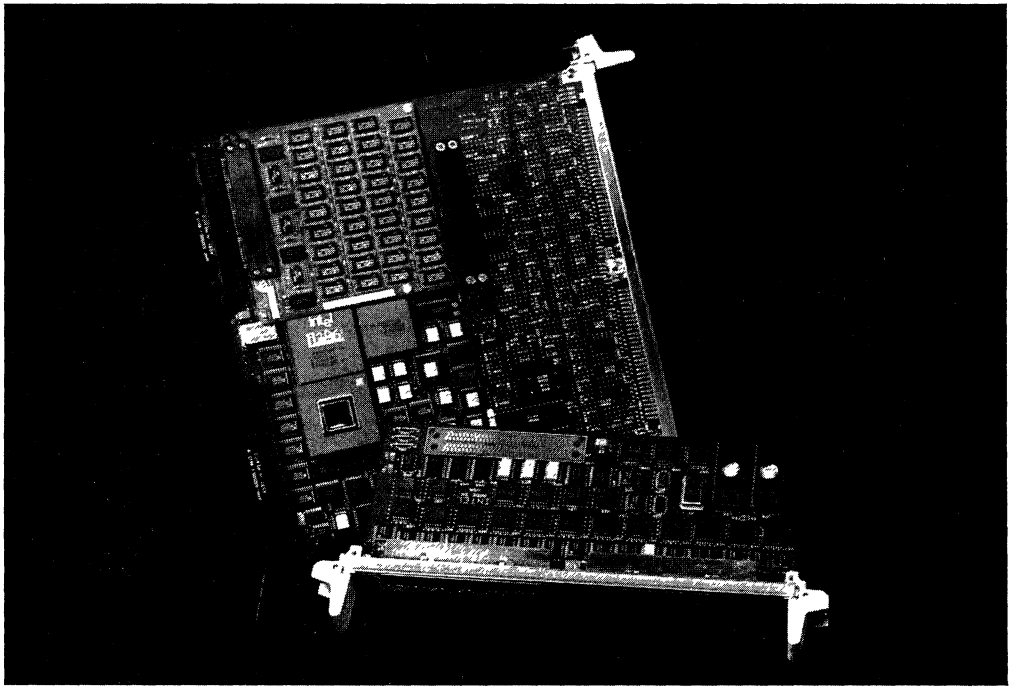
Physical Characteristics

Standard MULTIBUS II board

Power Requirements

(Excluding user-installed memory devices)

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+ 5	4	20
+12	.3	3.6
−12	.3	3.6



MIX 450 TERMINAL CONTROLLER

The MIX 450 terminal module, when combined with the 386™ CPU-based baseboard, provides high performance terminal server capability for MULTIBUS II systems. The MIX 450 module, as a single module on the MIX baseboard, is a powerful 12 port terminal I/O controller. The module can also be stacked three high to expand the terminal support to 36 ports. Stacking the MIX 450 with other MIX modules allows the system designer to build to a multi-function I/O server with terminal capabilities.

MIX 450 FEATURES

- 12 ports per board, RS232C compatible
- 8 signal support, RJ45 (Phone Jack Style) shielded connectors
- Asynchronous Terminal Control Software (ATCS) for interrupt processing, character handling and modem support
- Multiple host support including dynamic line switching
- Resident firmware to support Built-In-Self-Tests (BIST)

PERFORMANCE

The MIX 386/450 supplies the highest performance of the terminal controllers offered by Intel. The MIX 386/450 can support 12 channels in fully sustained, simultaneous input and output transmission at 19.2k baud. Two additional modules can be added to the MIX stack for up to 36 channels.

SUPPORT FOR MULTIPLE HOSTS

The ATCS software has the ability to service multiple hosts. The same terminal may be connected to multiple clients. Dynamic line switching is also supported.

FIRMWARE

The MIX 450 module contains two 32 pin EPROM sites with firmware that includes Built-In Self Tests (BISTs). Upon power-up, the MIX baseboard copies the MIX 450 BIST code up from the module to the baseboard, where it is executed during initialization.

The MIX baseboard firmware provides the capability of downloading the ATCS software.

MIX ARCHITECTURE

Intel's Modular Interface eXtension (MIX) architecture provides a high-performance terminal controller with built-in high performance, on-board I/O expansion. It is optimized for the i386(tm) microprocessor family and the MULTIBUS II System Architecture. The MIX bus allows for easy expansion of terminal support by stacking one to three additional MIX I/O modules. The I/O module interface to the MIX bus is open, with specifications and documentation available from Intel.

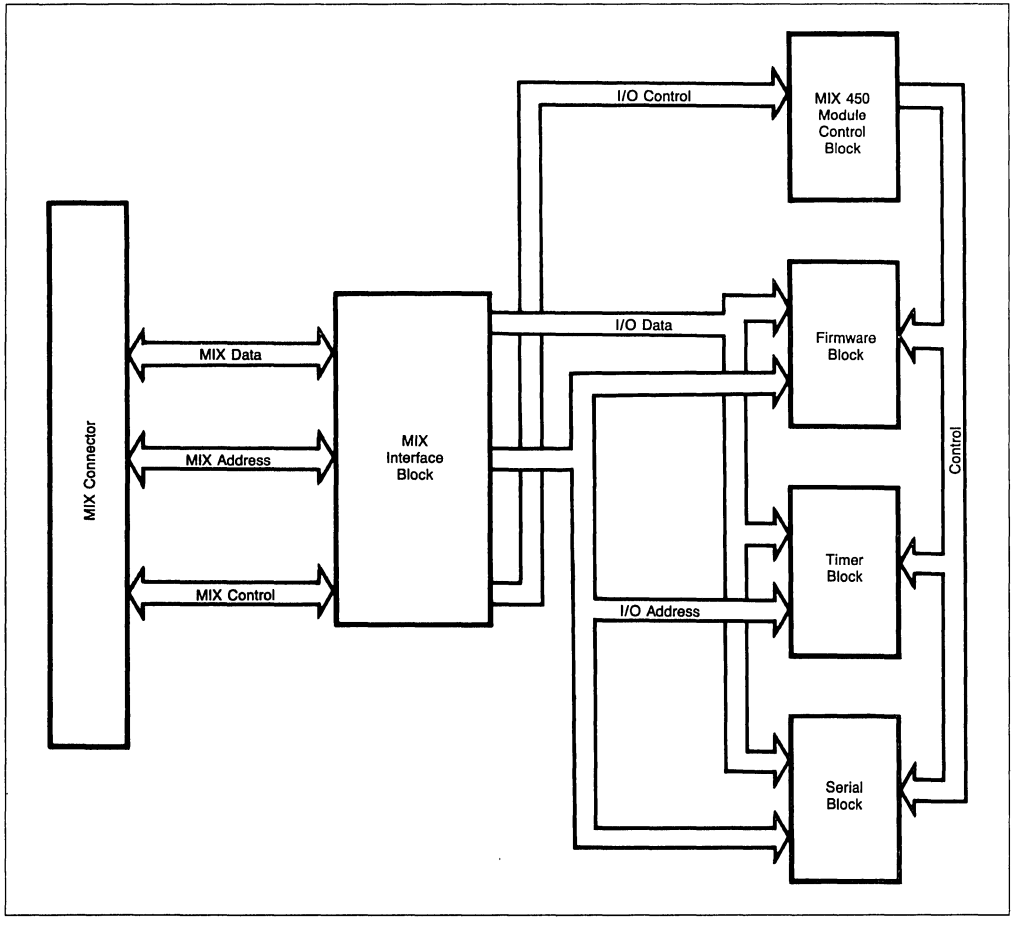


Figure 8: MIX 450 Terminal Controller Module Block Diagram

MIX 450 SPECIFICATIONS†

†For Baseboard specifications, refer to the section on the MIX 386/020 Baseboard

Clock Rates

82C54 Programmable Interval Timer 1.15 MHz

EPROM Memory

Two 32-pin JEDEC sites:

EEPROM

128 bytes installed on the module

Interfaces

Serial	12 channels, RS232C, 8-pin RJ45 connector, 82510 Controller
MIX	Bus slave

Device Drivers

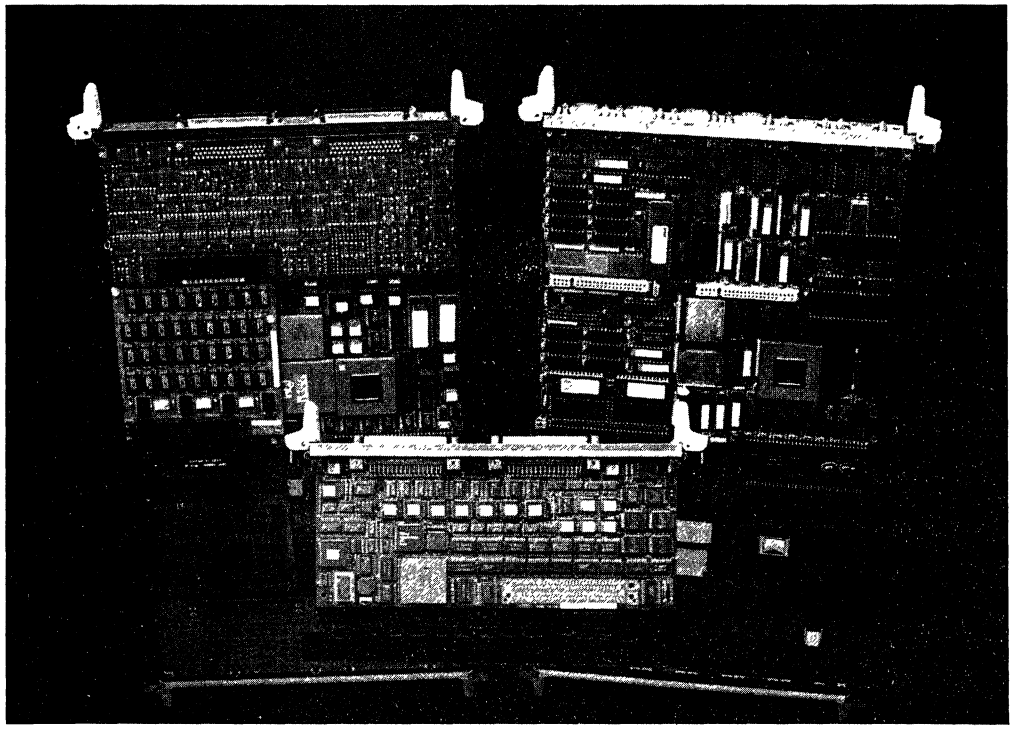
Check the latest release of the following operating systems for details.
iRMX II Operating System
UNIX* System V/386 Operating System

Physical Characteristics

Standard MIX Module

Power Requirements

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+ 5	3.0	15.0
+12	0.02	0.24
−12	0.02	0.24



A CHOICE OF HIGH PERFORMANCE SYNCHRONOUS CONTROLLERS.

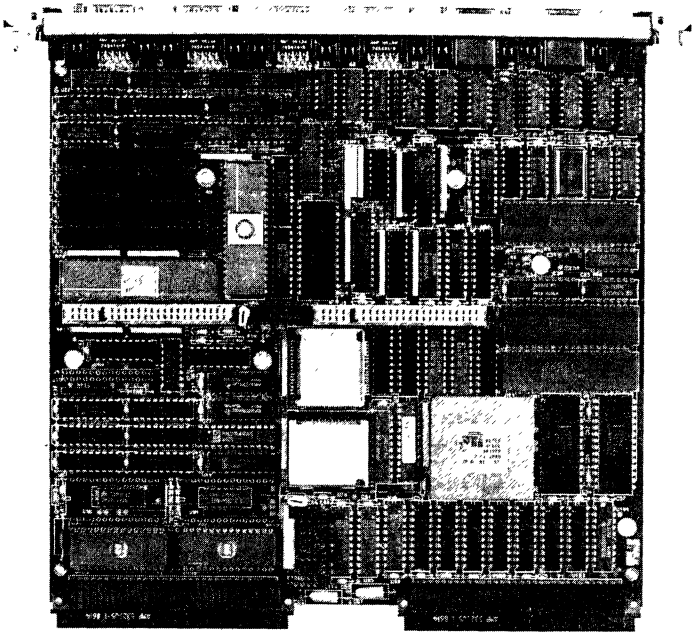
Intel provides two synchronous board solutions targeted towards Wide Area Network Applications. Both boards provide the hardware platforms that support commercial Wide Area Network protocols. The two boards provide price and performance options that can be tailored to individual application needs.

The iSBC 186/410 is a standalone communications controller. Within a MULTIBUS II system, the iSBC 186/410 can optimize overall performance by assuming control of the Wide Area Network administration, reducing the primary system CPU overhead.

For applications requiring high speed synchronous control and mainframe communication, the MIX 386/420 provides an optimal solution. The MIX 386/420 combines the intelligence and performance of the MIX 386 Baseboard with the focused synchronous control of the MIX 420 module. The Modular Interface eXtension (MIX) Architecture allows the user to stack up to three modules on the baseboard. This feature can be used to expand up to six channels of high speed synchronous control or to add other I/O capabilities.

WIDE AREA NETWORK CONTROLLER FEATURES

- Two high performance Wide Area Network Controllers address a range of price and performance requirements
- Intelligent controllers based on 80C186 and 386™ microprocessors with compatible synchronous serial controllers (82530 and 85C30)
- On board Built-In-Self-Test (BIST) with diagnostics



iSBC®186/410 WIDE AREA NETWORK CONTROLLER

The iSBC 186/410 MULTIBUS II Serial Communications Board is an intelligent 6-channel communications processor that addresses the needs of many standard communication applications. The board brings flexibility to the application with its multiple serial channels as well as I/O expansion through the SBX connections.

The iSBC 186/410 is designed to support serial communication within the system. The iSBC 186/410 board supports asynchronous, byte synchronous, and bit-synchronous (HDLC/SDLC) communications on the two full/half duplex RS232C or RS422A channels. On the remaining four channels, only asynchronous mode (RS232C) is supported in either full or half duplex operation. Each serial channel can be individually programmed for different baud rates to allow system configurations with differing terminal types.

iSBC®186/410 FEATURES

- 8 MHz 80C186 Microprocessor
- Six Serial Communication Channels, Two RS232C or RS422A, Four RS232C Only, Front Panel Connections
- 82258 DMA Controller Provides 4 Independent DMA Channels
- 512K Bytes DRAM Provided, Four 28 Pin JEDEC Sites available for EPROM
- Two iSBX Connector provided for I/O Expansion

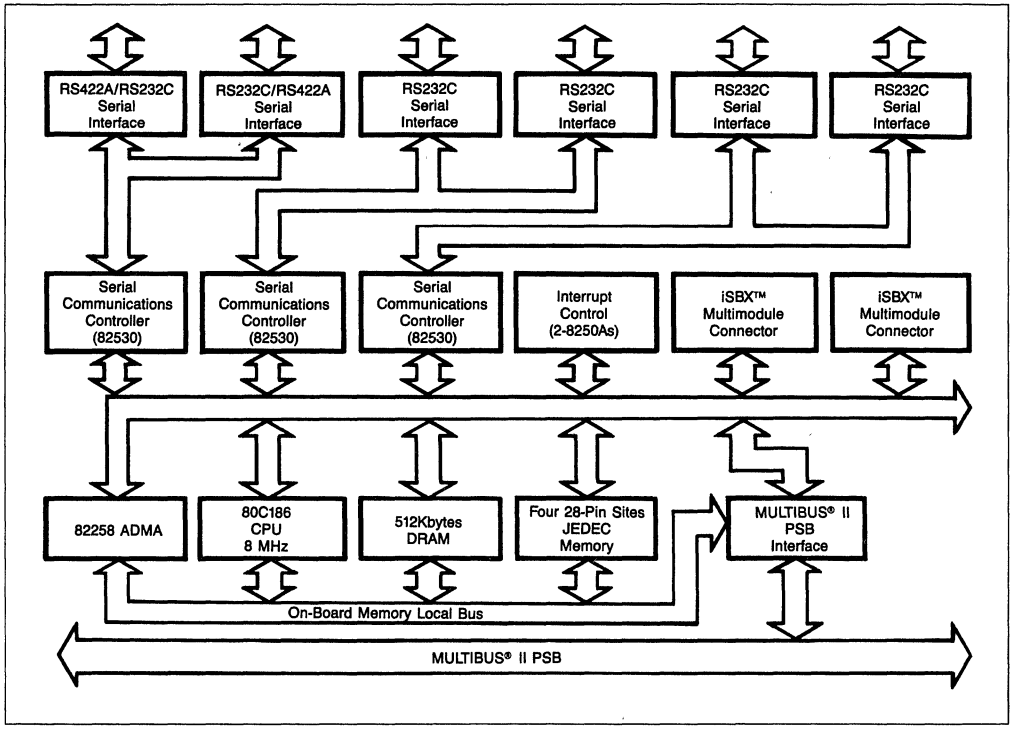


Figure 9: Block Diagram of iSBC® 186/410 WAN Controller

iSBC®186/410 WAN CONTROLLER SPECIFICATIONS

Clock Rates

80C186 Microprocessor	8 MHz
82258 ADMA	8 MHz
8751 Microcontroller	12 MHz
82C54 Timer	(Programmable)

EPROM Memory

Four 28-pin Socket

DRAM Memory

512K bytes

Interrupt Capabilities

14 programmable interrupts

Interfaces

P1	Full PSB
Serial	4 Channels, RS-232C only
	2 Channels, RS-232C or RS-422A
iSBX	2 Single Wide

Device Drivers

Check the latest release of the following operating systems for details:
iRMX I and iRMX II Operating Systems
UNIX* System V/386 Operating System

Serial Communications Characteristics

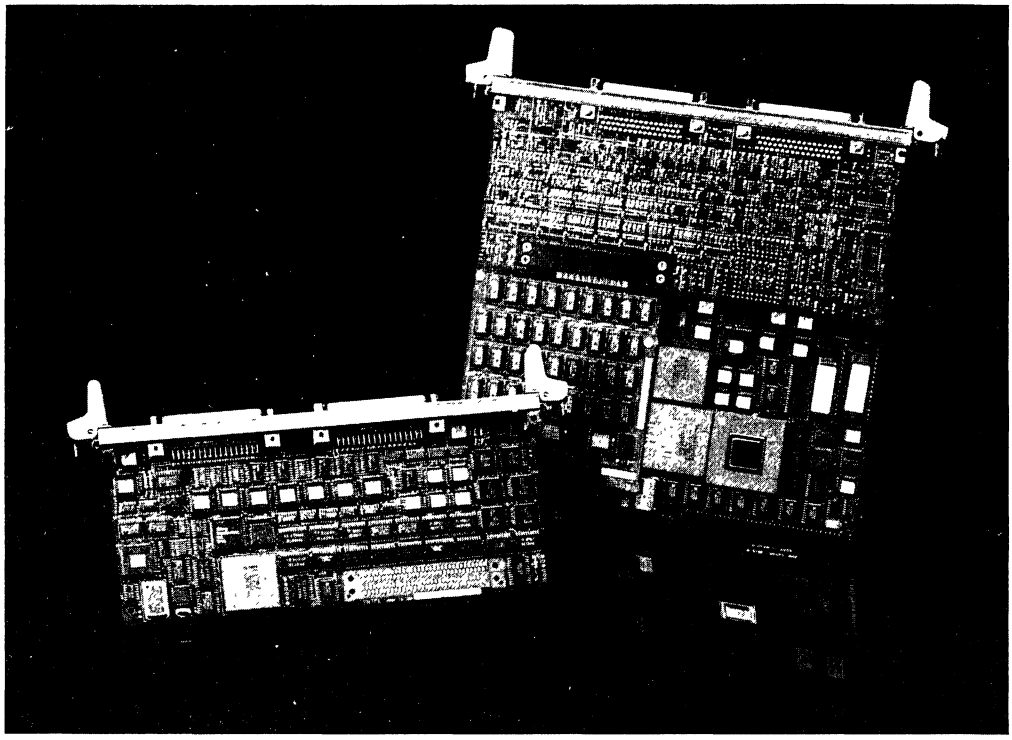
Synchronous:	internal or external character synchronization on one or two synchronous characters.
Asynchronous:	5-8 data bits and 1, 1-1/2 or 2 stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection.
Serial I/O:	RS232C or RS422A compatible, configured DTE only; 4 ch. RS232C IBM compatible only, configured DTE only.

Physical Characteristics

Standard MULTIBUS II board

Power Requirements

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+5	8.22A	43.16W
+12	150mA	1.89W
−12	150mA	1.89W



MIX 420 WIDE AREA NETWORK MODULE

The MIX 420 module combines two high speed synchronous channels with the 386™ CPU-based MIX baseboard to build a high performance Wide Area Network (WAN) platform. The MIX 420 modules can be stacked to a maximum level of three modules for expansion up to six channels. The MIX 420 can be stacked with other MIX modules, allowing the system designer to build a multi-function I/O server with WAN capabilities.

MIX 420 FEATURES

- Two Independent High Speed Synchronous Channels using a 10 MHz 85C30 Serial Communication Controller (SCC).
- High Performance communications capable of 64 kbit/second, with ADMA, and Bypass (slave) speed modes.
- Flexible communications with an 82C54 Programmable Interface Timer and either channel interrupts or hardware interrupts using the 82C59 component.
- "Smart Cable" interface using AT&T General Purpose Synchronous (GPSYNC) Standard
- Designed as a hardware platform for Synchronous Protocol Support, including SDLC/HDLC, SNA, Bisync/Async, SNA, X.25, X.21, X.21 BIS, LU6.2
- Firmware containing Built-In Self Test (BIST) code.

HIGH PERFORMANCE WIDE AREA NETWORK CONNECTION

The MIX 420 Module provides two independent Wide Area Network interfaces using the 85C30 SCC with resulting transfer rates up to 64 Kbit/sec on each channel simultaneously. Speed is enhanced with an 8 MHz, 82258 ADMA, which supports full duplex DMA access to each serial channel. A Bypass (slave) mode is also supported where the CPU handles the transfers, bypassing the ADMA. This mode allows for the MIX 420 to act as a slave module reducing software complexity or allows for the designer to check the hardware functions of the board.

VERSATILE PROTOCOL SUPPORT

The MIX 420 module provides a hardware platform for synchronous communication protocols. The 85C30 based channels provide standard hardware support for SDLC/HDLC, Bisync and Async. The "Smart Cable" interface provided by the AT&T GPSYNC cable addresses the high level protocols by controlling the electronic specification level via an intelligent cable. This interface allows the designer to switch protocols by merely addressing the software issues and swapping to a new cable. The cable will address the electronic difference between the interfaces such as X.21 RS232, AS449, V.35, V.36 or X.24. Finally, the symmetrical design offers channel independence allowing for unique protocols and baud rates to be run simultaneously on the module.

UTILIZING THE MIX ARCHITECTURE

The MIX 386/420 utilizes the Modular Interface eXtension (MIX) architecture. The MIX architecture provides an intelligent base CPU to be combined with specific I/O modules to create a communication platform. Specifically the MIX 386 baseboard provides a 20 MHz 80386 CPU and 1 to 17 MBytes Fast Page Memory. Modular stacking allows for up to three modules to be stacked per baseboard, allowing for up to six high speed synchronous channels through the use of the MIX 420 Module.

FIRMWARE

Two 32 pin EPROM sites reside on the MIX 420 for firmware. Included in the firmware is Built-In-Self-Test (BIST) to check basic functionality of the module and MIX interface. Upon power up, the MIX baseboard copies the MIX 420 BIST from the module to the baseboard where it is executed during initialization.

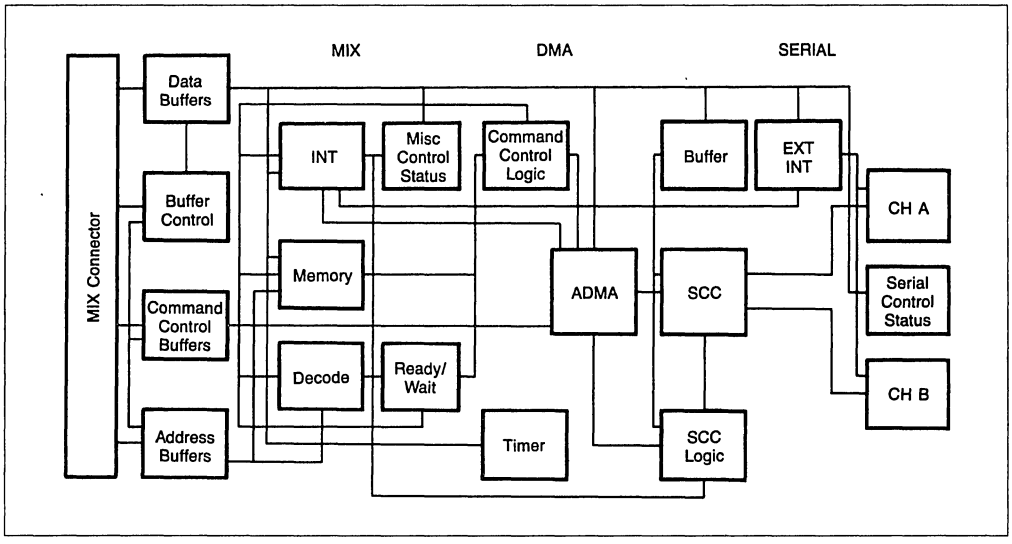


Figure 10: Block Diagram for MIX 420 Module

MIX 420 SPECIFICATIONS†

†For baseboard specifications, refer to the section on the MIX 386/020 Baseboard.

Programmable Baud Rates

110 K bit/sec – 64 K bit/sec

Interrupts

Mode	Level
Channel	2
Device	8

Programmable Interval Timer

82C54

EPROM Memory

Two 32-pin JEDEC sites

EEPROM

128 bytes installed

DMA

82258 ADMA 8 MHz

Interfaces

Serial 2 Channels, AT&T GPSYNC Interface, RS-232, AS 449, V.36, V.36, X.24, X.21, 85C30 Controller, 10 MHz

MIX Bus Slave

Device Drivers

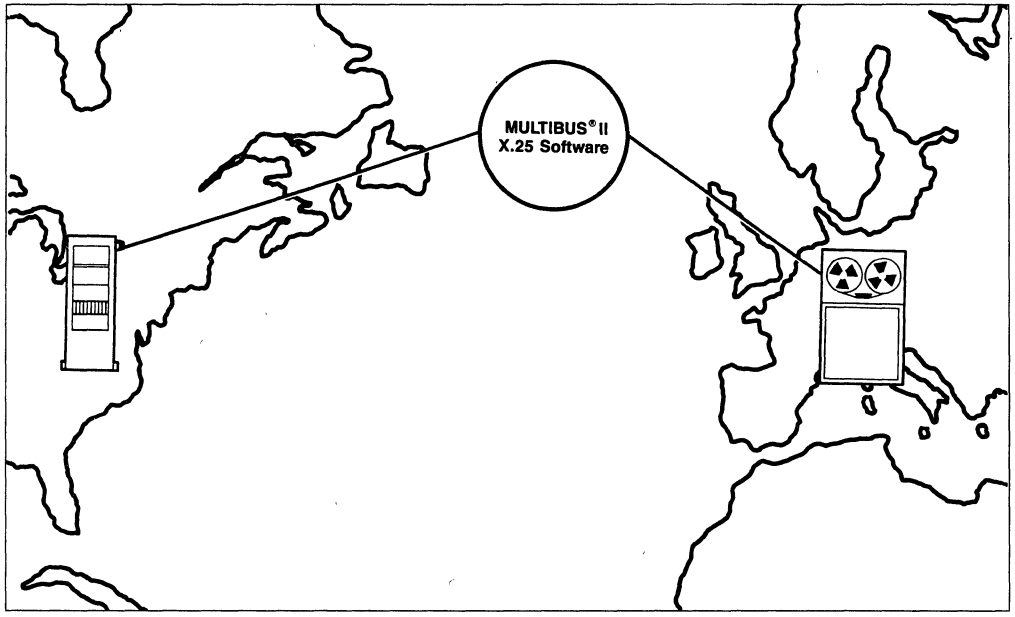
Check the latest release of the following operating systems for details:
iRMX II Operating System
UNIX* System V/386 Operating System

Physical Characteristics

Standard MIX module

Power Requirements

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+ 5	3.0	15.0
+12	.02	0.24
–12	.02	0.24



MULTIBUS® II X.25 SOFTWARE

X.25 is an international standard synchronous bit-orientated serial communications protocol based on CCITT Recommendations. The protocol provides connection oriented communications, ie virtual circuits. The basic unit of transfer is a packet of data. Performance can range with the speeds of the communication lines varying from 110 baud to 256k and above.

X.25 SOFTWARE FEATURES:

- Conforms to CCITT Recommendations 1976, 1980, 1984
- Supports LAPX and LAPB protocols at frame level
- Supports Permanent Virtual Circuits (PVC's)
- Supports Switched Virtual Circuits in the following Modes: Incoming-only, Outgoing-only and Two-way.
- Supports Networks services such as reverse charging, closed user groups, etc.
- Supports x.32 Dial-up features
- Operating parameters of each line can be dynamically changed (e.g. Baud rates, packet size, timeouts, etc.)
- Modular architecture allows optional functionality to be added (e.g. X.3/X.28, PAD, X.29, QLLC, SNA)

IMPLEMENTATION

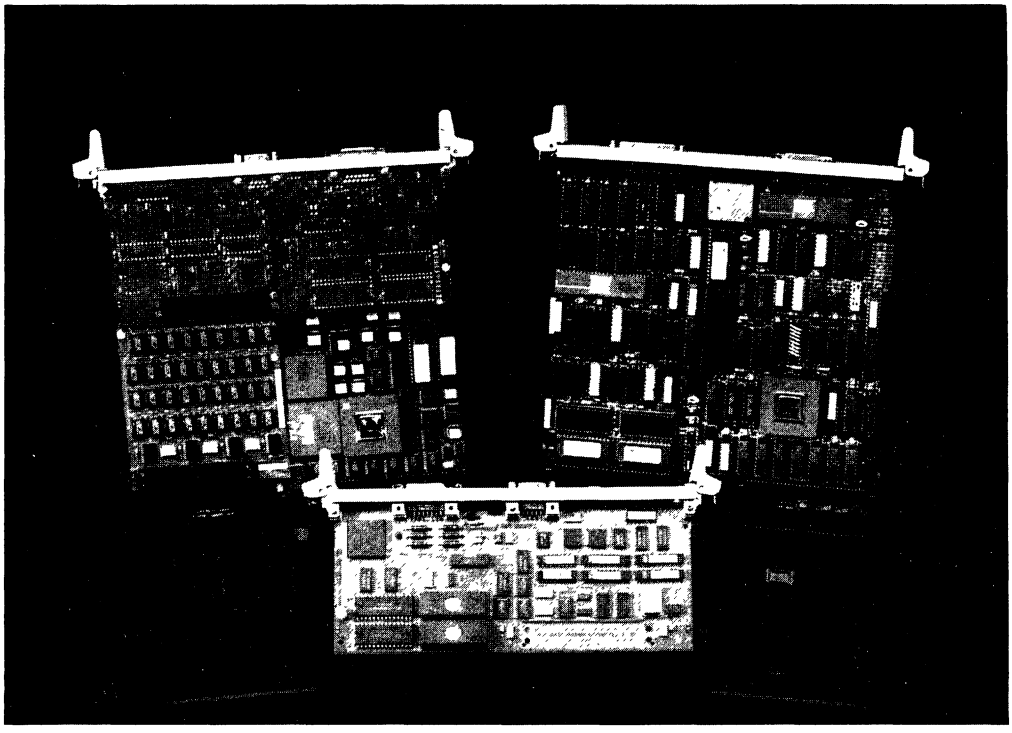
The X.25 package is currently available on the iSBC 186/410. The software can co-exits with the Asynchronous Terminal Controller Software (ATCS), allowing the iSBC 186/410 to support two X.25 lines and four terminals. Driver support for the iRMX II operating system is also available.

INSTALLATION AND SUPPORT

Included with the MULTIBUS II X.25 product is on-site installation performed by Intel Customer Support. This service insures that the software is tested and fully functioning. If further service of the network is desired,
a support contract may be ordered.

CUSTOMIZATION

Intel Customer Support is available and trained to customize the X.25 to fit various applications. Customization may include parameters such as performance tuning, specific OS drivers or application specific requests.



A FAMILY OF MULTIBUS® II ETHERNET LAN CONTROLLERS WITH OpenNET™ NETWORKING SOFTWARE SUPPORT

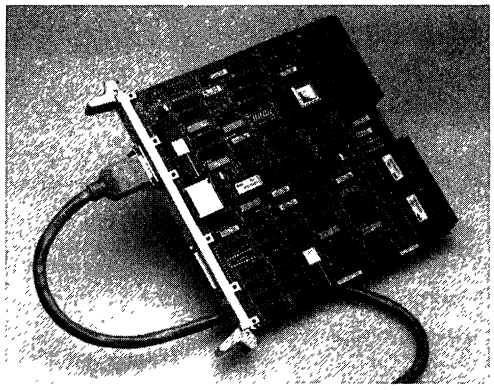
The Intel MULTIBUS II Ethernet LAN controller family provides a range of price and performance for handling MULTIBUS II Ethernet communication requirements. The iSBC® 186/530 is an 80186 CPU-based Ethernet LAN controller that provides a cost-effective LAN connection for many MULTIBUS II designs.

For high performance, the Modular Interface eXtension (MIX) 560 Ethernet module provides Ethernet I/O capabilities to MIX-based I/O subsystems. A MIX-based I/O subsystem that includes the MIX 560 Ethernet module can span the range from a single MIX 560 module mounted on a MIX baseboard to a MIX 560 module mounted in a stack of three MIX I/O modules to provide a tailored MULTIBUS II I/O solution that includes Ethernet communications.

Intel's iNA 960 networking software provides ISO network and transport layer support for both the iSBC 186/530 and the MIX 386/560.

ETHERNET LAN CONTROLLER FEATURES

- A choice of Ethernet LAN controllers providing a range of price and performance.
- Intelligent controllers based on the 80186 and 386 microprocessors and the 82586 LAN Coprocessor.
- Connection to IEEE 802.3 / Ethernet networks for MULTIBUS II systems.
- Support for downloading of networking software over either the MULTIBUS II Parallel System Bus or the Ethernet network.
- ISO Network and Transport (ISO/OSI Layers 3 and 4) networking software support provided by Intel's iNA 960 software.
- iNA 960 networking software executing on the LAN controllers provides a consistent transport interface to host CPU boards.



iSBC®186/530 ETHERNET CONTROLLER

The iSBC 186/530 MULTIBUS II Ethernet Controller is a dedicated IEEE 802.3 compatible front-end processor. The board's 8 MHz 80186, 512K DRAM, and host-to-controller software download capability allows the board to off-load LAN communications functions and I/O software processing from one or all of a MULTIBUS II system's host CPU boards.

iSBC®186/530 ETHERNET CONTROLLER FEATURES

- Provides IEEE 802.3 / Ethernet compatible networking capability for MULTIBUS II systems.
- Resident firmware to support Built-In Self Test (BIST), Initialization and Diagnostic eXecutive (IDX), and host-to-controller software download.
- Four 28-pin JEDEC sites, expandable to 8 sites with iSBC 341 MULTIMODULE™ for a maximum of 512K bytes of EPROM.
- One RS232C serial port for use in debug and testing.
- MULTIBUS II Parallel System Bus interface with full message passing capability.

iSBC®186/530 CONTROLLER SPECIFICATIONS

Clock Rate

80186 Microprocessor 8 MHz

EPROM Memory

Four 28-pin sites.
An additional four 28-pin JEDEC sites may be obtained by installing an iSBC 341 MULTIMODULE.

DRAM Memory

512K bytes installed on the board

Interrupt Capabilities

5 levels with 5 on-board sources

Interfaces

P1	Full PSB
Ethernet	1 channel, 15-pin connector, 82586 LAN Coprocessor
Serial	1 channel, RS232C, 25-pin connector, 8031 Controller

Device Drivers

Check the latest release of the following operating systems for details:
iRMX I and iRMX II Operating Systems
UNIX* System V/386 Operating System

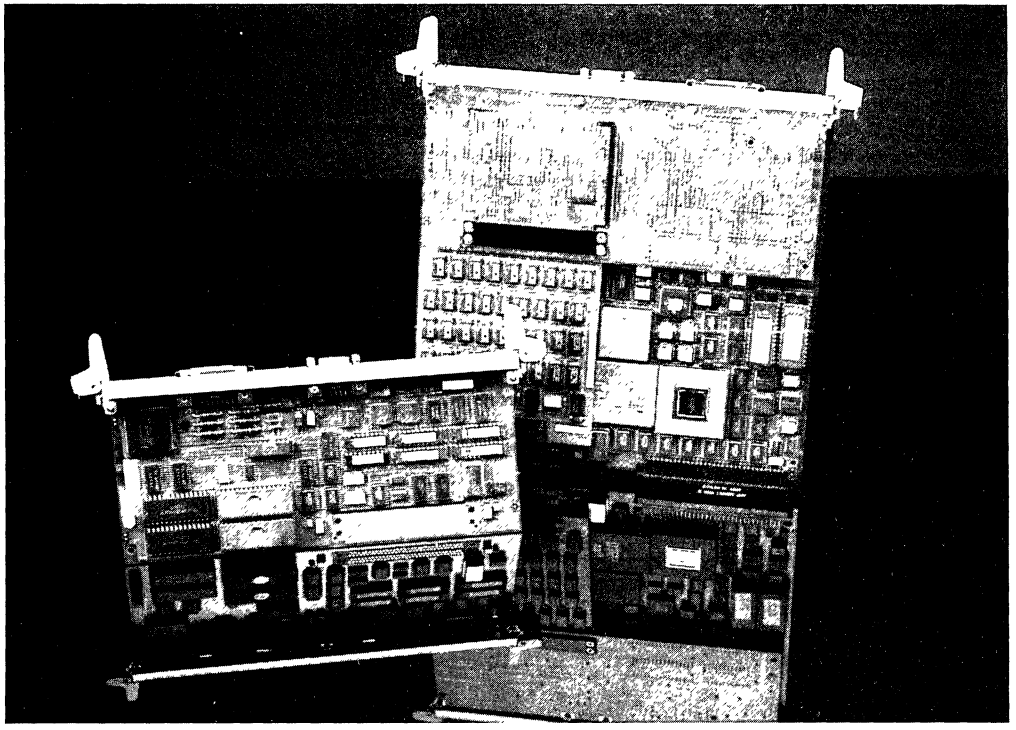
Physical Characteristics

Standard MULTIBUS II board

Power Requirements

(Excluding user-installed memory devices)

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+ 5	8.8	44.0
+12	0.05	.6
−12	0.05	.6



MIX 560 ETHERNET MODULE

The MIX 560 Ethernet Module combines an 82586 LAN Coprocessor, 82501 Ethernet Serial Interface, and 64K bytes of high speed SRAM data buffer to provide high performance Ethernet Modular Interface eXtension (MIX) I/O capabilities. The MIX 560 can be used either in a MIX module stack, to provide Ethernet capabilities to a MULTIBUS II MIX I/O server subsystem, or as a single module on the MIX baseboard, to provide a high performance MIX-based Ethernet controller. Stacking the MIX 560 Ethernet module with other MIX I/O modules allows the system designer to manage the system I/O requirements with a tailored MULTIBUS II I/O subsystem that includes Ethernet communications.

MIX 560 ETHERNET MODULE FEATURES

- 82586 LAN Coprocessor operating at 10 MHz. 82501 Ethernet Serial Interface.
- 64K bytes of SRAM data buffer for handling communications from the MIX baseboard to the Ethernet Interface.
- Support for 128K-265K EPROM.
- Firmware containing Built-In Self Test code.
- LED for 82586 Activity.
- Serial interface for system console or debug.

ETHERNET INTERFACE

The Ethernet interface is implemented using the 82586 Ethernet Coprocessor, the 82501 Ethernet Serial Interface controller, and the standard slide-lock 15 pin IEEE 802.3 connector. The 82501 is software configurable to either Ethernet V1.0 or IEEE 802.3 (Ethernet V2.0). IEEE 802.3 is the default. The Ethernet interface operates at a fixed rate of 10 Mbits per second. An Ethernet station address PROM is also provided.

SERIAL INTERFACE

The serial interface is implemented using the 82510 Asynchronous Serial Controller, an RS232 driver/ receiver, and a serial port connector. The connector is an IBM-compatible 9-pin DTE interface (only 3 pins are used). The port is intended for use as the system console or as a debug port. The serial interface supports baud rates up to 19.2K.

TIMERS

Two 16-bit interval timers are provided by an 82C54 for generating timed, independent interrupts at the MIX interface. A third timer, also provided by the 82C54, is used as a 16 bit prescaler to the other two timers. The timers are used by Intel's iNA 960 Networking Software.

STATIC RAM

The MIX 560 contains 64K bytes of Static RAM (SRAM). The memory is shared between the MIX interface and the 82586. Networking software executing on the MIX baseboard can use the SRAM as a buffer to send and receive data over the Ethernet interface as well as issue commands to, and receive status from, the 82586 Ethernet Coprocessor.

FIRMWARE

The MIX 560 module contains two 32 pin sockets, accommodating either two 27512 or two 27010 EPROMS. Firmware provided for the MIX 560 module includes MIX 560 Built-In Self Test (BIST) code and the software load commands.

The MIX 560 BIST code resides in EPROM on the module. On power up, the MIX baseboard copies the MIX 560 BIST code from the module to the baseboard where it is executed during initialization. The MIX 560 BIST contains 12 tests for exercising the module and verification of functionality. The MIX 560 firmware provides LAN software load commands for the MIX baseboard. The firmware will both upload and download software to the MIX baseboard using either the MULTIBUS II Message Passing Coprocessor or the Parallel System Bus shared memory. The firmware commands also provide the ability to read and set the Ethernet station address and start execution of LAN software code on the MIX baseboard.

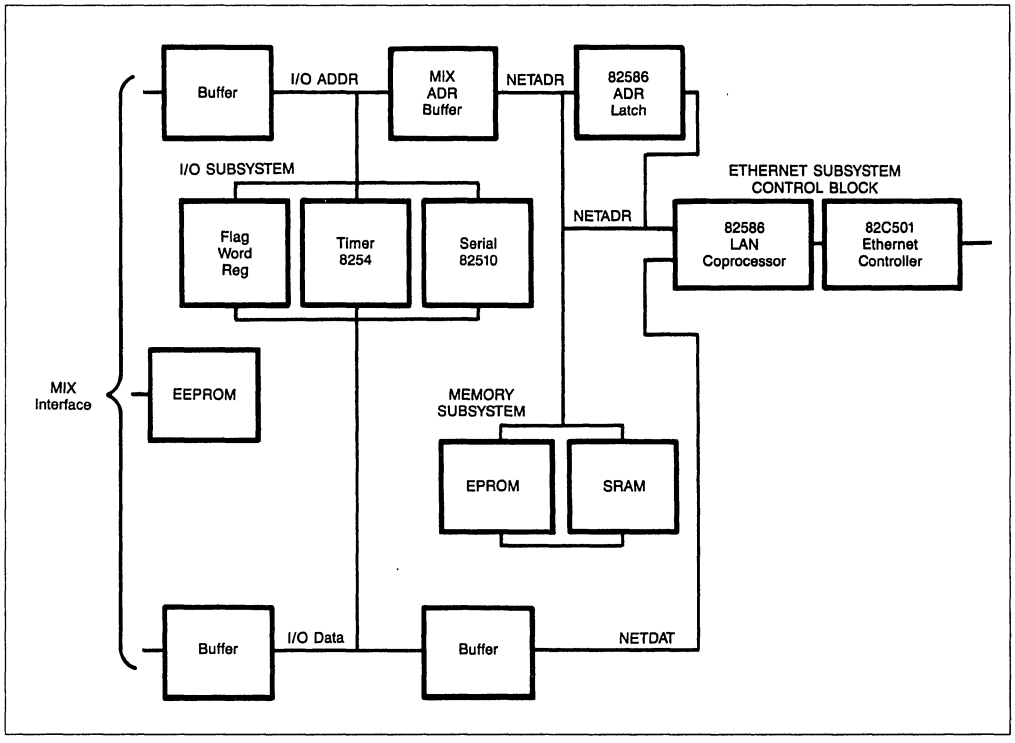


Figure 11: MIX 560 Module Block Diagram

MIX 560 SPECIFICATIONS†

†For Baseboard specifications, refer to the section on the MIX 386/020 Baseboard

Clock Rates

82586 10 MHz

EPROM Memory

Two 32-pin JEDEC sites:

SRAM Memory

64K bytes installed on the module

EEPROM

128 bytes installed on the module

PROM

6 bytes for Ethernet node address

Interfaces

Ethernet	1 channel, 15-pin connector, 82586 LAN Coprocessor
Serial	1 channel, RS232C, 9-pin (IBM-compatible) connector, 82510 Controller
MIX	Bus slave

Device Drivers

Check the latest release of the following operating systems for details:
iRMX II Operating System
UNIX* System V/386 Operating System

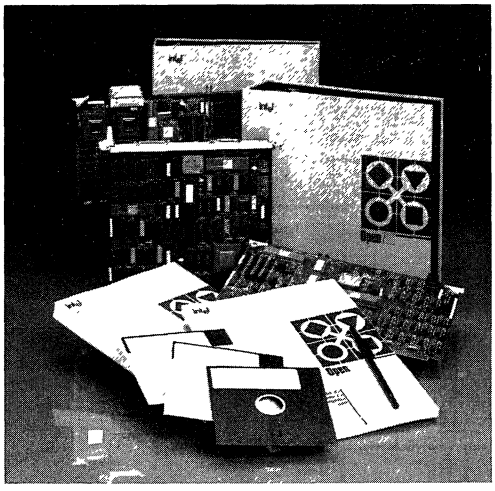
Physical Characteristics

Standard MIX module

Power Requirements

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+ 5	3.0	15.0
+12	0.02	0.24
−12	0.02	0.24

The MIX 560 module also passes fused +12 VDC through the Ethernet connector to an external transceiver. The transceiver may require up to an additional 0.5 Amps (max) of +12 V.



iNA 960 OpenNET™ NETWORKING SOFTWARE FEATURES:

- Certified ISO/OSI Transport and Network Layer Software
- ISO 8072/8073 Transport Class 4
- ISO 8602 Connectionless Transport
- ISO 8348/8473 Connectionless Network
- ISO 9542 End System to Intermediate System (ES/IS) Dynamic Routing
- Comprehensive Network Management Functions
- Remote Boot Server for diskless workstations
- Data Link Drivers for iSBC 552A, iSBX 586, iSBC 554, iSBC 186/51, iSBC 186/530, and MIX 386/560

FULLY COMPLIANT ISO/OSI TRANSPORT AND NETWORK LAYER SOFTWARE

iNA 960 is a complete Network and Transport (ISO/OSI Layers 3 and 4) software system plus a comprehensive set of network management functions, Data Link (OSI Layer 2) drivers for IEEE 802.3 Ethernet and IEEE 802.4 Token Bus (MAP), and system environment features.

FLEXIBLE AND HIGHLY CONFIGURABLE

iNA 960 is a mature, flexible, and ready-to-use software building block for OEM suppliers of networked systems for both manufacturing and office applications (e.g., MAP and TOP).

This software is highly configurable for designs based on the 82586 and 82588 LAN controllers, 82501 and 82502 Ethernet serial interface and transceiver, and the Intel 86 family of microprocessors.

CONFIGURABLE AT THE OBJECT CODE LEVEL

Consisting of linkable object modules, the iNA 960 software can be configured to implement a range of capabilities and interface protocols. iNA 960 has a large installed base and has been used reliably in a variety of systems from IBM PC XT/ATs to VAX/VMS to IBM mainframes.

BASED ON INTERNATIONAL STANDARDS

Based on the ISO/OSI seven layer model for network communications, iNA 960 implements ISO 8073 Transport Class 4 providing reliable full-duplex message delivery service on top of the internet capabilities offered by the network layer. The iNA 960 network layer is an implementation of the ISO 8473 Network Class 3 Connectionless Network Protocol and supports ISO 9542 End System to Intermediate System Network Dynamic Routing. iNA 960 also supports ISO 8602 Connectionless Transport Protocol (Datagram).

PRECONFIGURED iNA 961

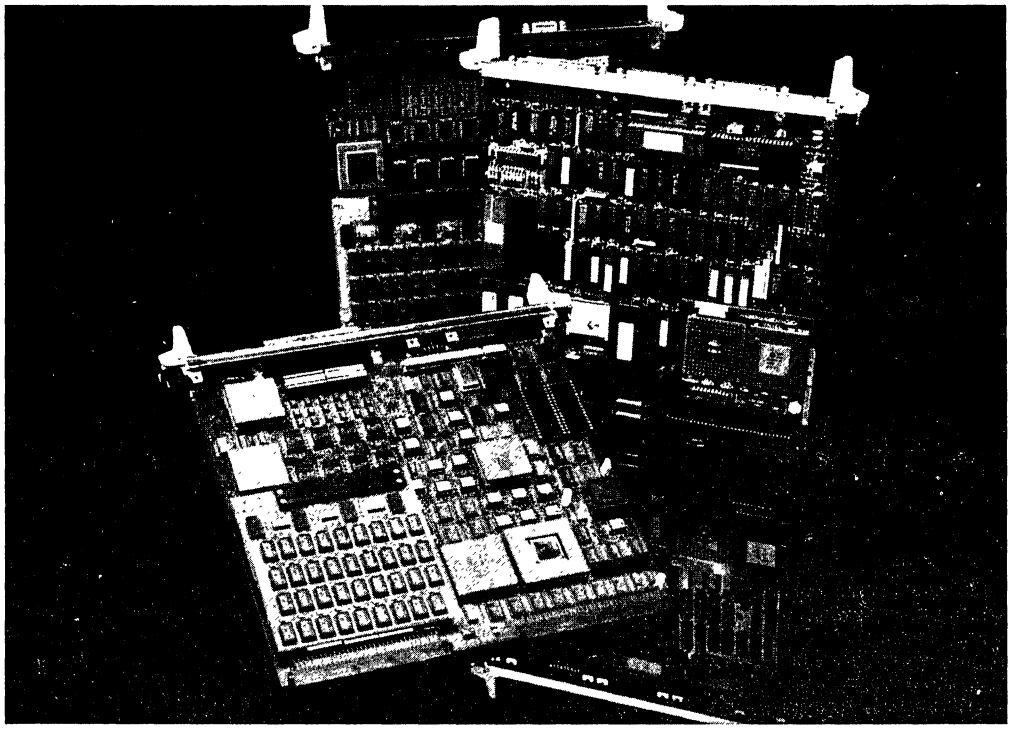
iNA 960 contains the preconfigured iNA 961 software modules which include support for the iSBC 552A, iSBC 554, iSBC 186/530, and the MIX 386/560.

REMOTE BOOT SERVER SUPPORT

iNA 960 provides basic boot server capabilities that will transmit predefined images to diskless network nodes that request them.

MULTI-SERVER/CONSUMER SUPPORT

iNA 960 supports the powerful MULTIBUS II feature of multiple host and communications boards. This is ideal for LAN load balancing and redundant networks for fault-tolerant systems.

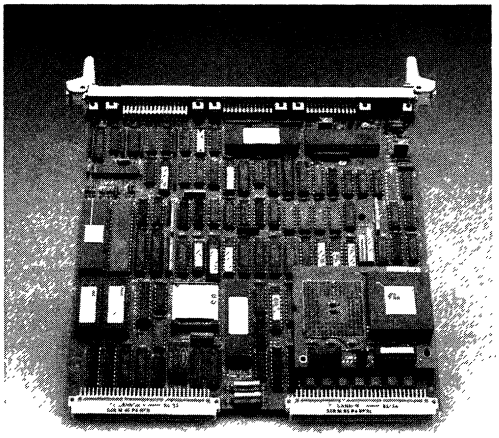


A CHOICE OF PERIPHERAL CONTROLLERS

Intel's product line of MULTIBUS II Peripheral Controllers addresses the diverse interfaces of peripheral communications. The iSBC 186/224A, a Multi-Peripheral Controller Subsystem, provides support for up to four ST506/412 Winchester disk drives, up to four SA450/460 floppy drives and quarter inch QIC-02 streaming tape drives. The iSBC 386/258, a versatile high-performance SCSI peripheral controller, provides performance tuning capability for peripheral devices individually for optimum system performance. Additionally, the iSBC 386/258 can complement the host by offloading it with its powerful 386™ microprocessor.

PERIPHERAL CONTROLLER FEATURES:

- Multiple peripheral interface support
  Small Computer Systems Interface (SCSI)
  ST506/412 Winchester Disk Drive Support
  SA450/460 Floppy Drive Support
  QIC-02 one quarter inch Streaming Tape Drive Support
- Full PSB interface with complete Message Passing Support
- On-board Built-In-Self-Test (BIST) with Diagnostics



iSBC®186/224A MULTI-PERIPHERAL CONTROLLER SUBSYSTEM

The iSBC 186/224A Multi-Peripheral Controller Subsystem provides peripheral I/O control for a variety of OEM applications and supports the full message passing protocol of the MULTIBUS II System Architecture. The iSBC 186/224A controller serves as a complete peripheral I/O subsystem and it supports the predominant types of storage media: Winchester disks, floppy disks and quarter-inch streaming tapes. On-board firmware for the board provides improved Winchester disk operation through multiple data track cacheing.

iSBC®186/224A FEATURES

- 80C186 Microprocessor at 5 MHz
- Controls up to Four ST506/412 Winchester Disk Drives, Four SA450/460 Floppy Drives, and Four QIC-02 Streaming Tape Drives.
- 128K Bytes of On-Board SRAM for multiple track cacheing on high speed Winchester data access.
- Built-In-Self-Test (BIST) Diagnostics On-Board
- Full Message Passing interface to the Parallel System Bus.

iSBC®186/224A PERIPHERAL CONTROLLER SPECIFICATIONS

Clock Rate

80C186 Microprocessor 5 MHz

EPROM Memory

Two 28-pin sites.

DRAM Memory

128K bytes installed on the board

Mass Storage Device Drives

Winchester

ST506/412 compatible 5-1/4″ drives with up to 1024 cylinders. Qualified manufacturers include: Quantum, CMI, CDC, Maxtor, Memorex, Atasi. Densities range from 10 to 140 MB.

Floppy

SA450/460 compatible 5 1/4″ drives. Qualified manufacturers include: Teac and Shugart. Sizes include half height, full height, 48 TPI and 96 TPI.

Tape

QIC-02 compatible, 1/4″ streaming tape drives. Qualified manufacturers include: Archive, Cipher, and Tandberg.

Interfaces

P1	Full PSB
ST506/412	50 pin D-type
SA450/460	25 pin D-type
QIC-02	25 pin D-type

Device Drivers

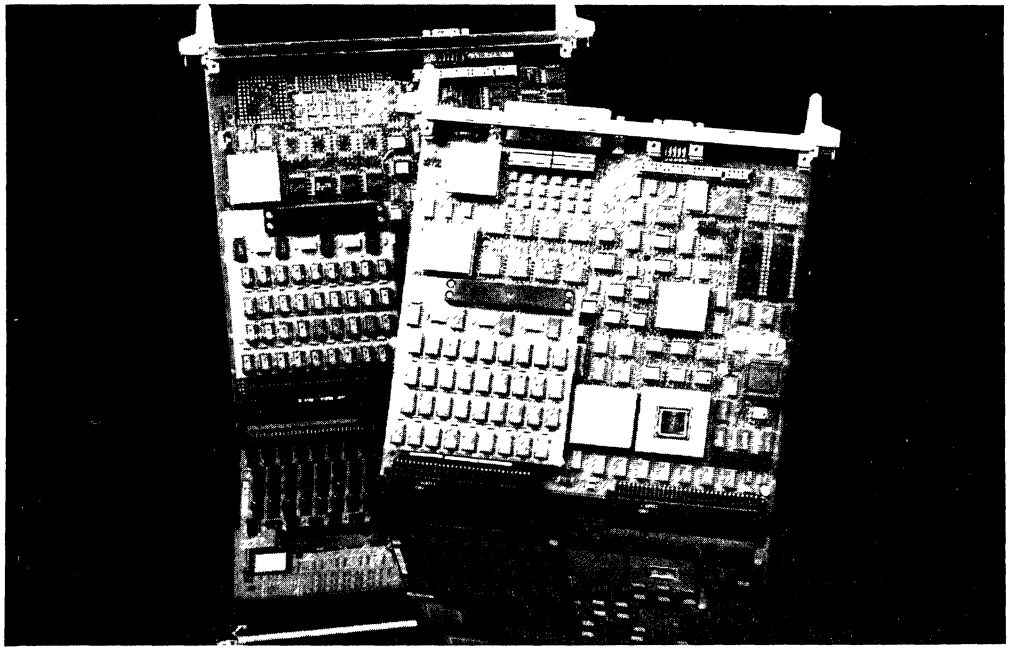
Check the latest release of the following operating systems for details:
iRMX I and iRMX II Operating Systems
UNIX* System V/386 Operating System

Physical Characteristics

Standard MULTIBUS II board.

Power Requirements

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+5	7.0	35.0
+12	0.05	.6
−12	0.05	.6



iSBC® 386/258 SCSI PERIPHERAL CONTROLLER

The iSBC 386/258 is a high-performance peripheral controller that combines powerful I/O performance and access to SCSI peripherals for MULTIBUS II applications.

Minicomputer-level I/O performance is achieved by utilizing the 386™ microprocessor and a large data cache. The added power of the 386™ processor gives the iSBC 386/258 the potential of off-load tasks from other system CPUs as an I/O server. The SCSI standard has achieved wide acceptance because of its extensive capabilities and excellent performance.

ISBC® 386/258 FEATURES

- 16 MHz 386™ microprocessor
- 1 or 4 MByte data buffer
- CSM002 module support
- Common Command Set (CCS) SCSI peripheral support
- Asynchronous SCSI to 1.5 MBytes/sec, synchronous to 4.0 MBytes/sec
- Two Versions: single ended SCSI port only or Dual SCSI ports
- Firmware support for BIST, IDX, slave test handler, and downloader
- 258 Peripheral Communications Interface (258-PCI) firmware

COMPLETE SCSI CAPABILITY

The iSBC 386/258 supports communication with up to seven other peripheral adapters and up to 56 possible devices. Vendor-unique features of peripherals can be accessed using the pass through capability. Also supported is the ability to be a bus initiator, and the use of disconnect/reconnect. Peripherals that support the SCSI standard today include magnetic hard disk, magnetic tape, floppy disk drive, optical disk, and line printers.

HIGH PERFORMANCE

I/O critical applications are accelerated by the combination of a 16 MHz 386™ processor, a large data buffer for cacheing (1 or 4 MBbytes), and the 4.0 MBytes per second synchronous transfer rate for SCSI.

FIRMWARE SUPPORT

The iSBC 386/258 includes EPROMs with firmware support for BIST (Built-in Self-Test), IDX (Initialization and Diagnostics Executive), Power up and slave handler, a downloader, and a peripheral communications interface.

The 258-PCI firmware establishes a high level software protocol to facilitate the exchange of data between host drivers and SCSI-based peripheral devices. It also insulates host drivers from knowledge of SCSI bus management. The 258-PCI server manages up to 64 outstanding commands and permits multiheaded I/O operations with up to 56 SCSI peripheral devices.

The 258-PCI server also allows tuning of the cache configuration, command ordering/seek optimization, and reporting of usage statistics, like the number of cache hits and misses, total number of reads, writes, and errors.

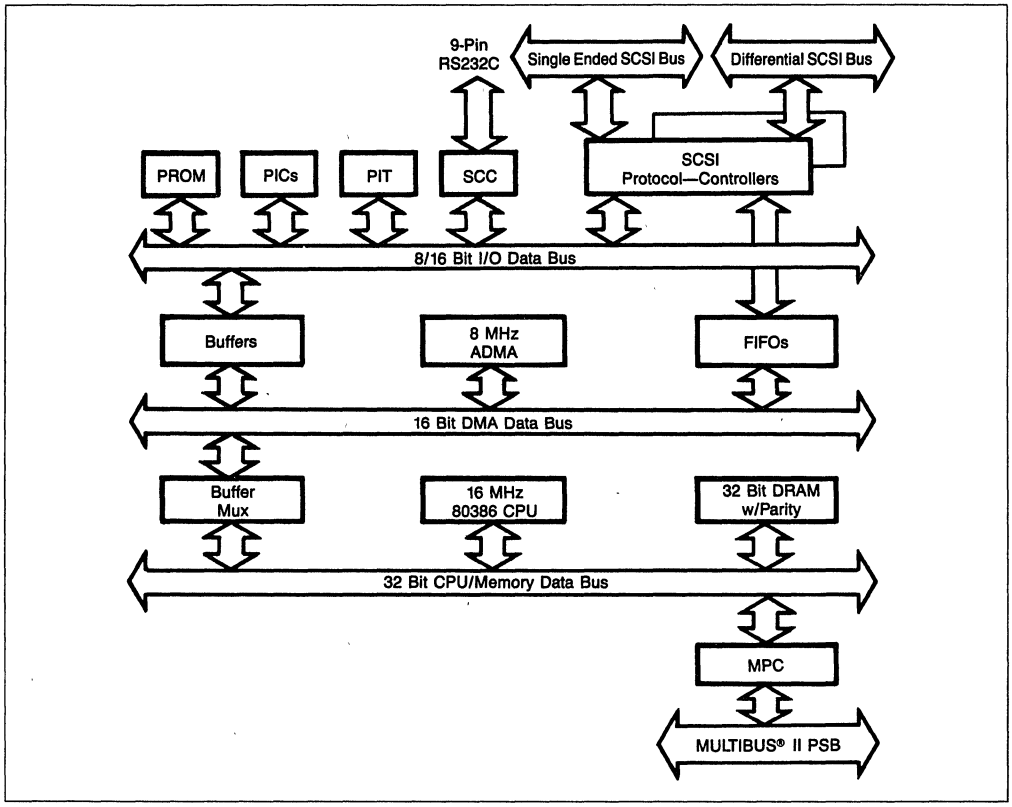


Figure 12: Block Diagram for iSBC® 386/258 Peripheral Controller

iSBC®386/258 PERIPHERAL CONTROLLER SPECIFICATIONS

Clock Rates

386™ DX Microprocessor	16 MHz
82258 ADMA	8 MHz
8751 Microcontroller	12 MHz

EPROM Memory

Two 32-pin Sockets

DRAM Memory

1M or 4M byte installed on the baseboard

Interrupt Capabilities

14 programmable interrupts

Interfaces

- P1, Full PSB
- P2, SCSI: ANSI X3.131–1986, Single-ended or dual versions available.
- iSBX Bus Interface
- Serial I/O Port: RS-232-C (subset)interface (DTE). 9-pin D-shell shielded connector.

Device Drivers

Check the latest release of the following operating systems for details:
iRMX II Operating System
UNIX* System V/386 Operating System

Physical Characteristics

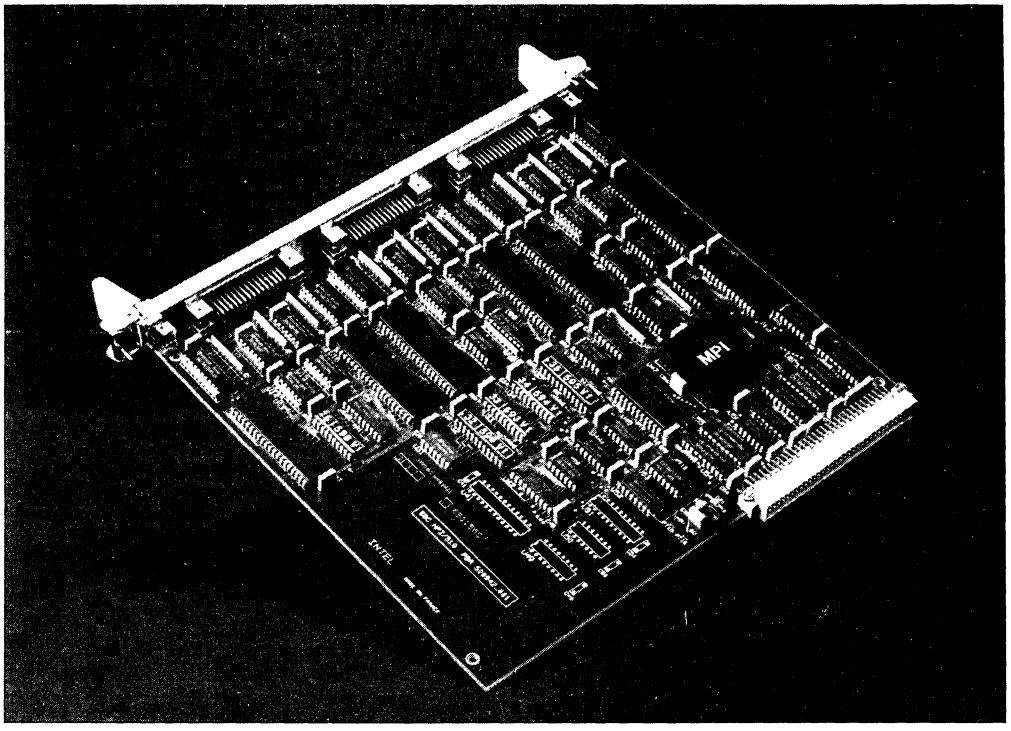
Standard MULTIBUS II board.

Power Requirements

Typical values for power are at the nominal voltageand at an ambient temperature of 25 degrees C. Maximum values are at nominal voltage plus 5% and at an ambient temperature of 0 degrees C.

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+ 5	11.0	55
+12	5.0	60
–12	5.0	60

Does not include power for installed iSBX MULTIMODULE boards



iSBC® MPI/519 72 CHANNEL DIGITAL I/O BOARD

The iSBC MPI/519 is a digital I/O interface board which provides 72 parallel channels of TTL level I/O in Multibus II I/O space. The board is capable of receiving interrupts from other MULTIBUS II agents, as well as generating interrupts from up to 8 sources. It is one of a family of MPI (Multibus II Peripheral Interface)-based I/O boards.

iSBC® MPI/519 FEATURES:

- 72 channels of TTL level I/O in banks of 24 channels each
- Banks configurable for general purpose industrial I/O or as Centronics compatible ports
- Output lines may be read back to verify output status
- Socketed buffer drivers and resistor networks for configuring I/O as high or low true
- 8 interrupt request lines
- Precision interval pulse triggering on one of three I/O lines

FUNCTIONAL DESCRIPTION

The iSBC MPI/519 is a digital I/O board suitable for applications such as industrial automation, printer interface, or for low cost inter-chassis communications requiring multiple parallel I/O lines.

The iSBC MPI/519 board is based on the MULTIBUS II Peripheral Interface (MPI) component which provides all the logic required to interface to the Parallel System Bus (PSB), allows the board to be a replier in I/O and interconnect space, and supports the sending and receiving of interrupt messages.

72 DIGITAL I/O CHANNELS

The 72 channels of TTL (5v) level I/O are arranged in three banks of 24 I/O each (Figure 13). Each bank is implemented using two Intel 82C55 Programmable Peripheral Interface (PPI) components (Figure 14). Port A of each PPI is connected to the front panel through bidirectional buffers. They can be software configured as input or output on a byte basis. Sockets in front of the buffers are provided for the user to add pull-up, pull-down or voltage dividing resistor networks (2.2 K-ohm pull-up resistors are provided). To allow data readback, Port B of each PPI is connected directly to the output side of the Port A buffers, for use in board diagnostics or to ensure the integrity of critical data.

Each bank also has an additional 8 bits I/O implemented through Port C, for use as general purpose I/O or as input and output interrupts.

I/O signals may be interfaced to industry standard signal conditioning and isolation modules through termination panels such as Intel's iRCX910 or OPTO-22's PB24.

CENTRONICS COMPATIBLE

Each bank may be used as a Centronics compatible port. Bank one can automatically generate the data strobe, eliminating an extra bus transaction.

8 INTERRUPT REQUEST LINES

Input interrupts coming from external sources through the front panel are implemented through an 82C59 Programmable Interrupt Controller (PIC) and cause the iSBC MPI/519 to send an unsolicited interrupt message. Up to eight input interrupts are supported. The interrupt source is encoded in the interrupt message. One input interrupt can be configured as a broadcast interrupt, which is sent to all agents. This interrupt is useful to synchronize processors or to alert all processors to an external system event. Output interrupts, received by the iSBC MPI/519 from other agents, cause a 82C54 programmable interval timer to output a precise interval pulse. These pulses can be from 1 msec to 6.5 msec. in length. There is one output line for output interrupts on each I/O bank.

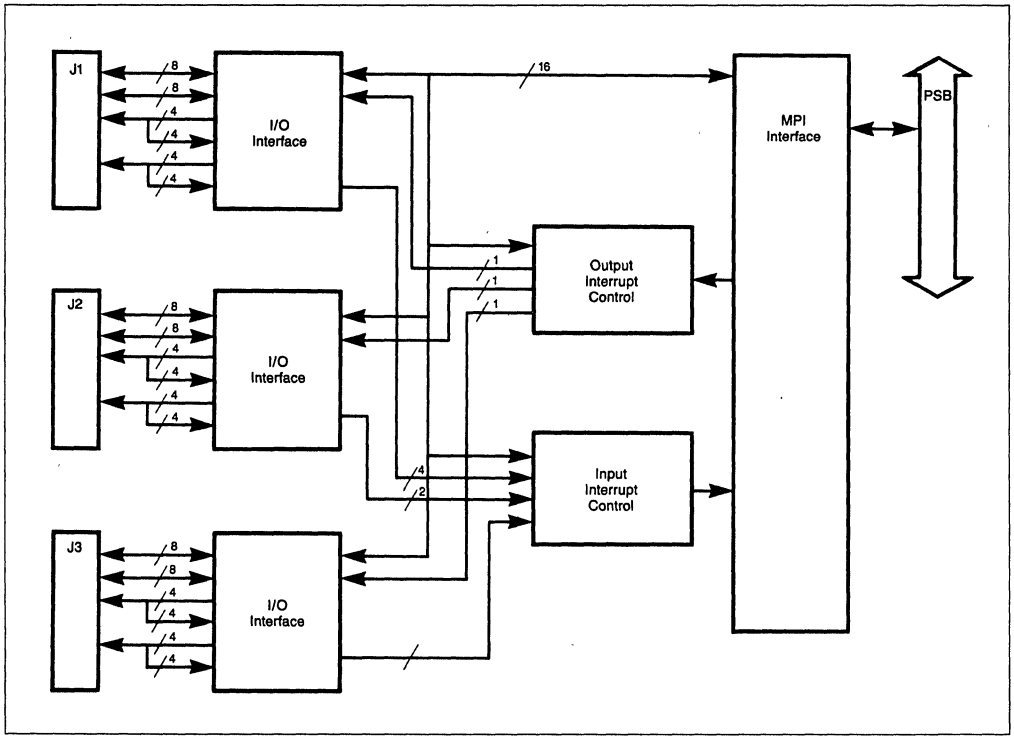


Figure 13: Block Diagram for iSBC® MPI/519 Board

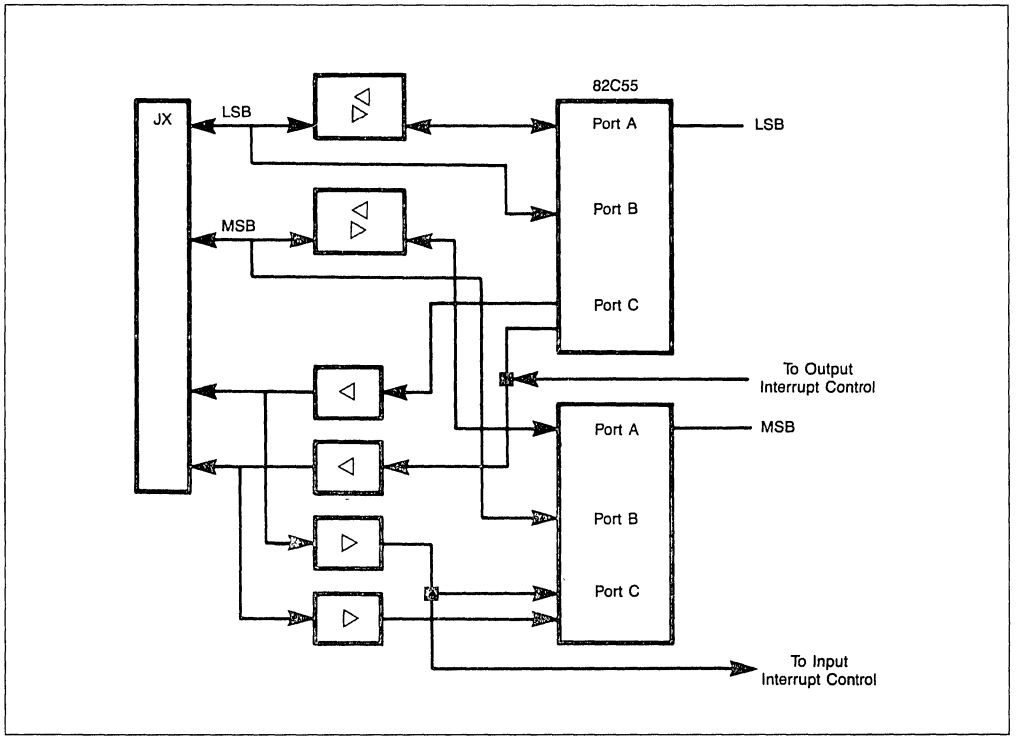


Figure 14: Block Diagram of one iSBC® 519 I/O Bank

iSBC®MPI/519 DIGITAL I/O BOARD SPECIFICATIONS

Interfaces

P1 Slave PSB

Centronics Each of the three I/O banks can be used as a Centronics compatible interface. Bank 1 is configurable to minimize handshaking and bus transactions when used as a printer interface.

I/O connector:3 Positronics ODD44F500TX

Physical Characteristics

Standard MULTIBUS II board

Power Requirements

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+5	3	15

I/O Buffer and Resistors Supplied

Bidirectional Buffers	Unidirectional Buffers	Resistor Networks
74ALS645	74ALS09	2.2 K-Ohm

Other Components Supported

Bidirectional Buffers	Unidirectional Buffers	Resistor Networks
74ALS638-74AS638 74ALS639-74AS639	74ALS00-74AS00 74ALS08-74AS08	9 or 10 pin SIPs all values supported
74ALS640-74AS640 74ALS643-74AS643	74ALS32-74AS32 74ALS37-74AS37 74ALS38-74AS38	
or equivalent	or equivalent	

ENVIRONMENTAL REQUIREMENTS

Operating Temperature: 0 to 55°C @ 200 LFM airflow
Non-operating: −40 to 70°C
Humidity: 0 to 85% non-condensing

INTERFACES

Full PSB

A full PSB interface is implemented with the 82389 MPC component. This interface is Intel's implementation of the IEEE/ANSI 1296 specification cast in silicon. All boards with a full PSB interface have the feature set of the MPC component described in the silicon section.

Slave PSB

A slave PSB interface is implemented with the MPI component. The MPI is a cost and function reduced "little brother" of the MPC component. All boards with the slave PSB interface have the feature set of the MPI component described in the silicon section.

MM0x Memory Expansion

The MM0x interface uses a custom surface mount connector to add expansion local memory to a CPU-based product. The connector allows up to two modules to be added to a baseboard. Memory modules are single-sided (1MB or 4MB) or double-sided (2MB or 8MB). A board with a single-sided module consumes a single MULTIBUS II slot, all other combinations require two MULTIBUS II slots (Note: If two MIX modules are used, then two slots are used).

MIX

The MIX interface is described in the MIX architecture section.

PHYSICAL CHARACTERISTICS

Standard MULTIBUS®II Format (Double 6U Eurocard)

Height: 23.3 cm (9.18 inches)
Depth: 22.0 cm (8 65 inches)
Width: 1.92 cm (0 76 inches)

MIX Expansion Module:

Module Height: 8.9 inches
Module Depth: 3.75 inches
Module Area: 33 square inches

iSBX™ Modules:

	Single-Wide	Double-Wide
Height:	2.1cm (0.827 inches)	2.1cm (0.827 inches)
Depth:	7.24cm (2.85 inches)	7.24cm (2.85 inches)
Width:	9.4cm (3.7 inches)	19.05cm (7.5 inches)

DEVICE DRIVERS

Check the latest release of the following operating systems for details:
iRMX I Operating System
iRMX II Operating System
UNIX* System V/386 Operating System

COMPREHENSIVE DEVELOPMENT AND OPERATING SYSTEM SUPPORT

Operating system support includes the iRMX II Real-Time operating system and UNIX* System V/386. The iRMK I real time kernel is available for 32-bit embedded applications. All three—iRMX, iRMK and UNIX operating systems include MULTIBUS II transport for full message passing support. To ease MULTIBUS II modules development, Intel offers both the iRMX and UNIX versions of the System 520 Development System which can support on-target and/or cross-hosted software development in one chassis.

WORLD WIDE SERVICE AND SUPPORT

Should this or any Intel board ever need service, Intel maintains a world wide network of service and repair facilities to keep you and your customers up and running. For unique applications requiring customization of our products, the Intel Systems Group is available to modify, integrate and test Intel boards and system components to your requirements.

INTEL QUALITY—YOUR GUARANTEE

All MULTIBUS II I/O products are designed and manufactured to meet Intel's high quality standards. Intel quality is then verified by rigorous testing in our state-of-the-art Environmental Test Laboratory.

*UNIX is a trademark of AT&T in the U.S.A. and other countries.

Product	Description	Manual Number
MULTIBUS®II SILICON PRODUCTS		
82389	Message Passing Coprocessor	176526
	Datasheet for 82389 Message Passing Coprocessor	290145
MPI	MULTIBUS II Peripheral Interface	
MIX DEVELOPMENT KIT		
MIX386020-1	MIX baseboard w 1MB	503353
MIX386020-1F01	MIX baseboard w 1MB + 1MB module	503353
MIX386020-1F04	MIX baseboard w 1MB + 4MB module	503353
MIXMDKIT-1	Kit with baseboard w 1MB	500731
MIXMDKIT-1F01	Kit with baseboard w 1MB + 1MB module	500731
MIXMDKIT-1F04	Kit with baseboard w 1MB + 4MB module	500731
MIXMDKIT-1S	Same as MIXMDKIT-1 w SE support	500731
MIXMDKIT-1F01S	Same as MIXMDKIT-1F01 w SE support	500731
MIXMDKIT-1F04S	Same as MIXMDKIT-1F04 w SE support	500731
MIX EXPANSION MODULES		
MIX 450	MIX Terminal Controller Module	500799
MIX 420	MIX WAN Module	500798
MIX 560	MIX Ethernet Module	459622
MIX MOD1	Test Module	
MIX MOD2	Breadboard Module	
MIX MOD3	Debug Module	
MIX SC10	Ten MIX Stacking Connectors	
FIRMWARE DEVELOPMENT		
MSABASEFDP	Firmware Development Package	Included
TERMINAL CONTROLLERS		
SBCMPI450	MPI-based terminal controller	502200
SBC186450	Mid-range terminal controller	502238
MIX386450-1	MIX Terminal Controller with 1MB	503353 + 500799
MIX386450-1F01	MIX Terminal Controller with 2MB	503353 + 500799
MIX386450-1F04	MIX Terminal Controller with 5MB	503353 + 500799
WIDE AREA NETWORK CONTROLLERS		
SBC186410	Mid-range WAN board	148941
MIX386420-1	MIX WAN board with 1MB	503353 + 500798
MIX386420-1F01	MIX WAN board with 2MB	503353 + 500798
MIX386420-1F04	MIX WAN board with 5MB	503353 + 500798
LOCAL AREA NETWORK/ETHERNET CONTROLLERS		
SBC186530	Mid-range Ethernet board	149226
MIX386560-1	MIX Ethernet Board with 1MB	503353 + 459622
MIX386560-1F01	MIX Ethernet Board with 2MB	503353 + 459622
MIX386560-1F04	MIX Ethernet Board with 5MB	503353 + 459622
PERIPHERAL CONTROLLERS		
SBC186224A	Multi-peripheral controller Subsystem	138272
SBC386258SM01	Single-ended SCSI Controller w 1 MB	149861
SBC386258SM04	Single-ended SCSI Controller w 4 MB	149861
SBC386258DM04	Differential SCSI Controller w 4 MB	149861
PARALLEL CONTROLLERS		
SBCMPI519	Digital I/O Board	502201

Product	Description	Manual Number
SOFTWARE PRODUCTS		
INA960J	Networking software	462250
X.25	Communications software	
OTHER MULTIBUS®II TECHNICAL LITERATURE		
	MIX Module Design Specification	500729
	Ap Note On: "Simple I/O Design Example Using. . . MIX. ."	281004
	Interconnect Interface Specification	149299-001
	MULTIBUS II Transport Protocol Specification	149247-002
	Initialization and Diagnostics	454077-001
	Bootstrap	455975-001
	A MULTIBUS II OVERVIEW, Article Reprints and Technical Papers	280684-002
ANSI/IEEE 1296	Order from: IEEE, 345 E. 47th Street, NY, NY 10017	

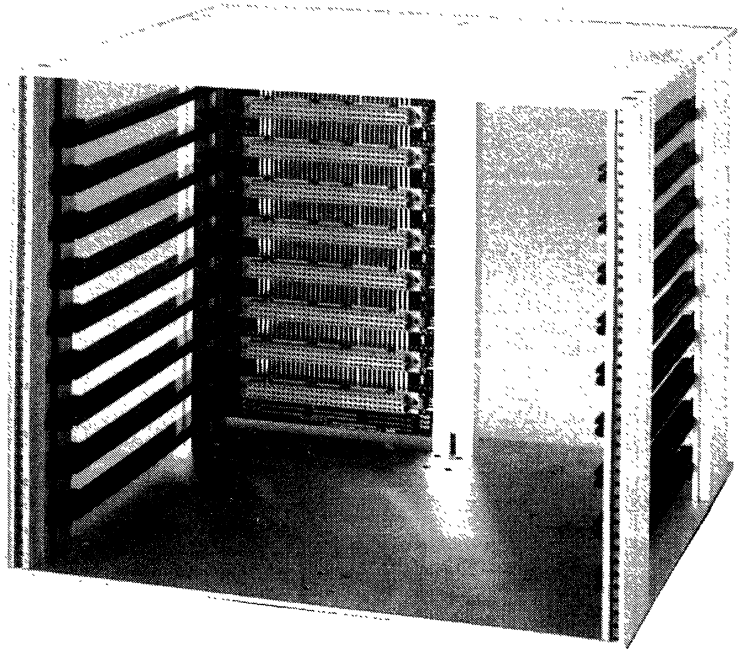
MULTIBUS® II System Packaging and Development Accessories

4

iSBC® PKG/606
iSBC PKG/609
MULTIBUS® II CARDCAGE ASSEMBLIES

- **Available in Two Sizes to Hold Up to 6 or 9 MULTIBUS® II Boards**
- **Designed to Mount Inside a Chassis or Other Enclosure**
- **Accommodates Intel iSBC® PKG/902 and iSBC® PKG/903 2 and 3 Slot iLBX™ II Backplanes**
- **All Lines Fully Terminated per the iPSB MULTIBUS II Specification**
- **Assembly Uses Aluminum Extrusion Construction for Strength and Rigidity**
- **Uses a 6 Layer Parallel System Bus (iPSB) Backplane**

The iSBC PKG/606/609 series of cardcages are designed to mount and interconnect up to 6 or 9 MULTIBUS II boards for small to medium size advanced MULTIBUS II microcomputer systems. The cardcages are compact in size and easily mount in standard or custom enclosures. Extra-wide support extrusions and heavy duty endplates help make the iSBC PKG/606/609 cardcage assemblies especially suited for installation in systems located in high vibration or high shock environments. Installed in the cardcage assembly is a 6 layer iPSB backplane that utilizes separate power and ground planes and fully terminates all signal lines. This layout minimizes system noise and ensures reliable operation even in a fully loaded, multiprocessor-based system.



280075-1

September 1986
Order Number: 280075-002

FUNCTIONAL DESCRIPTION

Mechanical Features

The cardcages accommodate up to 6 (iSBC PKG/606) or 9 (iSBC PKG/609) MULTIBUS II boards spaced at 0.8 inch centers. The assemblies are designed to hold "double high" (6U) Euro form-factor boards (233.4 mm high x 220 mm deep) or a mixture of "single high" (3U) and "double high" boards using additional hardware (not supplied). Each installed board is held in place by two screws supplied as part of the board retainer hardware.

The cardcage frame is built using five support extrusions and two aluminum end plates as shown in figure 1. Both cardcages are 10.5" wide and 10.1" deep and vary in height according to model (see specifications section).

The cardcages are designed to mount inside chassis or other enclosures and may be installed so that the MULTIBUS II boards load either horizontally or vertically in the unit. All assembly hardware is countersunk allowing the cardcages to be mounted flush against any internal chassis surface.

A Parallel System Bus (iPSB) backplane is mounted to the P1 side of the assembly, and one or more iLBX™ II backplanes (not supplied) can be mounted to the P2 side.

Electrical Features

The iPSB backplane uses a 6 layer design with separate power and ground layers and a signal routing scheme which minimizes ringing, crosstalk, and capacitive loading on the bus. Mounted on the backplane are 6 or 9, 96-pin, female DIN connectors (depending on model), bus termination resistors, decoupling capacitors, and power terminals. Press-fit technology is used throughout. The PC board is UL recognized for flammability. The card cages themselves are UL recognized components.

Single In-line Package (SIP) style resistors are used to terminate all address, clock, data, and control lines. Each termination consists of two resistors which connects the line to +VCC and ground. Different size resistors are used according to the type of driver connected to the line in an operating system.

The DIN type connectors are female, 96 pins, fully gold plated, and meet IEC standard 603-2-IEC-C096F. The connectors are mounted on 0.8" centers to match Intel's iPSB (Parallel System Bus) MULTIBUS II backplanes and are keyed to ensure proper mating to the MULTIBUS II board. The connector can provide up to 9 amps of current at +5V to each MULTIBUS II board in addition to the current available over the iLBX II backplane.

Screw terminals on the backplane are provided for connection to +5V, ±12V power and ground. In addition, an extra +5V terminal is provided for connection to a backup battery for memory protection during power fail conditions. These terminals, each of which can handle up to 25 amps of current at 55°C, provide a simple and highly reliable connection method to the system power supply.

The first slot position is designed to accept the Central Services Module (CSM) MULTIBUS II board. All other slots can accept any combination of MULTIBUS II boards.

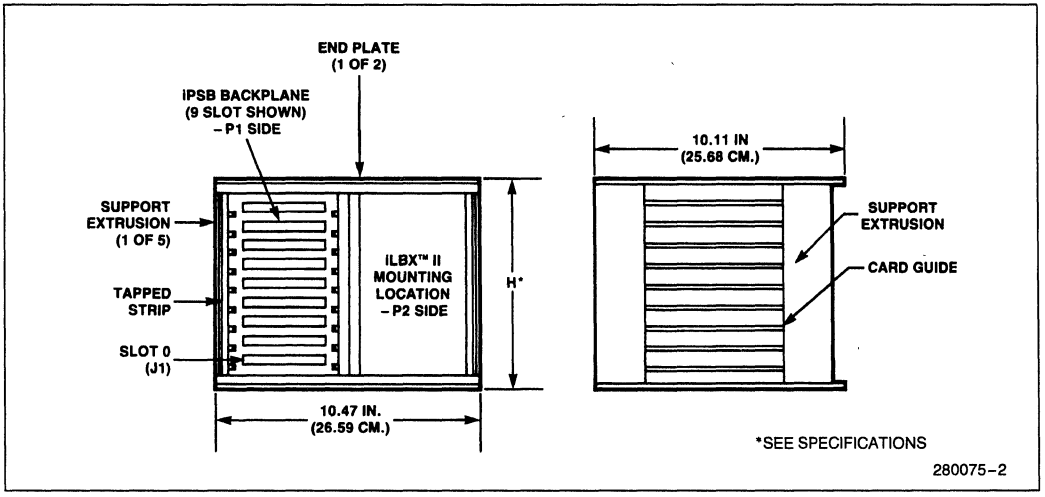


Figure 1. Cardcage Assembly Dimensions (iSBC® PKG/609 shown)

SPECIFICATIONS

Mechanical

Specification	iSBC® PKG/606 Cardcage	iSBC® PKG/609 Cardcage
Board Capacity	6	9
Dimensions Height	15.20 cm (5.98 in.)	21.20 cm (8.38 in.)
Width	26.59 cm (10.47 in.)	26.59 cm (10.47 in.)
Depth	25.93 cm (10.21 in.)	25.93 cm (10.21 in.)
Weight	4 lbs. (1.8 kg)	5 lbs. (2.3 kg)
Board Spacing	0.8 in. (20.3 cm)	
Mounting Hole Locations	See Figure 2	
Construction Materials, Cardcage Frame	Aluminum extrusions and end plates, nylon card guides	
Construction Method iPSB Backplane	Six layer backplane with separate VCC and ground layers; all connectors, power terminals, and resistor/capacitor sockets are press-fit into the backplane	
Connector Type	96 pin "DIN" female, gold plated, meets IEC standard 603-2-IEC-C096-F	

Electrical

iPSB Backplane— Meets Intel MULTIBUS II specification No. 146077 for board dimensions, layout, signal line termination, and transmission characteristics

Power Connections— Type: Screw terminal block, AMP P/N 55181-1, Winchester P/N 121-25698-2, or equivalent

Quantity of Power Terminals and Current Rating:

Voltage	iSBC® PKG/606 Cardcage		iSBC® PKG/609 Cardcage	
	Quantity	Current (amps)	Quantity	Current (amps)
+5	3	54	4	81
+12	1	12	1	18
−12	1	12	1	18
+5BB	1	12	1	18
GND	4	78	5	135

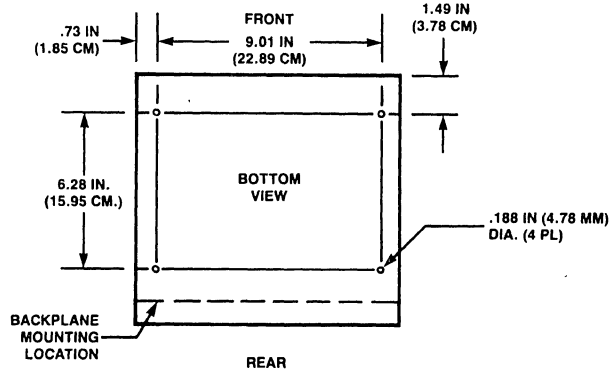


Figure 2. Mounting Hole Locations

Mating Connection: No. 6 locking spade or ring tongue lug

Maximum current available per slot:

Voltage	Current
+ 5V	9A
+12V	2A
−12V	2A
+5BB	2A

Operating Environment:
0–55°C (at 25 amps per power terminal);
0–70°C (at ≤ 18 amps per power terminal);
0% to 95% relative humidity, non-condensing;
0–10,000 ft. altitude.

Reference Manual— MULTIBUS II Cardcage Assembly and iLBX II Backplane User's Guide, P/N 146709-001 (supplied).

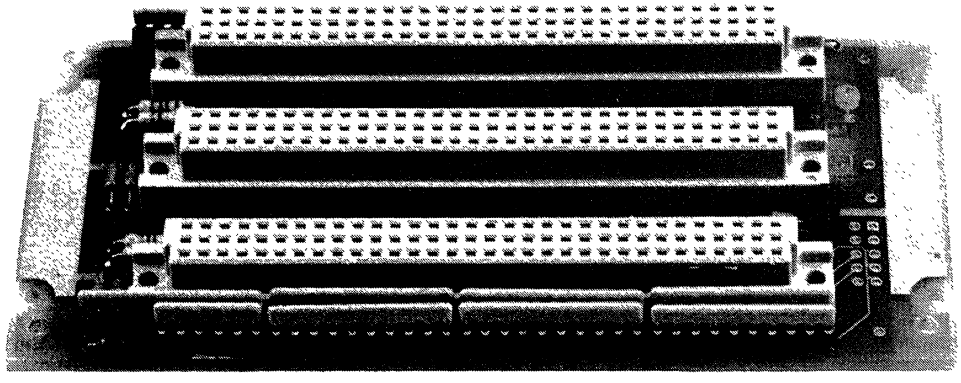
ORDERING INFORMATION

Part Number	Description
iSBC PKG/606	6 slot MULTIBUS II Cardcage Assembly
iSBC PKG/609	9 slot MULTIBUS II Cardcage Assembly

iSBC® PKG/902
iSBC® PKG/903
MULTIBUS® II iLBX™ II BACKPLANES

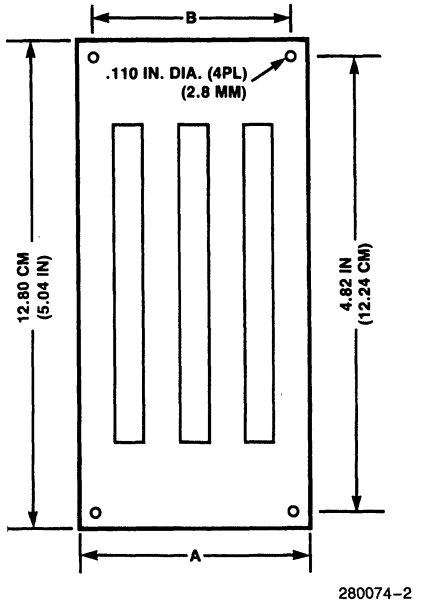
- **Provides iLBX™ II Interconnect for Fastest CPU/Memory Data Transfers**
- **Designed to Mount in MULTIBUS® II Cardcage Assemblies**
- **Meets All Electrical and Mechanical Requirements of the MULTIBUS® II Specifications**
- **Uses a 6 Layer, Fully Terminated Backplane**
- **Includes a 10 Pin Connector for BITBUS™ Applications**
- **Available in 2 Slot (iSBC® PKG/902) and 3 Slot (iSBC® PKG/903) Sizes**

The iSBC PKG/902 and iSBC PKG/903 series of iLBX II backplanes are designed to mount on the P2 side of Intel's MULTIBUS II cardcage assembly or other double Euro (6U) cardcage. One or more backplanes may be installed in a system to allow high speed data transfers between the CPU and memory boards installed in the system. The iLBX II backplane uses a 6 layer PCB with separate power and ground planes and full termination on all signal lines. This design minimizes system noise and ensures reliable operation in all applications.



280074-1

October 1986
Order Number: 280074-002



Dimensions		A	B
iSBC PKG/902	IN	1.55	.80
	CM	3.94	2.03
iSBC PKG/903	IN	2.35	1.60
	CM	5.97	4.06

Figure 1. iLBX™ II Board Dimensions (iSBC® PKG/903 Shown)

FEATURES

Mechanical and Electrical

The iSBC PKG/902 and iSBC PKG/903 iLBX II backplanes use a 6 layer printed circuit board (PCB) with separate power and ground layers and a signal lead routing scheme which minimizes ringing, crosstalk, and capacitive loading on the bus. Mounted on the PCB are two (iSBC PKG/902) or three (iSBC PKG/903) 96 pin DIN connectors, one 10-pin BITBUS connector, terminating resistors, decoupling capacitors, and power terminals. The resistors and capacitors are mounted into sockets, and all parts are press-fit into the backplane. The PCB is UL recognized for flammability.

Single In-line Package (SIP) style resistors are used to terminate all address, clock, data, and control lines. Each termination consists of two resistors which connects the line to +VCC and ground. Different size resistors are used according to the type of driver connected to the line in an operating system. The SIP style resistors help make the board compact in size and allows the designer to mount several backplanes directly adjacent to one another in a system without having to skip slots.

Mounted on the rear of the backplane is a 10-pin BITBUS connector. This connector serves as the serial communication interface for any iSBX 344 BITBUS controller boards installed in the system.

The DIN type connectors are female, 96 pins, fully gold plated, and meet IEC standard 603-2-IEC-C096F. The connectors are mounted on 0.8″ centers to match Intel's iPSB (Parallel System Bus) MULTIBUS II backplanes and are keyed to ensure proper mating to the MULTIBUS II board. The connector can provide up to 6 amps of current at +5V to each MULTIBUS II board in addition to the current available over the Parallel System Bus backplane.

Screw terminals on the backplane are provided for connection to +5V power and ground. These terminals, each of which can handle up to 25 amps of current, provide a simple and highly reliable connection method to the power supply.

SPECIFICATIONS

Mechanical and Environmental

Connector Spacing: 20.3 cm (0.8 in)

Number of Slots: iSBC PKG/902: 2 slots
iSBC PKG/903: 3 slots

Board Dimensions: See Figure 1

Weight: iSBC PKG/902—0.2 kg (8 oz)
iSBC PKG/903—0.3 kg (12 oz)

Connectors:

DIN: 96-pin female, gold plated, meets IEC standard 603-2-IEC-C096-F

BITBUS: 10-pin male, gold plated, T&B Ansley 609-1012M, or equivalent

Constructed Method: Six layer backplane with separate VCC and Ground layers
All connectors, power terminals, and resistor/capacitor sockets are press-fit into the backplane

Mounting Hole Location: See Figure 1

Operating Environment: 0°C–70°C ambient temperature; 0% to 90% relative humidity, non-condensing; 0 ft.–10,000 ft. altitude

Electrical

Backplane Electrical Characteristics and Line Terminations:	Per Intel MULTIBUS II specification 146077, Sec. II, iLBX II

Power Connections

Type: Screw terminal block: AMP P/N 55181-1; Winchester P/N 121-25698-2; or equivalent

Mating Connection: No. 6 locking spade or ring tongue lug

Quantity: 2(VCC, Ground)

Current Rating: iSBC PKG/902: 12 amps; iSBC PKG/903: 18 amps (Power and Ground)

Maximum Current Available Per Slot: 6 amps (over the iLBX II backplane)

REFERENCE MANUAL

MULTIBUS II Cardcage Assembly and iLBX Backplane User's Guide, P/N 146709-001 (not supplied)

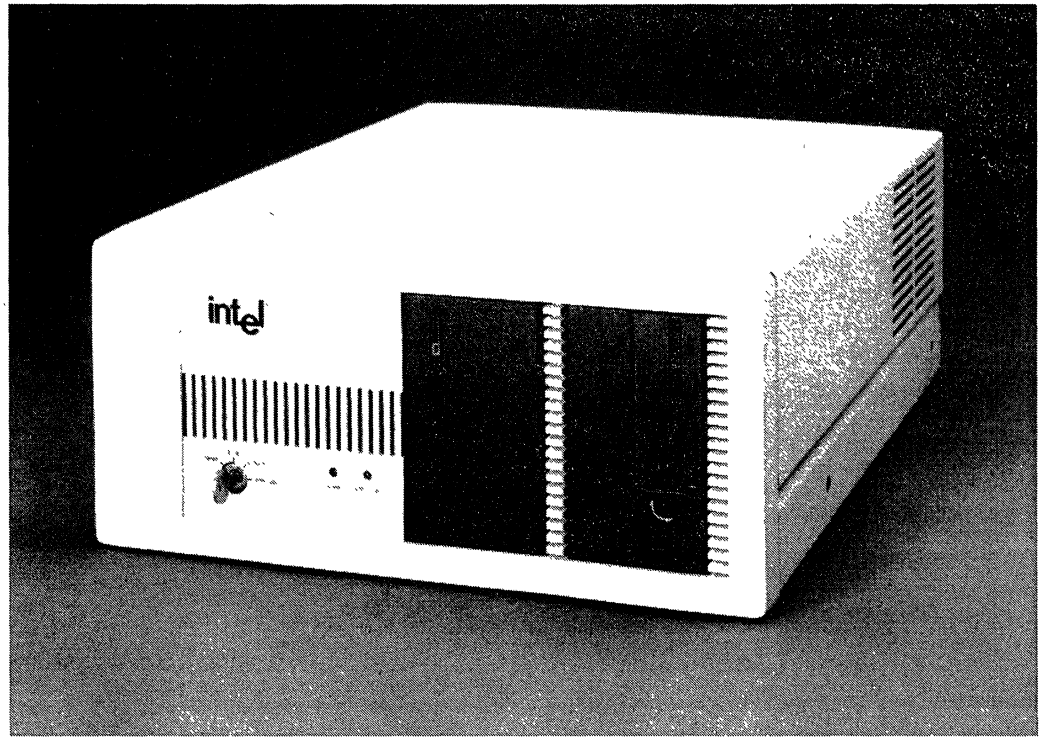
ORDERING INFORMATION

Part Number	Description
iSBC PKG/902	2 slot iLBX II Backplane
iSBC PKG/903	3 slot iLBX II Backplane

SYP 500
MULTIBUS® II SYSTEM CHASSIS

- **Full Enclosure MULTIBUS® II Design Development Tool or OEM Chassis**
- **Office and Industrial Applications**
- **3 Full Height/6 Half Height Peripheral Bays**
- **8 Slot MULTIBUS® II Cardcage Assembly**
- **3 Slot iLBX™ II Backplane**
- **535 Watt Power Supply**
- **Fully Tested: Low-Noise, Shock/ Vibration and Electrostatic Resistant**

The SYP 500 System Chassis is a MULTIBUS II design tool enabling product designers to begin work immediately on MULTIBUS II development projects. It is also ideal for OEM applications. Two front mounted LEDs indicated "Power On" and "Status" (PSB busy) while a keyswitch provides external "reset" capabilities for the chassis. The voltage selector, power-on switch and cardcage opening are located in the rear of the chassis. Three peripheral bays, two of which are accessible from the front of the chassis, support up to three industry standard 5.25″ full-height or six half-height peripherals. An eight slot cardcage, Parallel System Bus and iLBX II backplane assembly are integrated with a 535 Watt power supply.



280153-1

October 1989
Order Number: 280153-002

FUNCTIONAL DESCRIPTION

Mechanical Features

Intel's SYP 500 MULTIBUS II Chassis is a full enclosure, off-the-shelf design development tool and OEM chassis. Designers and systems integrators can integrate their MULTIBUS II board set with tape, Wini or floppy peripherals into a complete system. The SYP 500 has three full-height 5.25" peripheral bays. Peripheral power cables, office and industrial environment cooling, and peripheral mounting brackets for industry standard full- or half-height peripherals are provided with the chassis. Access via the front panel allows two of the bays to be configured with removable media peripherals e.g. tape and floppy drives.

This chassis includes an eight-slot MULTIBUS II cardcage assembly with 0.8" centers (slot width). The cardcage is made with heavy duty endplates and extra-wide support extrusions to ensure adequate support for most applications. For industrial applications, this chassis is mountable into any 19" vertical rack.

Two backplanes are installed in the cardcage assembly: the system backplane and the auxiliary backplane. The system backplane is the Parallel System Bus (iPSB) for communications between up to eight MULTIBUS II boards. This backplane utilizes separate power and ground planes and fully terminates all signal lines. The auxiliary backplane, on the other hand, provides direct high speed interconnection between a processor board and memory boards. It contains three iLBX slots. One of these slots has a 10-pin BITBUS connector that serves as a serial interface for any iSBX 344 BITBUS controller board installed in the system. This cardcage conforms to the published MULTIBUS II specification.

Electrical Features

The SYP 500 chassis has a 535 Watt switching power supply with selectable AC power input of 115V or 220V at 47 Hz–63 Hz. The AC input power is externally selectable with a slide switch mounted on the rear of the chassis. A power distribution board is installed in the chassis to allow easy connection to all peripheral bays through six plugs mounted on the power distribution board.

The chassis has been fully tested to ensure low-audible noise emission, resistance to electrostatic discharge and resistance to appropriate levels of vibration and shock in both office and industrial environments.

SPECIFICATIONS:

Electrical Parameters

Maximum Amperage:

Voltage	Current
+5V	75A
+12V	10A
−12V	2.5A

Designed to meet: UL 478
CSA C22.2 No. 154
FCC Class B
VDE Level B
IEC 435

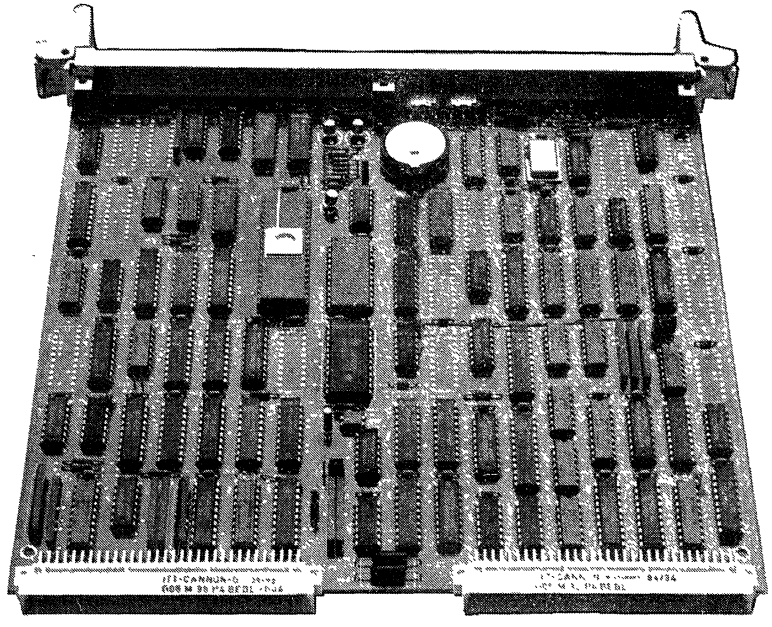
Operational Parameters

AC Power Input:	90–132 VAC or 180–264 VAC at 47 Hz–63 Hz
Operating Temperature Range:	10°C to 55°C
Storage Temperature:	−40°C to 60°C
Operation Humidity:	10% to 85% relative, non-condensing

iSBC® CSM/001*
CENTRAL SERVICES MODULE

- **iSBC® CSM/001 Central Services Module Integrates MULTIBUS® II Central System Functions on a Single Board**
- **MULTIBUS® II Parallel System Bus Clock Generation for all Agents Interfaced to the MULTIBUS II PSB Bus**
- **System-wide Reset Signals for Power-up, Warm Start, and Power Failure/ Recovery**
- **System-wide Time-out Detection and Error Generation**
- **Slot I.D. and Arbitration I.D. Initialization**
- **MULTIBUS II Interconnect Space for Software Configurability and Diagnostics**
- **Built-In Self Test (BIST) Power-up Diagnostics with LED Indicator and Error Reporting Accessible to Software via Interconnect Space**
- **General Purpose Link Interface to Other Standard (MULTIBUS I) or Proprietary Buses**
- **Time-of-day Clock Support with Battery Back-up on Board**
- **Double-high Eurocard Standard Form Factor, Pin and Socket DIN Connectors**

The iSBC CMS/001 Central Services Module is responsible for managing the central system functions of clock generation, power-down and reset, time-out, and assignment of I.D.s defined by the MULTIBUS II specification. The integration of these central functions in a single module improves overall board area utilization in a multi-board system since these functions do not need to be duplicated on every board. The iSBC CSM/001 module additionally provides a time-of-day clock and the general purpose link interface to the other standard (MULTIBUS I) or proprietary buses.



280070-1

*The iSBC® CSM/001 is also manufactured under product code piSBC® CSM/001 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280070-003

FUNCTIONAL DESCRIPTION

Overall

The iSBC CSM/001 Central Services Module integrates MULTIBUS II central system functions on a single board. Each MULTIBUS II system requires management of these central system functions as defined in the MULTIBUS II specification. Figure 1 illustrates a typical multiprocessing MULTIBUS II system configuration. To perform its central system functions, the iSBC CSM/001 Central Services Module has a fixed slot I.D. and location in the backplane. The iSBC CSM/001 board additionally provides an interface to the MULTIBUS I Link board and a time-of-day clock.

Architecture

The iSBC CSM/001 board is functionally partitioned into 6 major subsystems. The Central System Wide Control subsystem includes MULTIBUS II PSB bus clock generation and system wide reset signal generation. The Time-Out Control subsystem provides system wide time out detection and error generation. The System Interconnect Space subsystem controls I.D. initialization and software configurable interconnect space. The Link Board interface subsystem provides an interface to the MULTIBUS I Link board or links to other buses. The last two subsystems are of the Time-of-Day clock and the PSB bus interface. These areas are illustrated in Figure 2.

CENTRALIZED SYSTEM-WIDE CONTROL SUBSYSTEM

Parallel System Bus Clock Generation

The CSM generates the Parallel System Bus clocks. The Bus Clock (BCLK*) 10 MHz signal and the Constant Clock (CCLK*) 20 MHz signal are supplied by CSM to all boards interfaced to the Parallel System Bus. These boards use the Bus Clock 10 MHz signal for synchronization, system timing, and arbitration functions. The Constant Clock is an auxiliary clock. The frequency of the Bus Clock and Constant Clock can be halved via jumpers for diagnostic purposes.

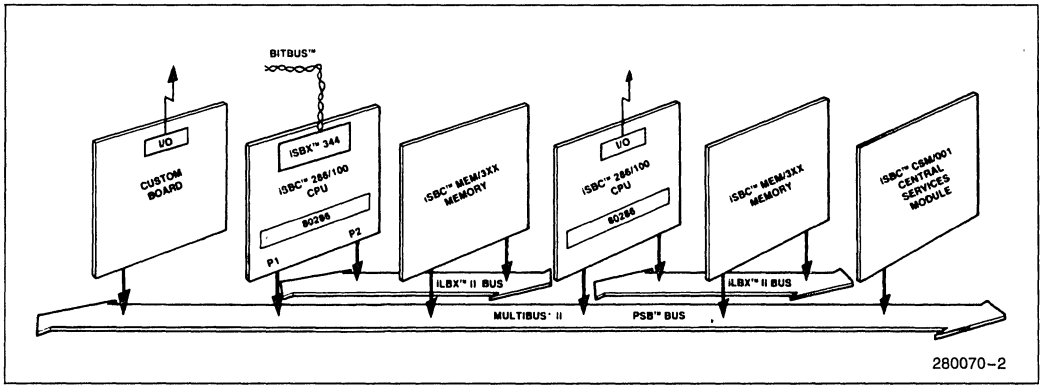


Figure 1. Typical MULTIBUS® II System Configuration

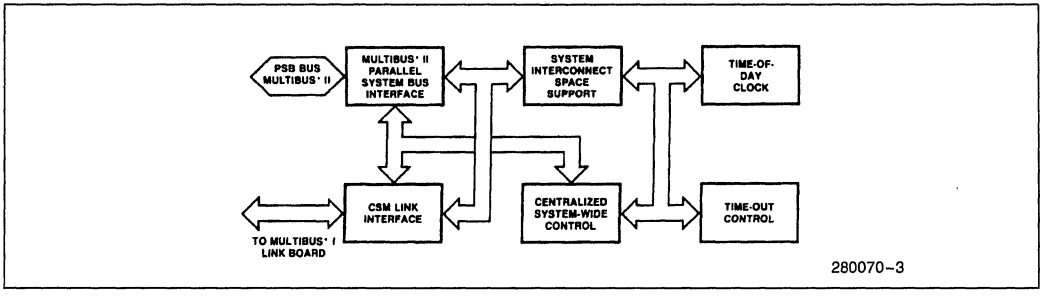


Figure 2. Block Diagram of iSBC® CSM/001 Board

Reset Control and Power-Fail/ Recovery

The CSM sends a system-level reset/initialization signal to all boards interfaced to the Parallel System Bus. The CSM assigns slot I.D. and arbitration I.D. to these boards during this initialization process. It provides this signal upon pressing of the reset switch, restoration of system power or a software request for reset received via the CSM interconnect space. The reset switch may be jumper-configured to cause a power-up or warm reset, with cold reset the default configuration. The reset switch is located on the front panel. Additionally, warm reset and cold reset signals can be input through the P2 connector.

The CSM power supply interface is accomplished via the ACLO input of the P2 connector. ACLO is an open collector input from the power supply which provides advance warning of imminent power fail. If battery backup is not required, a jumper is provided on the CSM to disable the power fail signal ACLO.

TIME-OUT SUBSYSTEM

The TIMOUT* (Time-Out) signal is provided by the CSM whenever it detects the failure of a module to complete a handshake. This TIMOUT* signal is received by all boards interfaced to the PSB bus and may be disabled via the interconnect space.

INTERCONNECT SUBSYSTEM

The CSM Interconnect subsystem provides arbitration I.D., and slot I.D. initialization, software configurable interconnect space, and on-board diagnostics capability.

At reset, the CSM supplies each board interfaced to iPSB bus with its slot I.D. and its arbitration I.D. The slot I. D. assignment allows user or system software to address any board by its physical position in the backplane.

The interconnect space has both read-only and software configurable facilities. The read-only registers hold information such as vendor number and board type, so that this information is available to the system software. The CSM software configurable interconnect space allows write operations to support board configuration and diagnostics under software control. The CSM also uses interconnect space for system wide functions such as providing a time/date record (from time-of-day clock), software access to diagnostics and software control of the system wide functions.

BUILT-IN-SELF-TEST (BIST) DIAGNOSTICS

Self-test/diagnostics have been built into the heart of the MULTIBUS II system. These confidence tests and diagnostics improve reliability and reduce manufacturing and maintenance costs. LED 1 (labeled BIST) is used to indicate the status of the Built-In-Self-Test. It is turned on when the BIST starts running and is turned off when the BIST completes successfully. In addition, all error information is recorded in interconnect space so it is accessible to software for error reporting.

The Built-In-Self-Tests performed by the on-board microcontroller at power-up or at software command are:

1. PROM Checksum Test—Verifies the contents of the 8751 microcontroller.
2. RAM Test—Verifies that each RAM location of the 8751 microcontroller may store 0's and 1's by complementing and verifying twice each RAM location.
3. Real Time Clock Chip RAM Test—Verifies that reads and writes to the RAM locations on Real Time Clock Chip are functional.
4. Real Time Clock Test—Reads and writes all RAM locations of the RTC chip. Not run at power-up due to destructive nature.
5. Arbitration/Slot I.D. Register Test—Verifies that arbitration and slot I.D.s can be read and written from on-board.
6. 8751 Status Test—Verifies that input pins of the 8751 are at correct level.
7. Clock Frequency Test—Tests accuracy of Real Time Clock to 0.2% against bus clock.

CSM LINK INTERFACE

The CSM Link Interface and the MULTIBUS I iSBC LNK/001 board provides a bridge between MULTIBUS I and MULTIBUS II systems. Hybrid systems can be built for development or target. The CSM Link Interface uses the P2 connector on the iSBC CSM/001 module for transferring commands and data from MULTIBUS II to a MULTIBUS I Link board. The MULTIBUS I Link board (iSBC LNK/001) is purchased separately from the iSBC CSM/001 board and includes the cable which connects the iSBC CSM/001 board and the MULTIBUS I Link board (see Figure 3).

The CSM Link Interface supports 8- or 16-bit transfers via a 16-bit address/data path. The iSBC LNK/001 board resides in the MULTIBUS I system and provides a memory and I/O access window to MULTIBUS I from the MULTIBUS II Parallel System Bus. Only one iSBC LNK/001 board can be connected to the iSBC CSM/001 module.

TIME-OF-DAY CLOCK SUBSYSTEM

The Time-Of-Day Clock subsystem consists of a clock chip, battery, and interface circuitry. The clock provides time keeping to 0.01% accuracy of fractions of seconds, seconds, minutes, hours, day, day of week, month, and year. This information is accessible via the interconnect space. The battery backup for the clock chip provides 2 years of operation.

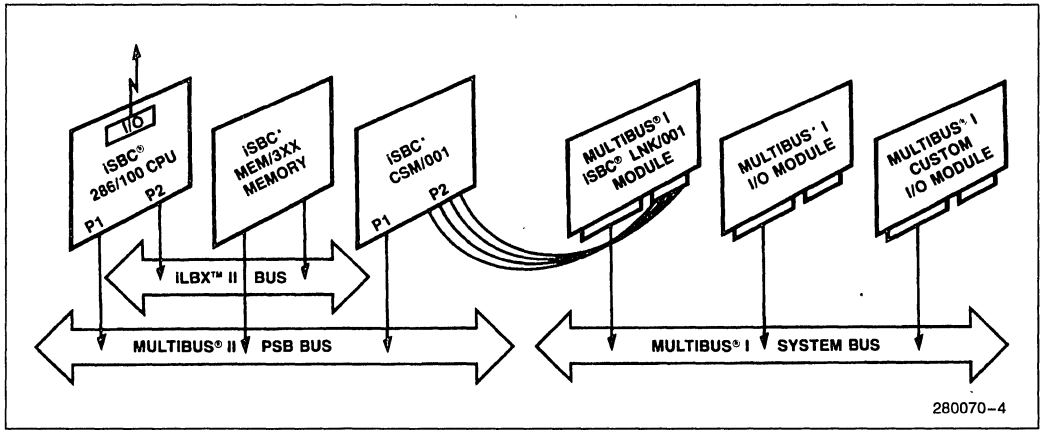


Figure 3. iSBC® CSM/001 Link Interface

SPECIFICATIONS

System Clocks

BCLK* (Bus Clock)	10 MHz
CCLK* (Constant Clock)	20 MHz
LCLK* (Link Clock)	10 MHz

Jumper option available to divide these frequencies in half

Link Cable

The Link cable uses a 64-conductor ribbon cable for interconnecting the CSM board to the Link Board. The maximum length for the cable is 1 meter.

Interface Specifications

Location	Function	Part #
P1	PSB Bus	603-2-IEC-C096F
P2	Link and Remote Services	603-2-IEC-C064-F

PHYSICAL DIMENSIONS

The iSBC CSM/001 board meets all MULTIBUS II mechanical specifications as presented in the MULTIBUS II specification.

Double-High Eurocard Form Factor:

Depth:	220 mm. (8.7 in.)
Height:	233 mm. (9.2 in.)
Front Panel Width:	20 mm. (0.78 in.)
Weight:	4820 gm. (16.5 oz.)

ENVIRONMENTAL REQUIREMENTS

Temperature: (inlet air) at 200 LFM airflow over boards
Non-operating: −40 to +70°C
Operating: 0 to +55°C

Humidity: Non-operating: 95% RH @ 55°C
Operating: 90% RH @ 55°C

POWER REQUIREMENTS

Voltage (volts)	Current (amps)
+5	6A (max.)
+5 VBB	1A (max.)

BATTERY CHARACTERISTICS

3V nominal voltage; capacity of 160 milliamp hours minimum.

BATTERY DIMENSIONS

Outside dimension	20 mm–23 mm
Height	1.6 mm–3.2 mm

REFERENCE MANUALS

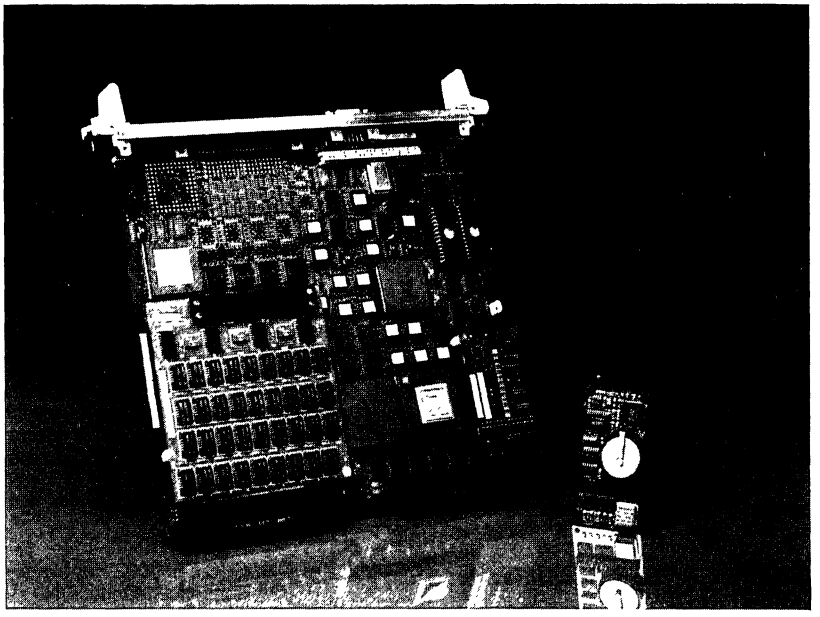
iSBC CSM/001 Board Manual (#146706-001)

Manuals may be ordered from any Sales Representative, Distributor Office, or from the Intel Literature Department, 3065 Bowers Ave., Santa Clara, CA 95051.

ORDERING INFORMATION

Part Number	Description
iSBC CSM/001	MULTIBUS II Central Services Module

MULTIBUS® II CENTRAL SERVICES MODULE*



COST REDUCED CENTRAL SERVICES MODULE

The iSBC® CSM/002 module is a small, surface mount circuit board which performs all central service module (CSM) functions as required by the IEEE/ANSI 1296 MULTIBUS® II specification. This credit card sized module mounts on a compatible base board such as the iSBC 386/258, iSBC 386/133, or iSBC 486/125DU single board computer. The combined host board and CSM module require only one card slot. The small size and high functionality of the iSBC CSM/002 module is achieved by taking advantage of silicon support for CSM functions on the MPC (Message Passing Coprocessor) bus interface component. This module reduces system cost while remaining software compatible with the previous iSBC CSM/001 board.

FEATURES

- Full IEEE/ANSI 1296 Compliance for CSM Functions:
  - Arbitration and Slot ID Initialization
  - BCLK and CCLK Generation
  - PSB Bus Timeout Monitoring
  - Reset Sequencing for Warm and Cold Resets
  - Power Fail Indication and Recovery
- Software compatible with the iSBC CSM/001 board, but saves a card slot
- Battery Backup Time-of-day Clock
- Slot 0 Detection Circuit
- Clock Based Alarm Function for Periodic Interrupt
- 28 Bytes Non-volatile RAM
- Chassis ID for Crate-to-Crate Addressing
- Low Battery and Oscillator Failed Warnings

* The MULTIBUS II Central Services Module, iSBCCSM002 is also manufactured under product code pSBCCSM002 by Intel of Puerto Rico, Inc and sSBCCSM00S by Intel Singapore, Ltd

September, 1989
Order Number 280694-001

WHAT IS A CENTRAL SERVICES MODULE?

The Central Services Module centralizes a variety of bus management tasks in a Multibus® II based system, reducing system overhead:

- Sequencing of reset signals on the PSB backplane.
- Assignment of card slot and arbitration IDs.
- Supplies a 10 MHz. system-wide clock signal (BCLK).
- Monitors the PSB bus for time outs, and signals a bus error when a parity error is detected.

FULL IEEE 1296 COMPLIANCE

The iSBC CSM/002 module meets all timing requirements for Central Services Module functions according to the MULTIBUS® II specification. This ensures reliable, clean system clock signals and correct reset sequencing for system power-on, power-fail, and front panel warm and cold resets. When used in a system where the power supply is capable of generating an ACLO indicator, the CSM module will signal a non-maskable interrupt to the host CPU shortly before the power goes down.

COMPACT SIZE: NO LONGER REQUIRES A SEPARATE CARD SLOT

The iSBC CSM/002 module reduces total system cost by supporting all CSM functions in an inexpensive, credit card sized module. This module mounts component side down onto compatible baseboards like the iSBC 386/258, iSBC 386/133, or iSBC 486/125DU single board computers which have built-in CSM connectors. The combined host board and CSM module occupy only one card slot. The iSBC CSM/002 module is fully software compatible with the earlier iSBC CSM/001 board, and is a direct replacement. Only one CSM module is required per system.

TIME OF DAY CLOCK FUNCTIONS

A battery backed up time-of-day clock is supplied on the iSBC CSM/002 module. This feature is software compatible with the existing Time and Date commands supported by various Intel supplied operating systems. In addition a new periodic alarm function is now available. This feature allows the user to generate an interrupt to the local processor based on the system clock. Intervals can be selected ranging from one second to one year. One example of how to use this might be to schedule a disk backup to tape at 1:00am on Friday of each week.

NON-VOLATILE RAM FUNCTIONS

A two byte chassis ID is stored in an interconnect register in order to identify a particular backplane segment in a network which consists of a large number of nodes distributed in multiple chassis. In addition, there are 28 bytes of user definable non-volatile RAM available. One application might be for a bootstrap password to prevent unauthorized access to a system.

PROGRAMMATIC INTERFACE

All access to the above functions is via a set of function records contained in interconnect address space. These registers are resident on the host in slot zero, but are accessible to any agent on the PSB. This allows other boards to query reset status, bus errors, system time, NVRAM contents, and many other centralized functions.

WORLDWIDE SUPPORT AND SERVICE

Assistance in developing and supporting MULTIBUS® II applications is available through Intel's network of field application engineers, system engineers, customer training centers and service centers.

INTEL QUALITY—YOUR GUARANTEE

The iSBC CSM/002 module is designed and manufactured in accordance with Intel's high quality standards. Quality is verified by rigorous testing in Intel's state-of-the-art Environmental Test Laboratory.

ORDERING INFORMATION

ORDER CODE: SBCCSM002

DOCUMENTATION: iSBC CSM/002 Hardware Reference Manual P/N 459706-001

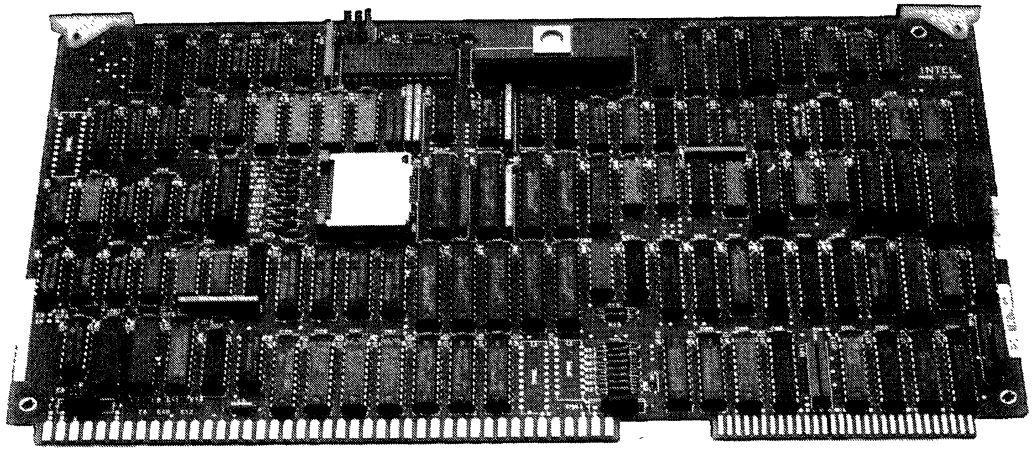
AP NOTE: "Design of a Cost Reduced Central Services Module for MULTIBUS® II"

For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada).

iSBC® LNK/001* BOARD
MULTIBUS® II TO MULTIBUS® I LINK BOARD

- **Development Vehicle Making MULTIBUS® I iSBC® Boards Accessible to MULTIBUS® II Board Designers**
- **On Board 128K Byte Dual Port DRAM Memory**
- **16M Bytes of MULTIBUS® I Memory Mapped into MULTIBUS® II Memory Space Configurable from MULTIBUS® II Interconnect Space**
- **32K Bytes of MULTIBUS® I I/O Mapped into MULTIBUS® II I/O Space Configurable from MULTIBUS® II Interconnect Space**
- **Conversion of MULTIBUS® I Interrupts to MULTIBUS® II Interrupt Messages**
- **MULTIBUS® I Form Factor Board**
- **Connects to MULTIBUS® II Central Services Module (iSBC CSM/001 Board) via a 3 Foot Flat Ribbon Cable**

The iSBC® LNK/001 board maps MULTIBUS I memory and I/O space into the MULTIBUS II iPSB bus and converts MULTIBUS I interrupts into MULTIBUS II interrupt messages. Up to 16M Bytes of MULTIBUS I memory and up to 32K Bytes of MULTIBUS I I/O is addressable from MULTIBUS II through the iSBC LNK/001 board. Additionally, 128K Bytes of dual port DRAM memory resides on the iSBC LNK/001 board for use by both MULTIBUS I and MULTIBUS II systems. MULTIBUS II OEM product designers can now speed hardware and software development efforts by using the iSBC LNK/001 board to access standard or custom MULTIBUS I products.



280135-1

*The iSBC® LNK/001 is also manufactured under product code piSBC® LNK/001 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280135-003

GENERAL DESCRIPTION

The iSBC LNK/001 board makes MULTIBUS I products accessible to MULTIBUS II designers. The iSBC LNK/001 board resides in the MULTIBUS I system and connects to the Central Services Module (iSBC CSM/001 board) via a 3 foot flat ribbon cable. The ribbon cable connects the P2 connector of the iSBC LNK/001 board to the P2 connector on the Central Services Module. The iSBC LNK/001 board supports:

a. 128K Bytes of Dual Port DRAM,

b. 16- and 24-bit addressing into 16M Bytes of MULTIBUS I memory with 8- and 16-bit data paths,

c. 8- and 16-bit addressing into 32K Bytes of MULTIBUS I I/O with 8- and 16-bit data paths,

d. MULTIBUS I interrupt to MULTIBUS II interrupt message conversions of up to eight levels of non bus-vectored interrupts via an 8259A programmable interrupt controller, and

e. initialization tests and Built-In-Self-Test (BIST) using interconnected address space.

APPLICATIONS

The primary application of the iSBC LNK/001 board is in the design development environment. The iSBC LNK/001 board allows designers to start their development efforts by leveraging existing MULTIBUS I products or to begin modular design efforts and preserve investments in custom products. In either case, the use of leverage with existing MULTIBUS I hardware and software allows designers to begin their MULTIBUS II product designs.

MEMORY AND I/O READ/WRITE SEQUENCE

The iSBC LNK/001 board establishes a master/slave relation between a MULTIBUS II system and a MULTIBUS I system. A MULTIBUS II agent requesting a memory transfer involving the iSBC LNK/001 board is directed through the CSM to the iSBC LNK/001 Dual Port memory or a MULTIBUS I slave. If the access address is within the MULTIBUS II Dual Port window, the transaction is acknowledged by the iSBC LNK/001 board and returned to the MULTIBUS II iPSB through the CSM. In the event the address is outside the MULTIBUS II Dual Port window, the transaction is directed to the MULTIBUS I system. Here the iSBC LNK/001 board enters arbitration for the MULTIBUS I system bus to complete the requested transaction. Once the iSBC LNK/001 board is the owner of the MULTIBUS I system bus, data is transferred to or from the iSBC LNK/001 board/Central Services Module connection. The MULTIBUS I slave acknowledges the transfer and the iSBC LNK/001 board passes the acknowledge on through the Central Services Module to the MULTIBUS II iPSB.

MULTIBUS II I/O operations are always directed to the MULTIBUS I I/O slaves and consequently require arbitration for the MULTIBUS I system bus.

INTERCONNECT MAPPING

The function record of the iSBC LNK/001 board, a function record within the Central Services Module interconnect template, appears as a board within a board (see Table 1). The actual iSBC LNK/001 board configuration is done through unique interconnect registers using the same slot ID as the Central Services Module. The iSBC LNK/001 function record begins at an offset of 256 from the start of the CSM template and the EOT (End Of Template) byte is attached as the last function of the iSBC LNK/001 function record.

Dual Port 128K Byte DRAM Memory

A dynamic RAM Dual Port, resident on the iSBC LNK/001 board, provides a 128K Byte media for

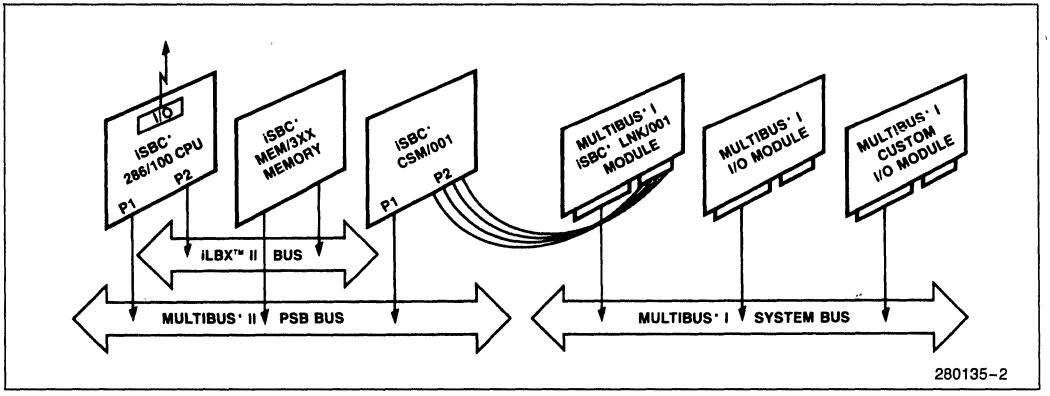


Figure 1. Sequence Diagram

MULTIBUS I and MULTIBUS II agents to pass data efficiently. With both buses sharing the Dual Port memory the need for the MULTIBUS II system to continuously arbitrate for MULTIBUS I system access is eliminated. Consequently, each bus can continue operating at its respective speed when accessing the iSBC LNK/001 Dual Port memory.

MULTIBUS® I Memory Addressability

The MULTIBUS I system views the iSBC LNK/001 Dual Port as a contiguous 128K Byte memory block mapped into the 16M Bytes of MULTIBUS I memory address space starting at the Dual Port Start Address register value. This memory block, configurable on any 64K Byte boundary within the MULTIBUS I memory address space, is set via interconnect accesses to the iSBC LNK/001 function records from the MULTIBUS II system (see Table 1). The first 16M Bytes of MULTIBUS II memory space can be mapped in the 16M Bytes of MULTIBUS I memory address space (see Figure 3).

MULTIBUS® I I/O Addressability

Up to eight 4K Byte blocks of MULTIBUS II I/O space can be mapped into MULTIBUS I I/O space

Table 1. Function Record Overview iSBC® LNK/001 Board

Offset	Description	Offset	Description
0–255	iSBC CSM/001 Header and Function Record	271	MBI Dual Port End Address
		272	MBII Dual Port Start Address
256	Board Specific Record Type	273	MBII Dual Port End Address
257	Record Length	274	MBII Memory Start Address
258	Vendor ID, Low Byte	275	MBII Memory End Address
259	Vendor ID, High Byte	276	I/O 4K Segment Control
260	Link Version Number	277	MBI Interrupt Enable
261	Hardware Revision Test Number	278	Link Interrupt 0 Destination Address
262	Link General Status	279	Link Interrupt 1 Destination Address
263	Link General Control	280	Link Interrupt 2 Destination Address
264	Link BIST Support Level	281	Link Interrupt 3 Destination Address
265	Link BIST Data In	282	Link Interrupt 4 Destination Address
266	Link BIST Data Out	283	Link Interrupt 5 Destination Address
267	Link BIST Slave Status	284	Link Interrupt 6 Destination Address
268	Link BIST Master Status	285	Link Interrupt 7 Destination Address
269	Link BIST Test ID	286	Interrupt Source Address
270	MBI Dual Port Start Address	287	Link Status Register
		288	EOT (End of Template)

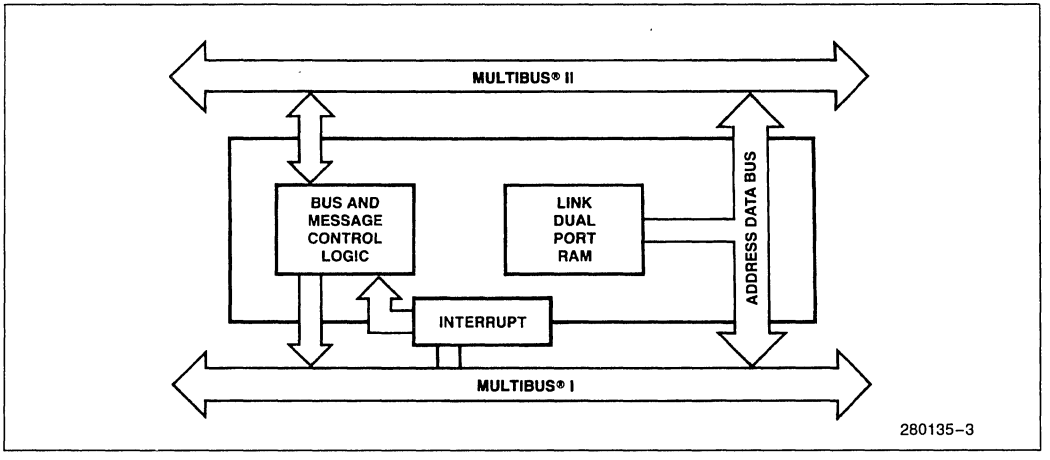


Figure 2. Link Board Dual Port Drawing

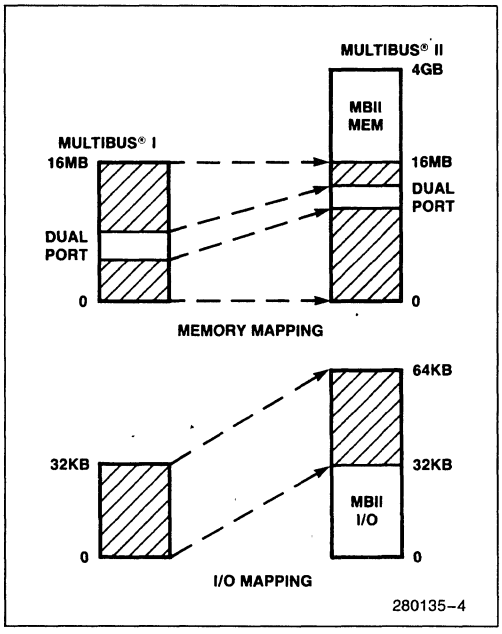


Figure 3. MULTIBUS® I Memory and I/O Mapping Diagram

(see Figure 3). MULTIBUS II I/O accesses must be from 32K Byte to 64K Byte in order to be mapped into MULTIBUS I I/O address space. These blocks are specified through an interconnect access to the "I/O 4K Segment Control" register (see Table 1). Each bit in the register represents a 4K Byte block of I/O addresses. When a bit (or bits) is set, the 4K Byte block of MULTIBUS II I/O space represented by that bit will be dedicated to MULTIBUS I I/O space.

Interrupt to Message Conversion

As the iSBC LNK/001 board receives non-bus vectored interrupts from the MULTIBUS I system, the on-board 8259A programmable interrupt controller (PIC) prioritizes the MULTIBUS I interrupts and initiates the MULTIBUS II unsolicited interrupt message generation process. Up to 8 levels of non-bus vectored interrupts are supported by the iSBC LNK/001 board.

The iSBC LNK/001 board generates the MULTIBUS II interrupt messages and is the Interrupt Source. The iSBC LNK/001 board is assigned a Source ID through interconnect space when the MULTIBUS II system is powered up or when the user programs the source ID register via interconnect space. The Interrupt Destination is the MULTIBUS II board to which the interrupt message is being sent. Each of the eight MULTIBUS I interrupt lines can be programmed to generate a unique MULTIBUS II destination address. These destination addresses are initialized through interconnect space by programming the iSBC LNK/001 Interrupt Destination Address Registers. The message source address is also configurable via interconnect space by writing to the Interrupt 0 Source Address Register with a base value. Once the base value of source Address 0 is established, Source Address 1 through 7 are set for incrementing values by the 8751A interconnect processor. The iSBC LNK/001 board recognizes MULTIBUS II Negative Acknowledge agent errors ("NACK") and performs an automatic retry algorithm.

Initialization Tests and BIST

Self test and diagnostics have been built into the MULTIBUS II system. The BIST LED is used to indicate the result of the Built-In-Self-Test and turns on when BIST starts running and turns off when it has successfully executed. BIST test failure information is recorded in the interconnect space and is accessible to software for error reporting.

PHYSICAL CHARACTERISTICS

Form Factor

The iSBC LNK/001 board is a MULTIBUS I form factor board residing in a MULTIBUS I system. Physical dimensions are identical to all standard MULTIBUS I boards.

Connection to MULTIBUS® II Bus

The iSBC LNK/001 board connects to the iSBC CSM/001 board in the MULTIBUS II system via a 60 pin conductor flat ribbon cable. The physical connection is made on the P2 connector of both the iSBC LNK/001 board and the iSBC CSM/001 board. The cable termination requirements and DC requirements for the signal drivers and receivers are detailed in the iSBC CSM/001 USERS GUIDE, Section 6.6.4. The maximum length of the cable is 3 feet. The cable and the connectors are shipped unassembled to allow user flexibility.

SOFTWARE SUPPORT

To take advantage of iSBC LNK/001 Dual Port architecture, existing software device drivers may require modification. Device driver changes depend on the specific application and vary in complexity depending upon the device driver.

SPECIFICATIONS

Word Size

16- and 24-bit Address Paths
8- and 16-bit Data Paths
Block transfers are not supported

Cable Characteristics

The cable is a 60 pin conductor flat ribbon cable with a maximum length of 3 feet. The P2 connector to the iSBC LNK/001 board is a 30/60 pin board edge connector with 0.100" pin centers, KEL-AM Part Number RF30-2853-5. The connector to the P2 DIN connector on the iSBC CSM/001 board is 3M Part Number 3338-000.

Interface Specifications

Location	Function
P1	MULTIBUS IEEE 796 System Bus
P2	Cable connection to P2 connector of iSBC CSM/001 board

PHYSICAL DIMENSIONS

The iSBC LNK/001 board meets all MULTIBUS I mechanical specifications as presented in the MULTIBUS I specification.

Depth: 17.15 cm (6.75 in.)
Height: 1.27 cm (0.50 in.)
Front Panel Width: 30.48 cm (12.00 in.)
Weight: Estimated 565 g (20 oz.)

ENVIRONMENTAL REQUIREMENTS

Temperature: Inlet air at 200 LFM airflow over boards
Non Operating: −40°C to +75°C
Operating: 0°C to +55°C

Humidity: Non Operating: 0 to 95% RH @ 55°C
Operating: 0 to 95% RH @ 55°C

POWER REQUIREMENTS

Voltage: +5V
Current: 7.14 Amps

REFERENCE MANUALS

iSBC LNK/001 Users Guide (#148756-001)

iSBC CSM/001 Users Manual (#146706-001)

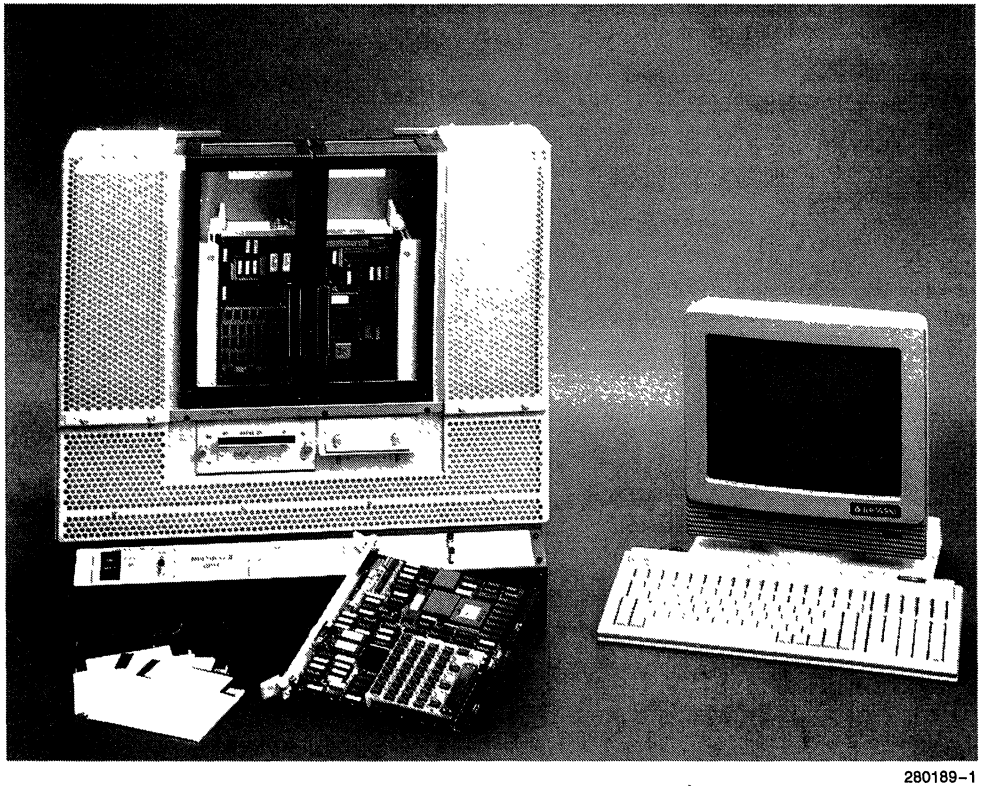
Manuals may be ordered from any Sales Representative, Distributor Office, or from the Intel Literature Department, 3065 Bowers Ave., Santa Clara, CA. 95051.

ORDERING INFORMATION

Part Number	Description
iSBC LNK/001	MULTIBUS II to MULTIBUS I iSBC LNK/001 Interface Board

MULTIBUS® II HIGH PERFORMANCE SBC GENERAL PURPOSE TEST FIXTURE (GPTF)

- Single Board Computer Tester for MULTIBUS® II Boards in a Systems Environment
- Tests up to Four MULTIBUS II Boards Simultaneously in a Range from Ambient Temperature to 70°C
  — Voltage and Temperature Margins are Software Controlled
- Multiprocessor, Multitesting Functional Tester with Totally Automated Test Sequence, Requiring Minimum Human Intervention
- Powerful Command Language for Troubleshooting and Evaluation
- One STBL (System Test Board Level) Test is Included. Additional Test Programs are Available for Intel MULTIBUS II Boards
- GPTF Includes Video Monitor for Error Message Display and Status of Testing, Also, a Comprehensive Installation Guide and Users Manual
- Bus Drawer Feature on P2 Connector Allows User Flexibility to Test Boards with Different Types of P2 Interfaces
- Available in Either USA, Japan or International Power Configuration
- Safety Features Including Thermal Cut Out at 90°C



280189-1

October 1988
Order Number: 280189-002

TESTER OVERVIEW

The MULTIBUS II General Purpose Test-Fixture (MULTIBUS II GPTF) is a state-of-the-art high performance tester used to test MULTIBUS II boards in a typical systems environment. The System Test Board Level or STBL, as it is usually referred to, is done using the MULTIBUS II GPTF. The STBL is used to validate that the iSBC board will perform in a system environment under a variety of temperature and voltage conditions. The MULTIBUS II GPTF is a fully automated tester with minimum operator intervention required. It can test from one to four boards of the same type at a time. A full range of keyboard commands are available for troubleshooting. The human interface is through the Front Panel and the CRT terminal. The MULTIBUS II GPTF requires the use of a Televideo 955 terminal which is included with the GPTF order.

The users manual is written at the operator's level and thus does not require a technician to perform tests. The users manual is written in two parts; operator's instructions and technician's troubleshooting section. An installation guide is also furnished.

The MULTIBUS II GPTF does not require any special Test EPROMs to do the STBL. The STBL can be loaded and stored in the hard drive using either the floppy drive or downloaded from an Intel Series III Development System. Once the STBLs are loaded into the hard drive, reconfiguration time (when testing different types of boards) is typically limited to exchanging the bus drawer. The STBLs for the most part use the Built-In Self Tests (BISTs) which are part of the MULTIBUS II Board Product Firmware, to test the Unit Under Test (UUT).

The MULTIBUS II GPTF adheres to MULTIBUS II architecture and follows the Intel Interconnect Interface Specification (IIS) and the Intel Initialization and Diagnostics eXecutive (IDX).

HARDWARE OVERVIEW

The MULTIBUS II GPTF is uniquely designed for ease of maintainability with three enclosures. The front enclosure is the heat chamber that houses the UUTs. Behind the heat chamber are the two computer systems; the Test Computer System and the Control Computer System. Each system has its own power supply.

The **Test Computer System**, which is MULTIBUS II based, is located immediately behind the heat chamber. It is the slave system to the Control Computer System. Its function is to perform the testing and report test status back to the Control Computer System. The Test Computer System contains three HOST MULTIBUS II boards which always reside in the GPTF.

The **Hot Box Test Chamber** has slots for testing one to four UUT's simultaneously. Both the +5V and temperature can be varied by the Control Computer (or the user) to test the boards in a worst case condition. The +5V voltage can be margined ±10%, and the temperature can be raised from room temperature to 70°C.

The **Control Computer System** is located in the rear of the GPTF and is a MULTIBUS I based system. Its function is to control and manage the Test Computer System. This system controls the AC power to the Test Computer System, has the capability to margin the DC voltages to the UUT, controls the heat chamber heater coils, reset and interrupt lines to the iSBC CSM/001 board, and controls the I/O to the CRT video display, front panel, and the secondary storage. The Control Computer contains an 8-slot MULTIBUS I backplane and five iSBC boards.

The secondary storage consists of a 3.5", 40 Mbyte winchester hard drive and a 5.25", 48 TPI floppy drive. Both iRMX86™ and PC-DOS™ format floppy diskettes can be used. The hard drive and the floppy drive are controlled by the Intel iSBC 214 Peripheral Controller board. Additional 3.5" and 5.25" Peripheral Controller board. Additional 3.5" and 5.25" peripheral bays are designed in for future Intel use.

Variable P2 Interface capability in the MULTIBUS II architecture allows for variable use of the P2 connector on iSBC boards. The iLBX™ II connector is used on some boards, like the iSBC 286/100 and the iSBC MEM/3XX boards, SCSI is used on boards like iSBC 386/258 etc. The MULTIBUS II GPTF has the bus drawer feature in the Test Computer System to support the variable P2 interface. Each bus drawer is designed for a specific P2 interface. For example, the CODE1 bus drawer, shipped with the GPTF, supports iLBX II. The bus drawers are easy to install—slide it in and tighten the two thumb screws. Only two types are shipped with the product. All the parts of the bus drawer are generic except the P2 connector itself. Each bus drawer is coded so that it can be recognized by the STBL software.

SOFTWARE OVERVIEW

The MULTIBUS II GPTF runs on iRMX 86 software specially configured for the GPTF. The operating system resides on the hard drive Control Computer System. The DIR command will assist in locating the various directories on the hard drive.

The Tester Control Program (TCP), also iRMX 86-based Operating System, resides on the hard drive and runs on the Control Computer System (iSBC

186/51 board). The TCP resembles a mini operating system. It supports a range of keyboard commands which are useful to run STBL and to troubleshoot suspect boards. A set of ten command strings can be stored in the STBL software and may be invoked at run time by the operator.

Using TCP commands, the operator can control the functions of the GPTF. TCP also responds to the front panel buttons, (START & QUIT) thus, making the GPTF automated. The CRT displays dedicated fields to indicate corresponding status of the testing such as: UUT board ID, UUT power supply status, voltage margin as percent of nominal voltage, and slot location of UUT.

The TCP operates in two modes, PRODUCTION TEST MODE (default) and TROUBLESHOOTING MODE. These modes allow the GPTF to be operated in a fully automated mode or a manually controlled mode. The PRODUCTION TEST MODE is turned off while troubleshooting with just a simple keyboard command.

The TCP works in conjunction with the firmware on the Host CPU board in the Test Computer System. The firmware is usually referred to as Host Firmware (HFW). Apart from communicating with the TCP, the HFW is an implementation of the Master Test Handler, as defined in the IDX. The Host firmware under the control of the TCP performs the testing of the UUTs.

The STBL can have tests of three different types. TYPE 1 tests run on the HOST only, TYPE 2 tests run on UUT only and TYPE 3 tests have both UUT and HOST code and can run on both. When testing more than one UUT, the TYPE 2 tests are executed in parallel by the UUTs. A given STBL can have any mixture of these three types of tests.

TESTER BLOCK DIAGRAM

Figure 1. shows a block diagram of the tester, in a level of detail sufficient to understand basic tester operation. The top of the sketch shows the MULTIBUS II system where testing takes place. On the left are the UUT slots, and on the right the host boards. Both iPSB and iLBX II busses are shown. The iLBX II backplane is physically installed in a removable bus drawer. Important communication paths shown are: a fast parallel path between host processor and control computer, and serial channels to the terminal and Series III development system. Details omitted for clarity include the heaters; most cabling; temperature sensors; +5B and heater relays.

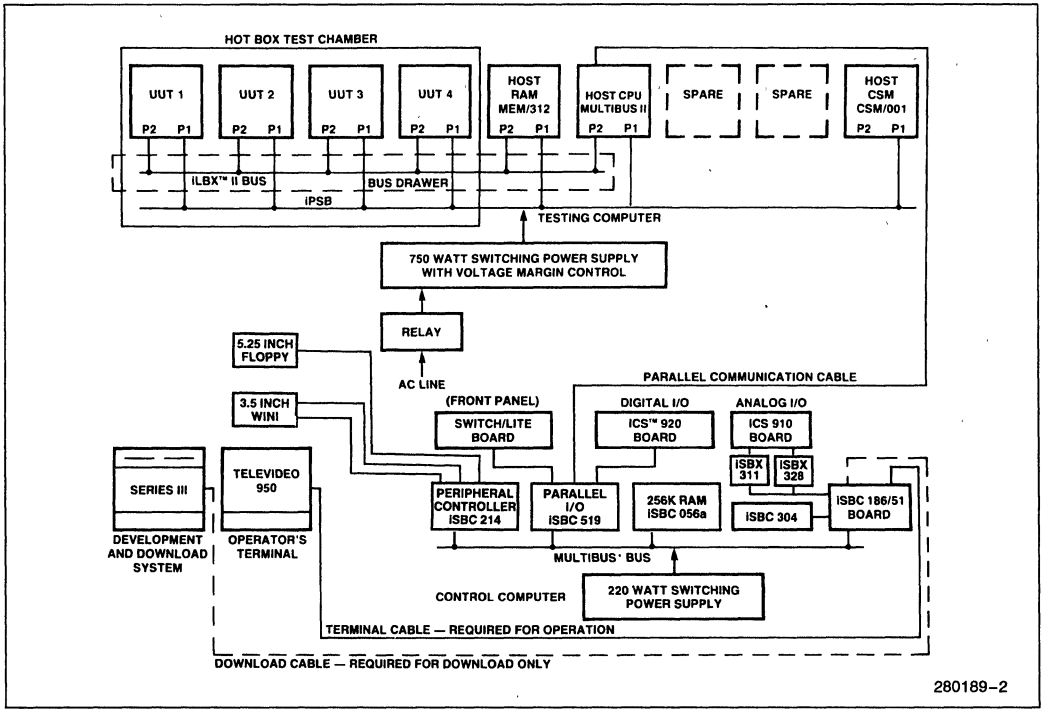


Figure 1. Tester Block Diagram

SPECIFICATIONS

Size: 25" W x 38" D x 24.5" H

Weight: 90 lbs.

Power Ratings	USA Units	International Units	Japan Units
Nominal Voltage Rating	110 volts	220 volts	100 volts
Current Rating	30 amperes	15 amperes	30 amperes
Frequency Rating	60 hertz	50 hertz	50/60 hertz

FUSE RATINGS

Power Ratings	USA/Japan	International
F1—Heater Coil 1 Fuse	10A @ 250V	5A @ 250V
F2—Heater Coil 2 Fuse	6A @ 250V	3A @ 250V
F3—MULTIBUS I Power Supply Fuse	7A @ 125V	4A @ 250V
F4—MULTIBUS II Power Supply Fuse	15A @ 250V	10A @ 250V

HEATER COIL RATINGS

Power Ratings	USA/Japan	International
Heater Coil 1	1000W 110V	1000W 220V
Heater Coil 2	660W 110V	660W 220V

Heater Coil 1 is to your right when you face the GPTF.

POWER SUPPLY RATINGS

Power Ratings	USA/Japan		International	
	*Input V	Output W	Input V	Output W
1. Control Computer System Power Supply	90–132V	220W	180–264V	220W
2. Test Computer System Power Supply	90–132V	750V	180–264V	750W

*"Input V" is the input voltage and the "Output W" is the output power.

POWER PLUGS

USA—The MULTIBUS II GPTF comes with a factory installed power plug which is a TWIST LOCK 30A, 125V PLUG.

INTERNATIONAL AND JAPAN—The MULTIBUS II GPTF is shipped WITHOUT a power plug because of the varied nature of the power outlets in other countries. CHOOSE A PLUG WHICH MEETS THE ELECTRICAL REQUIREMENTS OF THE TESTER. The GPTF is rated at 15A for INTERNATIONAL use and 30A for JAPAN.

The power outlet should be of proper rating. THIS APPLIES TO BOTH USA AND INTERNATIONAL UNITS. PLEASE USE THE FOLLOWING GUIDELINES:

INTERNATIONAL—A 15A drop with a receptacle of equivalent rating.

USA AND JAPAN—A 30A drop with a receptacle of equivalent rating.

MULTIBUS® II Architecture

5

The MULTIBUS® II Bus Structure

John Hyde
Multibus II Marketing Manager
Intel Corporation, Hillsboro, OR, 97123

Introduction

Many people equate the phrase "Multibus II" with the Parallel System Bus defined within the IEEE/ANSI 1296 specification. While this over simplification is often useful, the failure to appreciate that ***it is a contraction*** of a more embracing architecture can lead one astray when comparing the Multibus II bus structure with other buses. Comparisons between the Multibus II Parallel System Bus and other buses are often completed in isolation, without full regard to the framework in which the Multibus II architecture was defined. This chapter rebuilds this framework, describes its hierarchical structure and details how its features are **required** for multiple microprocessor designs of today.

Customer needs define the new bus structure.

Intel Corporation had had many years experience with the Multibus standard before embarking upon the requirements for a "next generation" bus structure. The first Multibus standard bus was introduced in 1974 and it was fundamentally a CPU/Memory bus. It evolved along with microprocessor technology to become a multi-master shared memory bus capable of solving most real time applications of the 1980s. The silicon trends throughout the 1980s were dramatic with DRAM densities increasing by a factor of two every three years, so projecting exactly what customers would require in the late 1980s and through to the 1990s was particularly difficult. Intel therefore set up a consortium with eighteen of its larger customers and other industry leaders who could see the potential within the single board computer industry, to define the scope and possibilities of what was to be called "Multibus II".

It was known that the rate of silicon integration would allow a complete computer system including CPU, Program Memory, Data Memory, Input/Output and bus interface to be fabricated upon a single board. With such a large transistor budget to be spent upon implementing a single board computer, where would be the optimal places to best utilize the technology? Self test and diagnostics could now be considered - with so much silicon on a board it would be prudent to use some of the transistor count to TEST the remainder of the board. Since board manufacturers are integrating more and more VLSI silicon onto their boards, the user needs some reassurance that the basic board functionality is intact before they load their value added code - the user is **demanding** on board diagnostics for these highly integrated boards. The bus interface itself, not a traditional candidate for high integration silicon circuitry, could use transistors for added sophistication IF this sophistication could make the single board computers easier to use. A trend began to develop; transistors added to improve ease of use filtered to the top of the implementation list.

With the increased silicon densities available semiconductor manufacturers turned their focus upon increased capability peripheral components. Their use on single board computers served to compound the boards complexity and the single board computer user was "rewarded" by having to wade through lengthy reference manuals and innumerable jumper options - often arriving at the final solution only by trial and error. Memory mapping options, arbitration priorities, interrupt levels and scores of other "tunable" parameters contributed to the hassle, leaving the systems engineer confused and amazed. Often the only solution was to locate a board which had already been properly configured and was operating and then copy off the jumper list.

Board manufacturers built in numerous options so that their products could be used in the broadest possible spectrum of applications. The number of options offered was not the core of the problem - **but managing them was.** Options allow interrupt routing, memory mapping, EPROM size selection, timing and other user installed components. When the jumper count exceeded 200, it no longer made sense to monopolize board real estate since an inexpensive microcontroller could be used to manage the resources more effectively.

A system bus requires standardized system-wide

configuration information to be made accessible to software, opening it to opportunities for centralized control and coordination. Ideally the end-user of these products will be completely unaware of the configuration process. They simply remove the board from it's shipping container, install the proper firmware, plug it into any free slot in the backplane, and apply power. Things work the first time around with no mess, no fuss, and no configuration errors.

The consortium therefore placed focus on the system aspects of a single board computer design. The developing model for a typical system built from these highly capable single board computers was based upon functionally partitioned subsystems interacting across a standardized communications channel. This precipitated a change in philosophy for the traditional system development from the single board computer outward to a higher level systems perspective, specified tops-down and bound together by rigid interfaces.

The consortium quickly reached consensus that no single bus could be used to satisfy all aspects of a design of this type - too many variables would have to be compromised, so a **multiple bus** structure was defined in a similar fashion to its Multibus I (IEEE 796) predecessor. Figure 1 shows the four sub-buses defined by the consortium: the iSBX® bus was retained for incremental I/O expansion, a local CPU/Memory expansion bus was proposed and two versions of a SYSTEM bus (serial and parallel) were defined. The concept of a SYSTEM bus is an important one to grasp - all open buses to date were basically CPU/Memory buses with little regard for system aspects. To have an open bus **SPECIFICALLY** designed to be a system bus was a bold step.

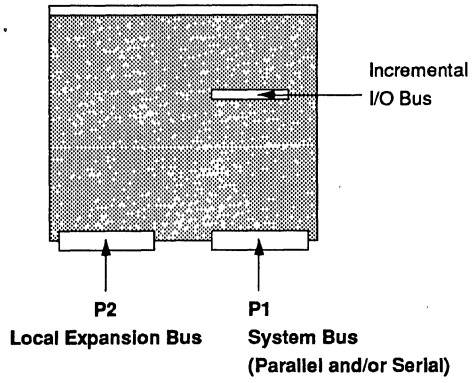


Figure 1. Since no single bus could solve the defined problem set, a multiple bus solution was proposed

Functional partitioning as a solution for non-obsolescence

Before detailing the attributes of each of the defined buses that make up the Multibus II systems architecture it is important to appreciate the model developed for the bus. Figure 2 shows a typical collection of systems connected to a local area network or LAN. This type of networked systems solution is very popular with systems builders since it boasts a large array of benefits. The solution is functionally partitioned - separate systems are used to tackle different facets of an overall problem. These systems are independent from each other and decisions made to optimize each of them for their individual task may be made in isolation with respect to the other systems in the network. This degree of freedom gives the systems architect an unquestioned edge when engineering tradeoffs are being made. The choice of hardware, options and software may be made with the sole goal of solving the small part of the overall problem currently in focus. Each system is typically tuned for its task using specially configured hardware and software and it is not uncommon to see multiple different operating systems within a single network. Systems that **MUST** respond in real time, for example, would use Intel's iRMX® Real Time operating system or their iRMK™

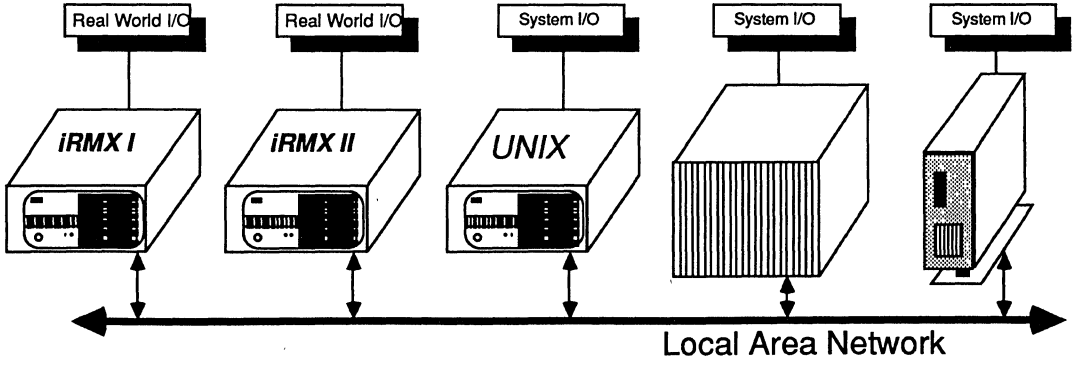


Figure 2. Multiple diverse systems can co-exist on a Local Area Network using defined protocols

Real Time kernel, while a data base manager would use an industry standard operating system such as UNIX® or DOS. The systems would probably contain a diversity of microprocessors.

Each of the systems are individually upgradeable. If something bigger/better/faster/cheaper becomes available it is easy to integrate this into a networked systems solution. New technology may be applied at strategic points on the network and no major overhaul of the complete solution is ever required. Since the overall system is continuously upgradeable it will not become obsolete and will serve for many years.

The systems are independent in their own right, capable of completing their assigned task in isolation and need not be connected to the LAN to function. The reason that they are connected to the LAN is to enable the sharing of data. The LAN defines a media type and details certain communications protocols that all the systems on the network must adhere to - in this way the diverse systems may share data in a consistent manner. Each system will require a hardware interface to the media and a software interface to the network protocols. Error recovery and retry algorithms are employed to ensure reliable communications between the individual systems on the network.

The software model for this functionally partitioned solution is "protocol based" with "data movement". In this type of model the computer population is split into server systems and consumer systems. A server system provides facilities and resources to the network such as file systems or access to a communications hierarchy. A consumer system does work using the facilities provided by the network servers. The consumer software model makes defined requests for data that a server will respond to. An inter-system communications standard was developed by the International Standards Organization and its seven-layer model is shown in Figure 3. All interfaces are rigidly defined but the implementation is not - this allows many diverse systems to interact successfully.

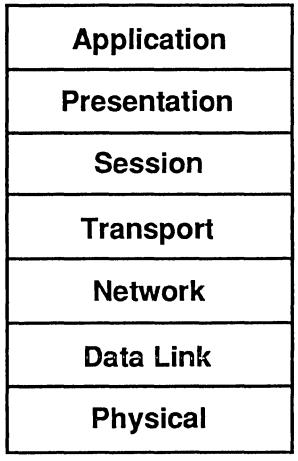


Figure 3. A seven layer inter-system communications model has been defined by the International Standards Organization

A major breakthrough.

Imagine now, keeping the same networked topology but using the advances in silicon integration to compress the systems into a single Multibus II chassis as shown in figure 4. We will use the parallel system bus as the network media. We will use protocols on the parallel system bus very similar to the protocols used over the local area network. Each of the networked systems will become a single board subsystem (or a multi-board subsystem if the circuitry exceeds the area of a single board). We have created a VERY Local Area Network or a "Backplane LAN".

A Multibus II chassis with multiple boards operating as a Backplane LAN is physically similar to a traditional system but its networked-subsystems philosophy realizes many benefits. This use of advanced Multibus II bus

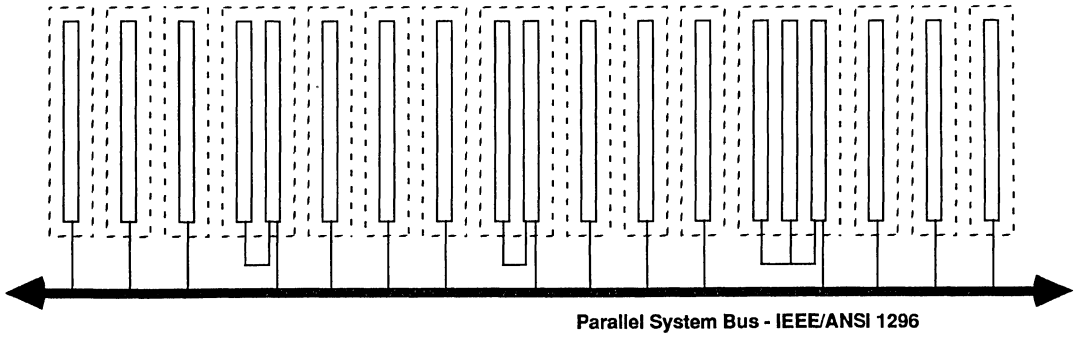


Figure 4. Higher levels of silicon integration allow a network topology to be used within a single system

technology allows the systems builder to tackle multi-CPU designs with confidence.

Partitioning a multi-CPU application into a set of networked subsystems allows us the opportunity to focus. We are able to break down a complex problem into solvable sub-problems. These sub-problems are also encapsulated therefore they can be solved independently of the other problems. Encapsulation is an important attribute of a Multibus II subsystem. Encapsulation assumes that a subsystem is intelligent and that it owns most of the resources required to complete its assigned task. These resources are private and are under the sole control of the subsystem CPU(s). In a complex subsystem these resources may be spread over multiple boards as shown in figure 5. Even though there are three physical boards in this example it is important to appreciate that, at a systems architecture level, they are treated as a single logical subsystem. The boards within a subsystem communicate using a local expansion bus defined on the P2 connector.

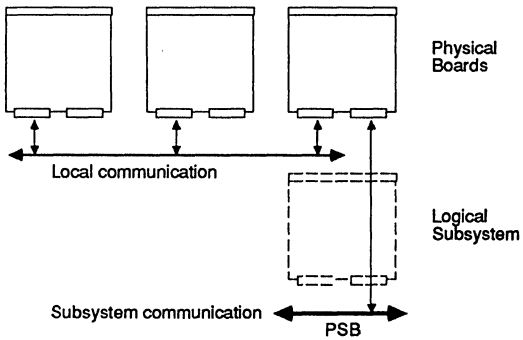


Figure 5. Multi-board subsystems interact with each other across a private P2 bus and appear as a single logical subsystem

Implementing the Sub-buses.

The realization that a single bus could not solve all of the problems is an important first step. The requirements of each sub-bus are so different that compromising their features will result in a sub-optimal system solution. The majority of this chapter will detail the system bus, but the other buses are discussed so that a context for decisions made will be evident.

The Incremental I/O bus needs to be simple. Its role is to allow the addition of a small piece of input/output onto a single board computer to customize it for a particular application. Performance is not an issue but low interfacing costs are. More extensive I/O would be added on the local expansion bus or on the system bus if an accompanying microprocessor was appropriate.

The Local CPU/Memory Expansion bus will always be dependent upon microprocessor technology. The interface between a CPU and its memory needs to be tightly coupled if we are to extract the maximum performance levels from a given microprocessor family. This bus will evolve with microprocessor technology and will typically exist for only two to four years before it has to be redesigned. If the CPU element requires more MIPs then additional identical microprocessors could be closely coupled on this local expansion bus; if these microprocessors had on-chip or local caches, as many of the higher performance offerings do, then this multiple microprocessor CPU/Memory bus must be cache coherent.

A major requirement of the SYSTEM bus is a technology independent communications media. Since this bus will remain constant throughout multiple generations of microprocessors it must be decoupled from the microprocessor technology used on the single board computer. This loosely coupled approach, whereby each single board computer subsystem is independent, will enjoy all of the benefits of the systems networked on a Local Area Network. Global system functions such as initialization, diagnostics and configuration must be added in a standardized way to this long-lived system bus.

Physical Standards

A reasonably large card size with ample power is key to making the best use of the available levels of silicon integration. While no real data has proven that edge connectors should not be used, there is a definite trend towards gas-tight pin-and-socket connectors. A double Eurocard format, IEEE/ANSI 1101 Standard, with dual 96 pin DIN connectors was chosen for the Multibus II standard. A 'U' shaped front panel, licensed from Siemens, West Germany, was chosen for its enhanced EMI and RFI qualities.

The Incremental I/O Bus

The large array of existing iSBX (IEEE 894) modules for the Multibus I family of products encouraged its adoption within the Multibus II standard. The iSBX strategy has proven itself with customers and vendors alike.

The Local Expansion Bus

The exact bus used for local expansion will vary according to the specific requirements and performance levels required in a subsystem design. As far as the IEEE/ANSI 1296 specification is concerned, this is an open option and ANY bus that is suitable may be used. Intel initiated a standard called iLBX® II which was optimized for a 12MHz 80286 microprocessor although other manufacturers have implemented this using members of the 68000 family. Siemens have implemented Multibus I on the P2 connector and called

Address Space		Address Space Size	Sequence Type	Transfer Width (bits)	Block Transfers	Number of Repling Agents
CPU/Memory Space	Memory	2**32 bytes	Read/Write	8,16,24,32	Supported with increment	One
	Input/Output	2**16 8-bit ports	Read/Write	8,16,24,32	Supported without increment	One
System Space	Message	2**8 -1 Agents 1 Broadcast	Write Only	32	Supported without increment	One or All
	Interconnect	2**9 8-bit registers each agent	Read/Write	8	Not supported	One

Figure 6. The Multibus® II System Bus has two address spaces each subdivided into two sections

it the AMS bus. Intel has also offered the PC/AT® bus as a subsystem option on a range of PC compatible products - while this subsystem bus is low performance, it is a low cost method to add dumb I/O to a Multibus II subsystem. The IEEE 896 committee is currently working upon cache coherent extensions to Futurebus; this bus, discussed in Chapter 7, would be a good candidate for a high performance local expansion bus.

The System Bus

The CPU/Memory bus defined on most buses is inadequate to support "systems-level" requirements so a **SYSTEM SPACE** was added to the definition of the Multibus II System Bus. [A good analogy here, from the software world, is the User/Supervisor Spaces common in advanced operating systems]. This system space is divided into two portions - Interconnect Space to fulfill the initialization, diagnostics and configuration requirements and Message Space to fulfill the standardized communications requirements. Figure 6 shows the four address spaces available on the Multibus II system bus - note that the traditional CPU/Memory space is retained for compatibility with existing buses and to aid migration of existing applications into the Multibus II environment. The system bus is optimized for system space operations but CPU/Memory space operations can perform well in their limited single cycle mode.

Intel's implementation of the Multibus II Parallel System Bus is contained in their VLSI bus interface device, the Message Passing Coprocessor (MPC or 82389), whose functional block diagram is shown in Figure 7. The MPC bus controller is a 70,000 transistor single chip device designed to minimize the board area required by the bus interface circuitry. By standardizing the bus interface in publicly available silicon, all users of

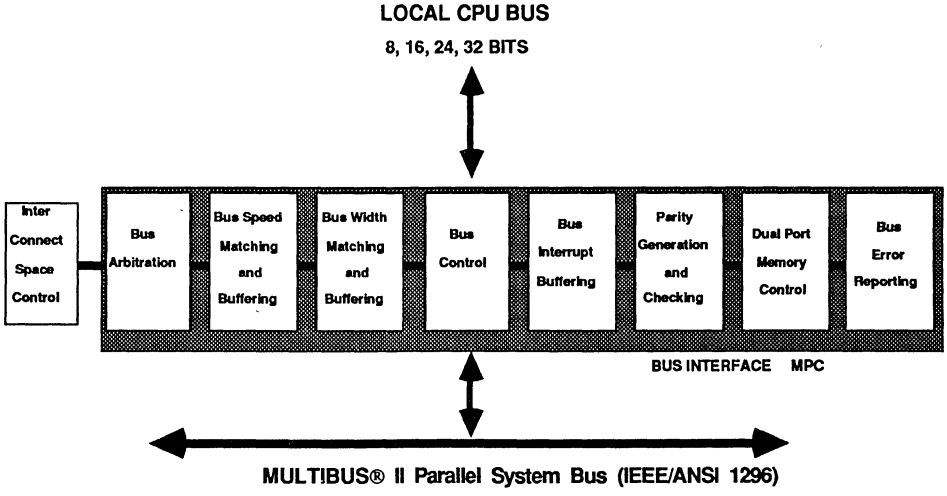


Figure 7. Intel's implementation of the IEEE 1296 specification is cast in silicon as the 82389 component.

the Multibus II standard can look forward to lowering costs and ensured compatibility. This standardization in silicon is similar to Intel's work with IEEE 754 floating point standard implemented in the 8087, 80287 and 80387 components and the IEEE 802.3 Ethernet® standard implemented in the 82586 and 82588 components.

The 70,000 transistors which make up the MPC bus controller implement a variety of functions as shown in Figure 8. As seen from Figure 8 most of the MPC bus controller deals with message space, either interrupt messages or data transfer messages, or with interconnect space.

Traditional Bus Functions	
Bus Control	4,000
Bus Arbitration	1,000
Dual Port Memory Control	2,000
Off-Board References	1,000
Interrupts	20,000
Advanced Bus Functions	
Parity Generation/Detection	1,000
Interconnect Space	6,000
Built-In-Self-Test	1,000
Message Passing	34,000
Total	70,000

Figure 8. The majority of the 70,000 transistors within the MPC support the System Space functions of the Parallel System Bus

The MPC bus controller contains almost all of the logic needed to interface any microprocessor to the Parallel System Bus - indeed all of today's popular 32-bit microprocessors are available on Multibus II products. One of the few required external components are the high current bus drivers as shown in Figure 9. Optional external logic to support dual-port memory selection and off-board memory and I/O references may be included if traditional bus functionality is required. All of Intel's Multibus II boards also includes a microcontroller (8751) to implement interconnect space but some members of the Multibus Manufacturers Group have chosen to implement this using the host microprocessor or a simple state machine.

The alternate system bus, the Serial System Bus or SSB, is currently defined but is not implemented in silicon. The goal of this bus was to reduce the cost of coupling multiple boards together and it was specified as a 2Mb/sec serial link. All software interfaces to an SSB chip would be identical to that of the MPC parallel bus controller so NO SOFTWARE CHANGES would be necessary to use the serial system bus. Performance would be much less using this serial system bus but, for many designs, this would be acceptable. Other designs, however, would benefit from a 200Mb/sec link and Intel has joined others on the IEEE 1394 serial bus

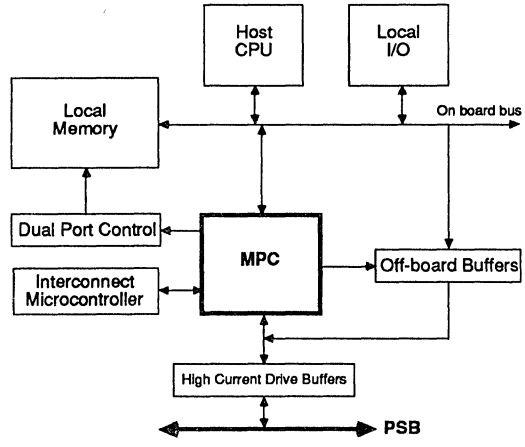


Figure 9. The MPC Integrates all of the System Bus functions into a single VLSI component

standardization committee to deliver this. This group of multiple vendors is driving for a standard which will allow ALL systems to interoperate. Implementation of the SSB interface chip is on hold pending resolution and recommendation from this IEEE committee.

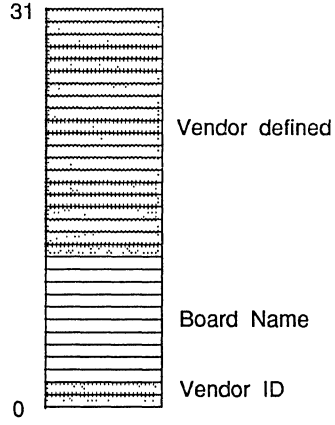
Interconnect Space

Interconnect address space is a fundamental part of the IEEE/ANSI 1296 specification and it addresses three major customer requirements: Board identification, initialization, configuration and diagnostics. Interconnect space is implemented as an ordered set of eight-bit registers on long word (32 bit) boundaries - in this way little endian microprocessors such as the 8086 family and big endian microprocessors such as the 68000 family access the information in an identical manner. One objective of interconnect address space is to allow higher level software to gain information concerning the environment in which it operates, independent of who manufactured the board, the functions it contains, and the card slot it is in. To accomplish this goal, a comprehensive Interconnect Interface Specification which builds upon the concepts introduced within the IEEE/ANSI 1296 specification has been published by Intel Corporation and is available from the Multibus Manufacturers Group.

Board identification registers are read-only locations containing information on the board type, its manufacturer, what components are installed, and other board specific functions. Configuration registers are read/write registers which allow the system software to set and change the configuration of many hardware options. In most cases hard wired jumper options can now be eliminated in favor of software control. Diagnostic registers are used for the starting, stopping, and status reporting of self-contained diagnostic routines supplied

with each board. These diagnostics are commonly known as Built-in Self Tests (BISTs).

Interconnect space is based on the fundamental principle that you can locate boards within a backplane by their physical slot position. This concept, known as geographic addressing, is a very useful tool during system-wide initialization. Each board in the system contains firmware which conforms to a standardized header format as shown in Figure 10. At boot time, the system software will scan the backplane to locate its resources before loading device drivers. This approach eliminates the need for reconfiguring the software every time a new board is introduced to the backplane. It also solves the problem of how to configure multiple controller and processor boards in large multiprocessing systems. Slot independence is achieved by having all boards in the system carry their own initialization and diagnostic functions on-board in firmware. Operating systems can generate a map of where resources are located during initialization time, and then use this list as the basis of message passing addresses.



NOTE: Location 32 must return 0FFH

Figure 10. All IEEE/ANSI 1296 compatible boards contain an Interconnect Space Header Record

In addition to the header record, a board manufacturer may also supply additional function records which make other features of the board accessible to the user through interconnect space. An example is shown in figure 11. Function records begin with a byte specifying the record type, followed by the number of bytes which the function record contains. The data contained in a function record is organized by the manufacturer according to published specifications which accompany the board. Many types of function records have already been defined. Some examples include memory configuration, parity control, serial I/O, and

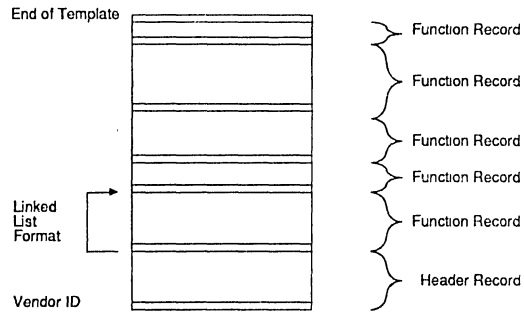


Figure 11. Extended records within Interconnect Space give System Software knowledge of the hardware

other commonly used functions. If there is no existing record type which adequately describes a given function, new record types can be defined, up to a maximum of 1020 different record types. System software will search for a particular record by starting with register number 32 (end of the header record; start of the first function record), and scanning the record type field, then counting bytes to the next function record until either the correct record is found or an "End of Template" record (hex value 0FFh) is encountered.

Diagnostic Philosophy within Interconnect Space

Intel has taken the usefulness and standardization of interconnect space one step further by embracing a standard diagnostic philosophy. Each intelligent board should have the capability to test itself and report error status in interconnect space if problems exist. There are two occasions when diagnostic testing can be invoked. A subset of the complete on-board diagnostics will be run during power-on initialization, and more extensive testing can be invoked from an operators console. Following power-on, most boards will go through a series of initialization checks, where the basic functioning of the MPC bus controller and microcontroller is verified. These checks are followed by a power-on test suite which is controlled automatically by each local microprocessor. If a hardware failure is detected at this point, a yellow LED on the front panel will illuminate so that the failing module can easily be identified and replaced by an operator, additionally test results are posted in interconnect space to be read across the backplane. Note that a CPU board when scanning interconnect space can now discover the operational status of boards in the backplane as well as their identity.

If further testing is desired, extended diagnostics can be invoked by placing a diagnostic request in the BIST registers of interconnect space. Usually one board will operate as a Master Test Handler, and will request services from other boards in the system which function

as Slaves while under test. A menu of available tests is accessible via interconnect space. This test philosophy can be applied on-site by the end-user or service representative, or remotely executed via modem from a regional repair center. In most cases, downtime can be minimized by sending out a replacement board, thus avoiding an expensive repair call.

The firmware content of Multibus II boards is much greater than on previous industry standard buses. In addition to the 8751 microcontroller, there are likely to be EPROMs on board which contain the extended diagnostics, test handlers, reset initialization sequencing, debug monitors, and numerous other functions. The location of diagnostic firmware on a board will depend on the complexity of the code and the speed at which it runs. For simple replier agents, it may be that the on-board EPROM of the 8751 microcontroller contains enough program store for rudimentary diagnostic functions as well as the interconnect core firmware. In contrast, most requestor/replier boards (those capable of becoming bus masters) are more complex, and most diagnostic code is run by the microprocessor from on-board EPROM. In this case, the 8751 serves primarily as the communications interface for diagnostics.

Interconnect Space - The Manufacturers Perspective

From the perspective of a board designer, interconnect is a mixed blessing. The board manufacturer is certain to enjoy the benefits of reduced support costs, easier fault isolation in field repairs, and enhanced customer satisfaction, but these advantages do not come for free. One would anticipate longer development times, increased parts count on-board, and configuration in firmware to increase the amount of effort it takes to prepare a Multibus II board for market. Indeed this is so. In order to minimize this development time Intel has produced an Applications Note which details the steps and discusses the options available for a full featured interconnect space implementation. The core microcontroller code is also provided on a DOS diskette and is designed to be user extensible. It is now straightforward to add these advanced capabilities to any Multibus II board design.

The Message Passing Mechanism

While the previously described features make more reliable systems easier to build using the Multibus II standard, it is the innovative message passing scheme that gives the parallel system bus its high performance in a multiple microprocessor application. The underlying theory behind message passing is simple - it decouples activities between the host microprocessor's local bus and the system bus. This decoupled-bus approach provides two major advantages. First, it allows increasing parallelism of operation - resources that would otherwise be held in traditional wait states while arbitration occurs are freed, and second, one bus bandwidth does not limit the transfer rate of another. The local microprocessor bus and the system bus can perform full speed synchronous transfers independently and concurrently. The decoupling is achieved within the MPC bus controller using high speed FIFO circuitry as shown in Figure 12.

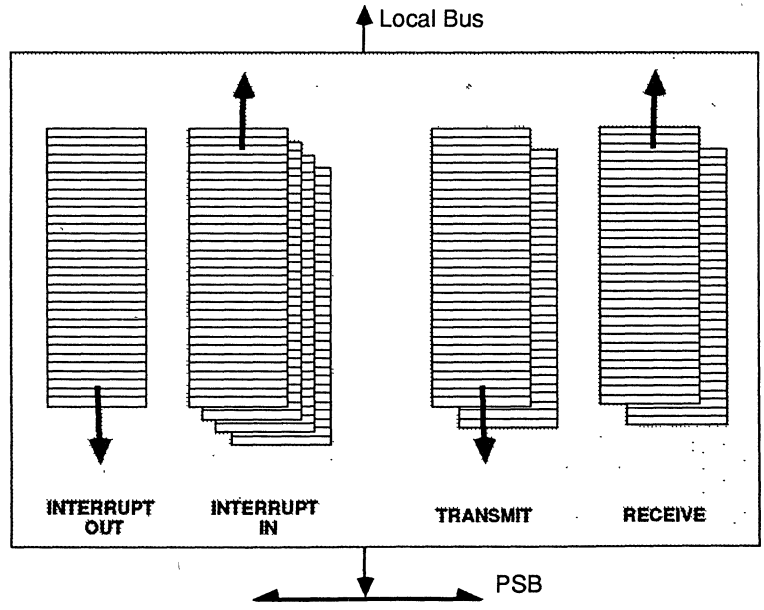


Figure 12. Decoupling of the local bus from the system bus is achieved with nine very high speed FIFOs

Nine 32 byte FIFO's are integrated into the MPC bus controller. Five of them are used for interrupt messages (one transmit and four receive) and four are used for the transfer of data blocks (two transmit and two receive). To understand the impact of message passing, let us consider a simple example of transferring a 1K block of data from CPU A to CPU B as shown in Figure 13. We will first use a shared memory method and then a message passing method.

To use a concrete example lets assume that A is a 186 based board and can transfer data at 1 MB/sec and B is a 386™ based board that can receive data at 10 MB/sec. We will ignore DMA controller setup. DMA controller A will put a destination address onto the system bus and the address decode logic on board B will respond. We wait for the address to propagate through the dual- port controller on board B and then wait for the access time of the memory on board B. Data is transferred and once accepted by board B a ready signal will be generated and DMA controller A will move on and generate the next address. This address-wait-data cycle repeats until the full 1KB of data is transferred. The overall speed of the transfer will be 1 MB/sec (the slower of the two boards) so it will take 1 msec to transfer the complete 1K buffer. If the system bus was required by an alternate CPU then the current data transfer would be delayed or the alternate CPU would have to wait.

Now lets consider the message passing case. This time we have to set up both DMA controllers. CPU A could probably transfer data faster than 1 MB/sec into

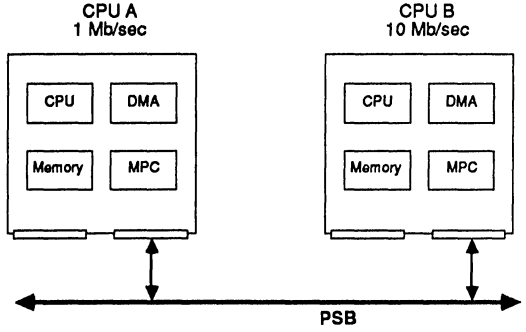


Figure 13. Let us move a 1KByte block of data from Board A to Board B

a local I/O port (the MPC bus controller) but we will ignore this potential performance improvement. The speed of this transfer will still be 1 MB/sec, the speed of the slower board, and the total transfer time will still be 1 msec. What did we gain then for the overhead of setting up two DMA controllers?

Let us look in detail at what is happening inside the MPC bus controller. Figure 14 shows a fragment of each board with different areas of each MPC bus controller highlighted. Data is being DMA'ed into MPC-A at 1 MB/sec and flows into one of the transmit FIFO-pairs. Once 32 bytes have been received the MPC automatically switches to the alternate transmit FIFO and starts to fill that. The full transmit FIFO empties

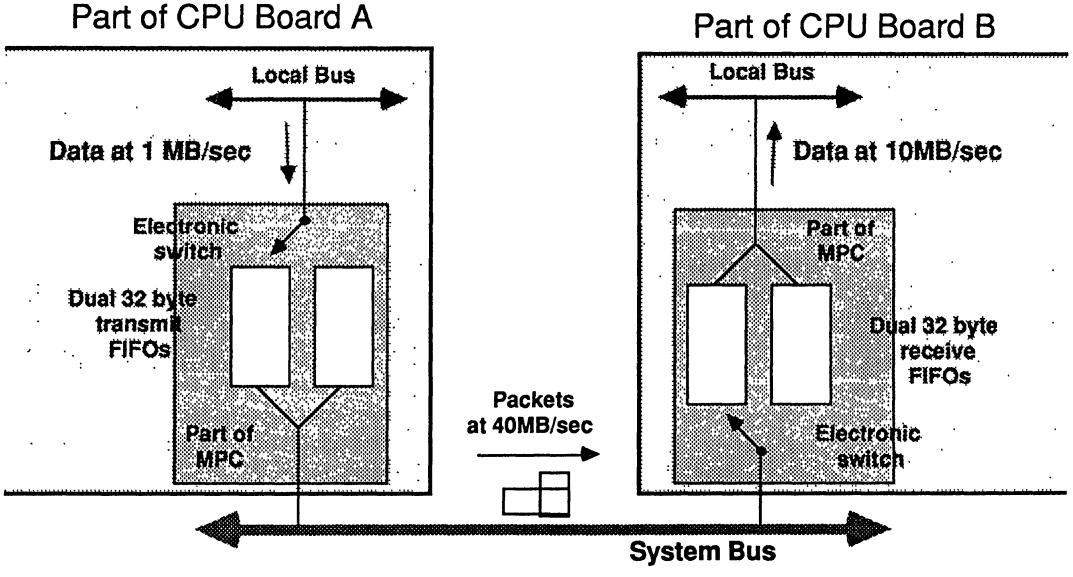


Figure 14. Looking closely at the message based data transfer mechanism

across the system bus into a receive FIFO in MPC-B. This transfer of a 32 byte packet occurs at the full bus bandwidth of 40 MB/sec. A data packet has a two clock cycle header which describes the source, destination and type of this packet which reduces the effective data transfer rate to 32 MB/sec. The packet therefore takes 1 microsecond to pass between the two MPC bus controllers. Bus arbitration is done in parallel with the packet transfer so this does not add to the transit time. Once the packet is inside of MPC-B then DMA-B empties its receive FIFO at 10 MB/sec.

The transmit FIFO-pair of MPC-A alternate between filling from local memory and emptying into MPC-B until the full 1K of data has been transferred. No programming, save the initial setup, is required. If we look at the system bus activity we see that 1 microsecond packets are being transferred at 32 microsecond intervals - the bus is only busy for 3% of the total data transfer. We have gained 97% bus availability. Compare this with the 0% bus availability in the shared memory case. Message passing frees up system bus bandwidth to enable many other single board computer pairs to interchange data at no loss in performance. In a multiple microprocessor application the most precious resource will be system bus bandwidth and the Multibus II message passing scheme gives you more, much more.

We gain a lot more than system bus bandwidth using message passing. Note that CPU A transferred data from its local memory into a local I/O port (the MPC bus controller). CPU A did not have to understand the memory layout or restrictions of memory on CPU B - this also allowed CPU B to do its own memory management and buffer allocations. Similarly CPU B has no concern over how CPU A does its memory management. We have isolated the data away from known memory locations and do not have to deal with semaphore flags or similar mechanisms. This simplifying step makes intercommunicating with multiple microprocessors as straight forward as communication with a single microprocessor. This isolation of concerns regarding the local environments of each board, through the use of a standardized data transfer mechanism, is especially important in the general case where each board is running a different operating system (probably on a different microprocessor). A real time operating system can now simply exchange data with, say, UNIX using this standardized message passing mechanism. Message passing also standardizes inter-CPU signalling since interrupts are special TYPEs of packet (more later).

This short explanation has over-simplified the transfer - some setup is required so that the sending MPC bus controller knows the message address of the receiving MPC bus controller etc. This overhead is more than compensated for by the ignored increase in local transfer data rates. I also simplified the issue by having a receiving board much faster than the transmitting board (10 MB/sec vs. 1 MB/sec) - if I had transferred data in the opposite direction (from B to A) then MPC-A would have rejected some packets because its receive FIFOs would be full and caused MPC B to retry some data transfers. No data would ever be lost but bus activity would have increased. The MPC bus controller uses a logarithmic backoff algorithm on retries so the bus activity increase would not be excessive. Alternately MPC B could be preprogrammed to use a lower packet duty cycle if it had known that MPC A would always be slower.

Having the underlying architectural support to permit multi-CPU solutions is, of course, only the first step. To build systems we need software. Intel, working with other vendors, has defined a Transport Protocol specification above the MPC bus controller which provides services such as large block transfers and acknowledged transactions. Data fragmentation at the sender or receiver is detailed so that large data buffers are neither assumed or required. The implementation is efficient across all CPU architectures; indeed, Intel has supplied implementations on the iRMX Real Time Operating System, the iRMK Real Time Kernel and the UNIX System V.386 operating systems; these are compatible with offerings from Digital Research (FLEXOS®), Microbar (VRTX®) and Tadpole Technologies (UNIX68K).

Message Space Details

The MPC bus controller introduces a hardware recognized data type called a packet as shown in Figure 15. The MPC contains FIFO circuitry such that these packets may be moved very efficiently between MPCs - data is moved on subsequent clock edges of the 10MHz synchronous bus; this defines the maximum bus occupancy of a packet to be one microsecond. Each MPC bus controller has an address in message space and these are used in the message header (source and destination fields).

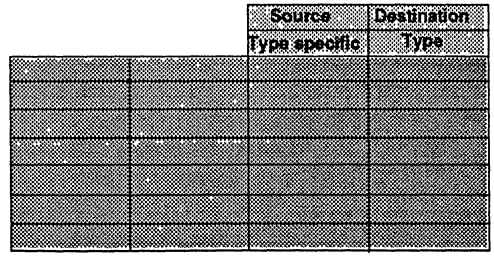


Figure 15. The MPC bus controller introduces a hardware-recognised data type called a packet

Seven different packet types are currently defined and are summarized in Figure 16. These divide into two catagories; unsolicited, or interrupt packets and solicited, or data transfer packets. The data fields within a packet are user defined and the length may vary from zero to a maximum of 32 bytes (28 for an unsolicited packet) in four-byte increments. Note that a packet with no data bytes will only consume 2 clocks or 200nsec of system bus time.

Unsolicited Packets.

Unsolicited packets, as the name implies, are always a surprise to the MPC bus controller. Their arrival is unpredictable so each MPC has four FIFOs in which it can queue unsolicited packets. These packets are equivalent to the interrupts of a conventional bus with the added feature of having up to 28 data bytes provided with the signal. There are five different unsolicited packets: two are used for interrupts and three are used to set up solicited data transfers. A general interrupt request may be sent between any pair of single board computers and a broadcast interrupt may be sent to all boards in a system. The three special interrupts, Buffer Request/Grant/Reject are used to initiate large data transfers (solicited messages) between pairs of single board computers.

Solicited Packets.

Solicited packets are never a surprise since their arrival is negotiated - the receiving MPC bus controller knows what to do with them. These packets are used for the transfer of data from one board to another and the transfer is set up using unsolicited packets. To summarize the operation of solicited packets the MPCs cooperate in the moving of blocks of data between boards, they break the data into 32 byte packets, send them across the bus, and reassemble the data transparently to the sending/receiving boards. All operations such as packetization, bus arbitration, error detecting and recovery is handled by the MPC bus controllers - this is done transparently to the local microprocessors. Key to system performance is the packetization of data on the system bus which limits the maximum bus occupancy to one microsecond.

Bus arbitration guarantees low bus latency.

The Multibus II system bus uses a distributed arbitration scheme as shown in Figure 17. Each board that requires access to the system bus contains the circuitry of figure 17 (note that this is contained within the MPC bus controller) and no active components are required on the backplane. The MPC bus controller uses a software assigned identifier to request the bus and the arbitration circuitry will indicate that the MPC is preparing for a bus cycle. The MPC supports two arbitration algorithms, fairness and high priority.

The fairness mode is used for data transfers and

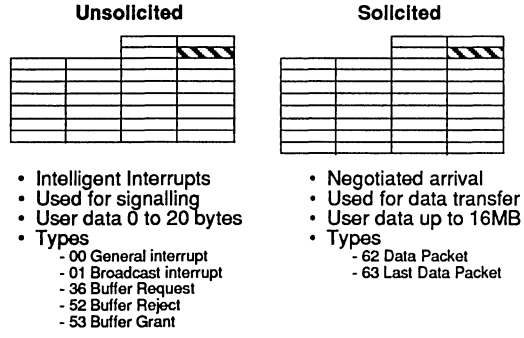


Figure 16. There are seven types of packet subdivided into two catagories

is "polite". If the bus is being used, the MPC will wait before requesting use of the bus; once the bus is not busy the MPC will request the bus and will wait for it to be granted; once the MPC uses the bus it will not request it again until all other requesters have used the bus. "Parking" on the bus is permitted - if no other board has requested the bus since the last time that this board accessed the bus (remember that the bus is continually monitored) then it may access the bus directly without executing an arbitration cycle. These algorithms assure that a single board cannot monopolize the bus and keep others from using it. Remember that each MPC will only use the bus for a maximum of one microsecond and since the arbitration is being resolved in parallel there are no wasted clocks as bus ownership is transferred; all transfers operate back-to-back.

The System Bus has Deterministic Interrupt Latency

The high priority mode is used for interrupts and is "impatient". The MPC bus controller, when in this mode, will "barge in" on an arbitration cycle and be guaranteed the next access to the bus. The MPC bus controller can set up interrupt packets specifically to operate in high-

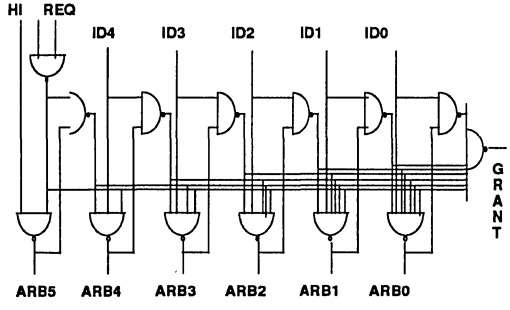


Figure 17. The Multibus® II Parallel System Bus uses a distributed arbitration scheme

priority mode so these will only have a one microsecond latency to access the bus. In the rare instance that two MPCs try to initiate an interrupt packet within the same one microsecond window the highest priority board will be granted the bus and the other board will have to wait a maximum of two microseconds for its bus access.

Interrupt packets and data transfer packets interleave on the bus (actually interleave within a single MPC too) with preference always going to the interrupt packet. A single MPC bus controller will operate with interrupt packets in high priority mode and data transfer packets in fairness mode; this will ensure that interrupt packets have a deterministic bus transit time of 1µsec (2µsec sometimes and 20µsec conceivable worse case when all 21 MPCs try to initiate an interrupt packet within the same one microsecond window, a very very rare occurrence).

System Bus Characteristics

Figure 18 shows the pinout of the P1 connector . The signals can be classified into five groups; 1. Central control, 2. Address/data, 3. System control, 4. Arbitration and 5. Power.

Pin	Row A	Row B	Row C
1			
2	0volts	PROT*	0 volts
3	+5 volts	DCLOW*	+5 volts
4	+12 volts	+5 volts (Batt)	+12 volts
5	0 volts	SDA	BCLK*
6	TIMOUT*	SDB	0 volts
7	LACHn*	0 volts	CCLK*
8	AD0*	AD1*	0 volts
9	AD2*	0 volts	AD3*
10	AD4*	AD5*	AD6*
11	AD7*	+5 volts	PAR0*
12	AD8*	AD9*	AD10*
13	AD11*	+5 volts	AD12*
14	AD13*	AD14*	AD15*
15	PAR1*	0 volts	AD16*
16	AD17*	AD18*	AD19*
17	AD20*	0 volts	AD21*
18	AD22*	AD23*	PAR2*
19	AD24*	0 volts	AD25*
20	AD26*	AD27*	AD28*
21	AD29*	0 volts	AD30*
22	AD31*	Reserved	PAR3*
23	+5 volts	+5 volts	Reserved
24	BREQ*	RST*	BUSERR*
25	ARB5*	+5 volts	ARB4*
26	ARB3*	RSTNC*	ARB2*
27	ARB1*	0 volts	ARB0*
28	SC9*	SC8*	SC7*
29	SC6*	0 volts	SC5*
30	SC4*	SC3*	SC2*
31	-12 volts	+5 volts (Batt)	-12 volts
32	+5 volts	SC1*	+5 volts
	0 volts	SC0*	0 volts

Figure 18. The complete Parallel System Bus is implemented on a single 96pin DIN connector.

Central Control

The parallel system bus is a synchronous design and great care is taken, especially within the backplane electrical specifications, to maintain a crisp 10MHz system clock. All other signals are referenced to the system clock for setup and hold times. The IEEE/ANSI 1296 specification details precisely what happens upon each of the synchronous clock edges so there is no ambiguity. The specification also details numerous state machines that track bus activity and are implemented to guarantee compatibility.

A central services module (or CSM) in slot 0 generates all of the central control signals. This CSM may be implemented on a CPU board, a dedicated board or on the backplane of a cardcage. The CSM drives reset (RST*) to initialize a system; a combination of DCLOW* and PROT* are used to distinguish between cold start, warm start and power failure recovery. Two system clocks are generated, BCLK* at 10MHz and CCLK* at 20MHz. RSTNC* and LACHn* are used for advanced facilities within the bus and their description is deferred.

SDA and SDB are reserved for a serial system bus (currently being investigated by the IEEE) and there are two pins reserved for future use.

Address/data

The Multibus II parallel system bus is a full 32 bits (AD0..31*) with byte parity (PAR0..3*). The system control lines will define when address information or data is contained upon these multiplexed lines. Note that all transfers are parity checked and, in the case of message packets, the MPC bus controller will retry an operation that failed due to a parity error. If, after sixteen tries, the error is not recoverable, the MPC bus controller will interrupt its host microprocessor to ask for assistance.

System Control

Ten lines (SC0..9*) are used for system control and their functions are multiplexed too. SC0* defines the phase of the current bus cycle (request or reply phases) which then defines how SC1..7 should be interpreted. SC8 provides even parity over SC0..3 and SC9 provides even parity over SC4..7. Figure 19 shows the decoding of the Status/Control signals throughout a typical bus cycle.

Signal	Function during Request Phase	Function during the Reply Phase
SC0*	Request Phase	Reply Phase
SC1*	Lock	
SC2*	Data Width	End-of-transfer
SC3*	Data Width	Bus Owner Ready
SC4*	Address Space	Replier Ready
SC5*	Address Space	Agent Status
SC6*	Read or Write	Agent Status
SC7*	Reserved	Agent Status
SC8*	Even parity on SC<7..4>*	
SC9*	Even parity on SC<3..0>*	

Figure 19. The Status/Control lines are encoded to preserve lines on the system bus

Arbitration

All boards request use of the bus through a common bus request line, BREQ*. A distributed arbitration scheme is defined which grants the bus to the numerically highest requesting board as identified on lines ARB0..5. Two arbitration algorithms are supported: fairness, which gives each board an even portion of the available bus bandwidth, and priority, which permits a high priority request (such as an interrupt) to be guaranteed the next access to the system bus.

Power

There are ample power and ground lines defined and these are spread over the length of the P1 connector to minimize ground shift and other problems.

Typical Bus Cycle

The parallel system bus is particularly easy to interface to. This section will cover the sequencing of a typical REPLIER interface as an illustration of the bus timing. The IEEE/ANSI 1296 specification details numerous state machines that track bus activity and are implemented to guarantee compatibility. An I/O replier need only implement a single "Replying Agent" state machine. This is shown in Figure 3.5-5 in the IEEE/ANSI 1296 standard and repeated here in Figure 20 for reference. Remember that an application CPU (a REQUESTOR) will start the cycle that the REPLIER will respond to.

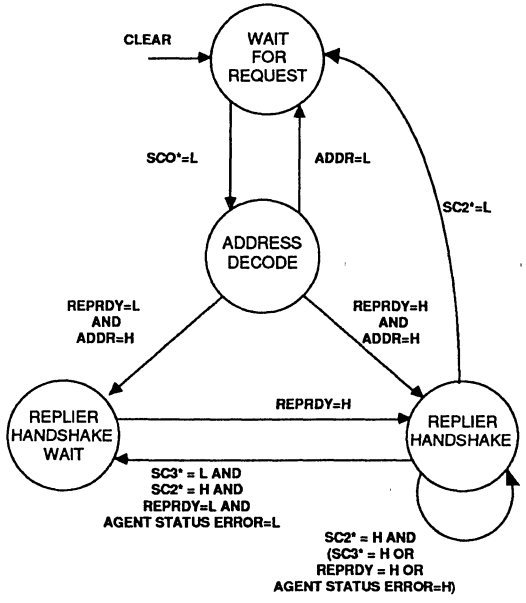


Figure 20. The IEEE/ANSI 1296 Specification details numerous state machines. A replier is shown here.

In order to progress quickly through this discussion, an assumption that the requestor always issues valid requests will be made. Error handling for invalid requests will be added later. Figure 21 summarizes the design task. The logic required to map the multiple signals and protocols from the Multibus II parallel system bus into the simple read strobe, write strobe and chip select of an I/O device must be designed. In this example features will be kept at a design minimum but all essential circuitry will be discussed in detail.

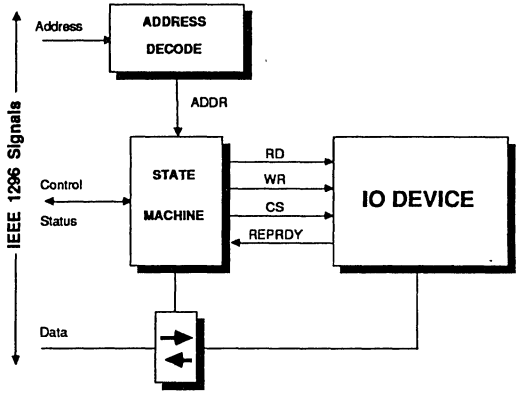


Figure 21. The design of a REPLIER is fundamentaly a bus monitor.

The replying agent state machine is fundamentally a bus monitor. State transitions in figure 20 occur at the falling edge of bus clock. The state machine remains in the wait-for-request state until it detects the start of a requestor cycle on the system bus (SC0* LOW) then it moves into an address decode state. If this requestor cycle is not ours (local decode signal ADDR is LOW) then return to the wait-for-request state. If the requestor cycle is detected as ours (ADDR is HIGH) then transition to a new state controlled by a local ready signal (REPRDY). If not ready (REPRDY is LOW) then wait until ready. Once ready then wait until the requestor is ready (SC3* is LOW) and provide/consume valid data. Check to see if this is a multi-byte transfer (SC2* is HIGH) and if it is not return to the wait-for-request state.

If a multi-byte transfer is detected then decide to accept or to ignore the data in the remainder of the cycle. If the additional data cannot be handled then signal an agent status error (Continuation error) and wait for the requestor to terminate the cycle. If a multi-byte transfer can be supported then oscillate between the replier wait state and the replier handshake state where data is strobed. Eventually the requestor will signal the last data element (SC2 set LOW) and return to the wait-for-request state.

At the start of each requestor cycle that status lines (SC1* through SC6*) detail the type of cycle; SC1* signals a locked transfer, SC2* and SC3* encode the data width, SC4* and SC5* encode the address space and SC6* signals a READ or WRITE cycle. A replier must latch these status lines with the address bus and use the information to control its subsequent cycle. A complete list of the Status/Control decoding is shown in Figure 19.

An I/O replier has certain responsibilities that must be adhered to. A requestor expects an I/O replier to generate status information and to signal when ready so that the requestor may proceed with the cycle. The cycle will only terminate once both requestor and replier have signalled that they are ready (the IEEE/ANSI 1296 includes a time out feature which prevents the bus from hanging if both ready signals are not generated). A replier drives SC4* LOW to indicate READY and status information is driven on lines SC5* through SC7*; SC8* must also be driven and identifies parity across lines SC4* through SC7*. If a replier is supplying data to a requestor then correct data parity must also be driven onto the system bus.

Summary

The Multibus II Parallel System Bus was **DESIGNED** to implement all of the "systems features" of a single board computer based system. The bus does have some CPU/Memory attributes but these were only included for compatibility and to aid migration into the Multibus II environment - comparing these CPU/Memory features in isolation with those of other buses is a complete disservice to the Multibus II architecture and misses the complete design goals and motivation set forth for this standard.

The silicon revolution forced the design of the Multibus II Parallel System Bus - technology was advancing faster than our abilities to use it so we had to find new implementation strategies to benefit from these advances. Functional partitioning was chosen as the vehicle to embrace the technology; by partitioning the problems into smaller and smaller sub-problems we reach a point where the sub-problems are implementable. The Multibus II consortium chose this path and executed with precision; transistors were applied at strategic points to simplify implementations and encourage ease-of-use. The Multibus II architecture is completely defined, documented and available.

intel®

APPLICATION NOTE

AP-422

October 12, 1987

Designing a Central Services Module for MULTIBUS®II

JORY RADKE
MODULES DEVELOPMENT ENGINEER

Order Number: 280634-001

PURPOSE

This paper describes and presents methods for implementing the functions provided by the Central Services Module, as defined in the IEEE 1296 specification, and is intended to assist the sytem designer in understanding and effecting these functions. Function options and other design considerations are discussed. It is assumed the reader is familiar with the terms and definitions used in the IEEE 1296 specification and with basic logic design prinicples.

1.0 INTRODUCTION

The IEEE 1296 specification, based on the Intel® MULTIBUS® II bus architecture, defines certain general system-wide functions to be provided by a Central Services Module (CSM). These system-wide functions include power-on and power-fail reset sequences, clock generation, bus timeout detection and signal generation, and the assignment of cardslot and arbitration identification (ID) to each board (agent). The communication path between the CSM and the other agents in the system environment is via the Parallel System Bus (PSB) interface.

The implementation of the defined CSM ensures uniformity in providing a single source for those system-wide functions required in an open-bus architecture, such as that established in the IEEE 1296 specification. Centralizing system-wide functions reduces system cost and frees board area for other functions since only one board in the system need contain the CSM logic. The IEEE 1296 specification stipulates that only the agent in cardslot 0 contain the active CSM functions although other system agents may contain CSM functions.

2.0 CSM FUNCTIONS AND PSB SIGNALS

The following paragraphs identify and briefly describe the system-level services and functions supplied by the CSM and the PSB signals generated, monitored or used to implement these services. An asterisk following the signal name indicates that the particular signal or group of signals are active when at their electrical low.

2.1 CSM Functions

The IEEE 1296 specification defines the minimum required functions of a CSM as:

- Generation of system clock signals
- Generation of reset sequences for both cold and warm start and power failure indication
- Cardslot and arbitration ID initialization
- Timeout signal generation for PSB data transfer cycles.

Depending on system requirements, the CSM may additionally provide:

- Power-fail recovery reset
- Bus ownership timeout.

Other system-wide resources, such as a time of day clock or interface to another bus system, may be conveniently implemented with the CSM on the same PSB agent. We shall see that the CSM functions require very little board area to implement.

2.2 PSB Signals Used by the CSM

The CSM utilizes signals from each of the five signal groups defined in the IEEE 1296 specification. These signals are identified and their use by the CSM is described briefly in table 2-1.

3.0 FUNCTIONAL OVERVIEW

The following sections discuss how to add the CSM functions to a PSB agent. The agent could contain only the CSM and interconnect relier modules or additional functional modules as well. The design example provided in section 4, (excepting the PSB buffers), requires less than six percent of the area on a standard MULTIBUS II board. The CSM module cannot be added to agents which employ Intel's Message Passing Coprocessor, due to the current and capacitive loading requirements of the PSB signals in table 2-1 which the MPC drives directly.

This paragraph provides a functional overview of the design and discusses signal requirements. A detailed design example is illustrated and discussed in paragraph 4. Additional design considerations are described in paragraph 5.

Functionally partitioning the CSM functions results in the block diagram shown in figure 3-1. The signal terminations identified on the righthand side of the diagram are the actual PSB pin assignments identified in the IEEE 1296 specification.

3.1 Clock Generator (CLKGEN) Function

Listed in table 3-1 and depicted in figure 3-2 are the timing relationships between the **BCLK*** and **CCLK*** signals as specified in the IEEE 1296 specification. The circuits used to develop and supply the **BCLK*** and **CCLK*** signals must

Table 2-1. PSB Signals Used by the CSM

Signal		CSM Function
Name	Group	
BREQ*	Arbitration Cycle	Bus Request. All agents that require access to the PSB assert the **BREQ*** signal. The CSM monitors this signal as part of its bus timeout function.
ARB<5 . . 0>*	Arbitration Cycle	Arbitration lines. The CSM uses these lines during a reset sequence to assign a cardslot ID and an arbitration ID to each agent in the system.
AD<20 . . 1>*	Address/Data Bus	Address/Data lines. See figure 2-1. Each Address/Data line is connected to the **LACHn*** pin of a cardslot. The **LACHn*** signal is used to latch the cardslot and arbitration IDs to each agent (except cardslot 00) during a reset sequence.
SC<4 . . 2, 0>*	System Control	System Control lines. The CSM monitors these control signals between agents to sense bus timeout during data transfer cycles.
TIMOUT*	Exception Cycle	Bus Timeout. **TIMOUT*** is asserted by the CSM to signal that an agent is taking too much time to respond to a handshake.
BUSERR*	Exception Cycle	Bus Error. An agent activates **BUSERR*** to indicate its detection of a data integrity problem during a transfer. The CSM monitors this signal as part of its bus timeout function.
BCLK*	Central Control	10MHz Bus Clock. Driven only by the CSM to provide all system timing references.
CCLK*	Central Control	20MHz Central Clock. Driven only by the CSM as an auxiliary clock for use as an additional timing reference among bus agents.
RST*	Central Control	Reset. Driven only by the CSM as a system-level initialization signal.
DCLOW*	Central Control	DC Power Low. Driven only by the CSM as a warning to system agents of an imminent power failure. Part of the CSM reset generation function.
PROT*	Central Control	Protect. Driven only by the CSM during power-fail sequences. Part of the reset generation function.

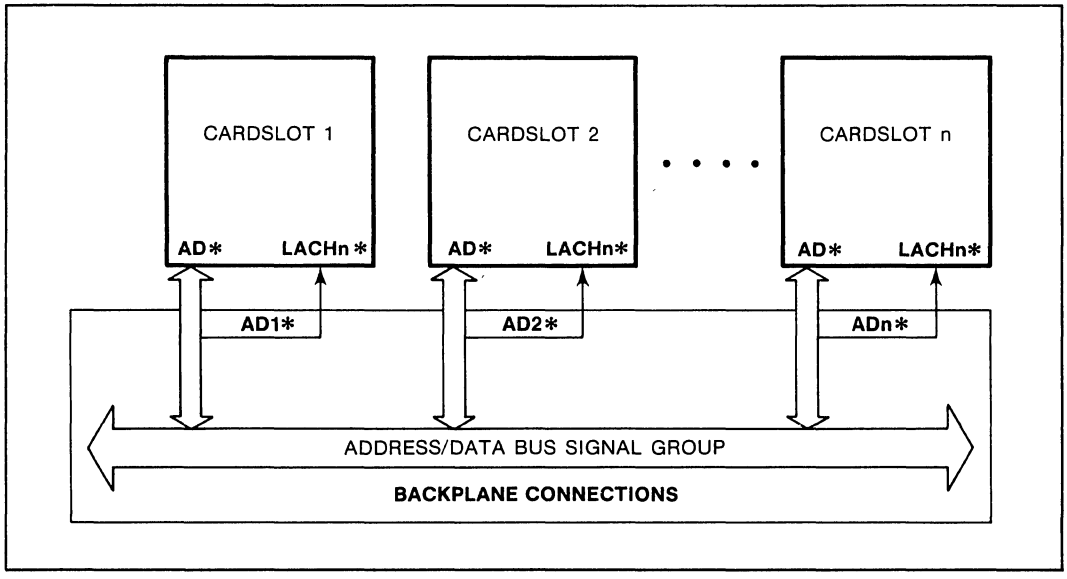


Figure 2-1. Backplane Connection of LACHn*

guarantee t2 (high time), t4 (low time), and t5 (period). The circuits must also guarantee t6 (clock-to-clock) and the correct phase relationship between clock signals. Signal parameters t1 (rise time) and t3 (fall time) must be met by the buffer device driving the clock signals onto the PSB bus interface.

Note:
The **BCLK2*** and **CCLK2*** signals are only required for backplanes containing more than 12 cardslots.

The CSM clock functions can be implemented by use of a crystal oscillator, frequency divider, and two or more bus drivers.

3.2 Reset Generator (RSTGEN) Function

The IEEE 1296 specification defines three types of reset sequences for the CSM: cold, warm and recovery. The **RST***, **DCLOW*** and **PROT*** signals are used to encode the reset type. The **DCLOW*** and **PROT*** signals are defined as being asynchronous while the **RST*** signal is defined as being synchronous.

The **ACLOW*** input is only from the power supply in systems supported by battery backup (VBB) and is required for power fail and recovery resets. The **WARM** and **COLD** inputs represent user-controlled signals for use in generating warm or cold resets. They might be supplied from a system front panel or via a status register in the agent's interconnect space. **BCLK1*** is an input to clock the synchronous **RST*** signal.

Tables 3-2 through 3-4 and figures 3-3 through 3-5 list and depict the timing specifications for the cold, warm and power-fail recovery resets, respectively.

There are various system and user defined parameters beyond the scope of this article which can be added to the design and implementation of the RSTGEN function. Exploring the flexibility presented by such additional factors as the characteristics of the **ACLOW*** signal, whether or not to support battery backup, the ramp-up time of the power supply, the number of front panel or user inputs, or which options to permit when multiple resets occur simultaneously are left for the design engineer's consideration.

The RSTGEN function described above can be implemented using voltage monitors, timers and basic control logic.

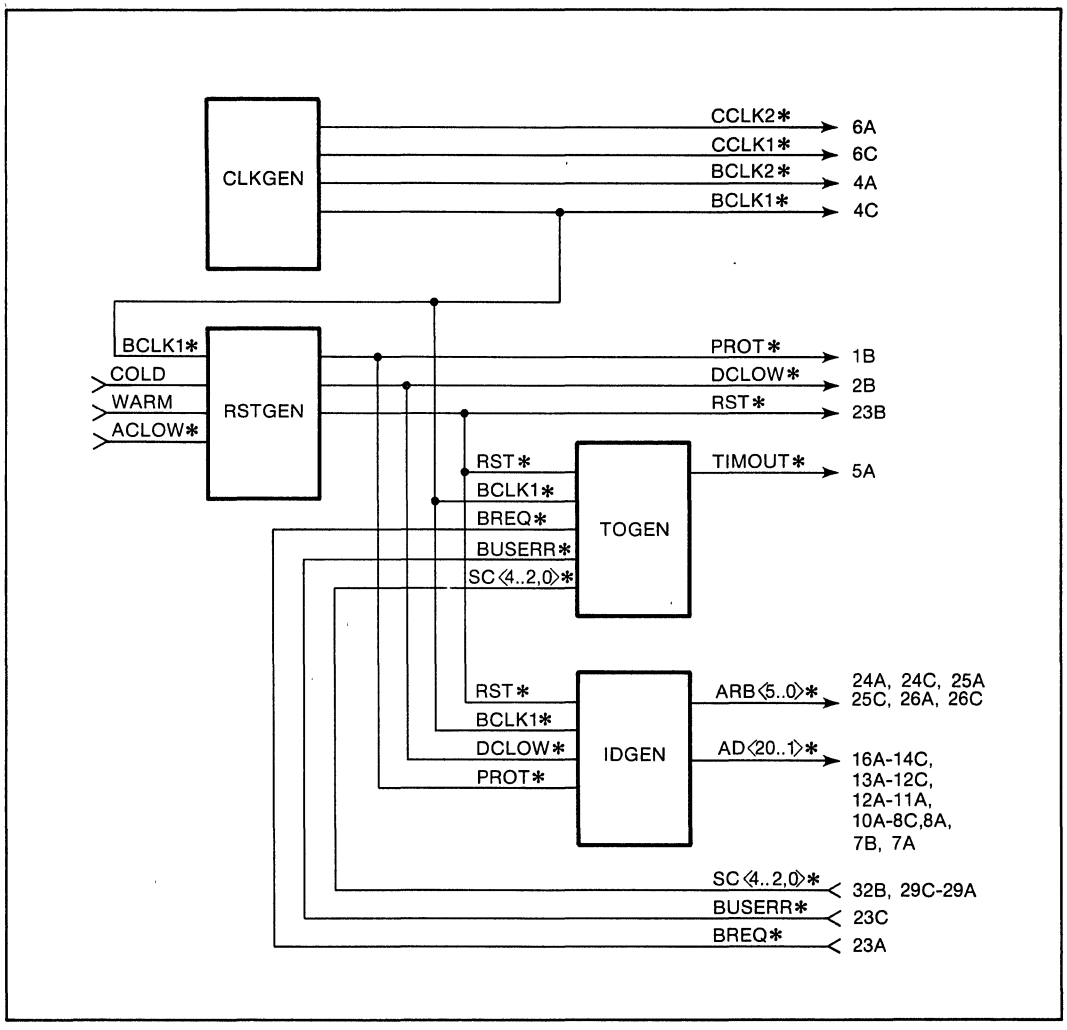


Figure 3-1. CSM Functional Block Diagram

Table 3-1. CSM Clock Timing Specification

Parameter	Parameter Description	BCLK*		CCLK*		Units
		Min	Max	Min	Max	
t1	Rise Time	—	2.0	—	2.0	ns
t2	High Time	48.0	52.1	23.0	27.0	ns
t3	Fall Time	—	2.0	—	2.0	ns
t4	Low Time	48.0	52.1	23.0	27.05	ns
t5	Period	99.9	100.1	49.95	50.05	ns
t6	Clock-to-Clock	0	+10	—	—	ns

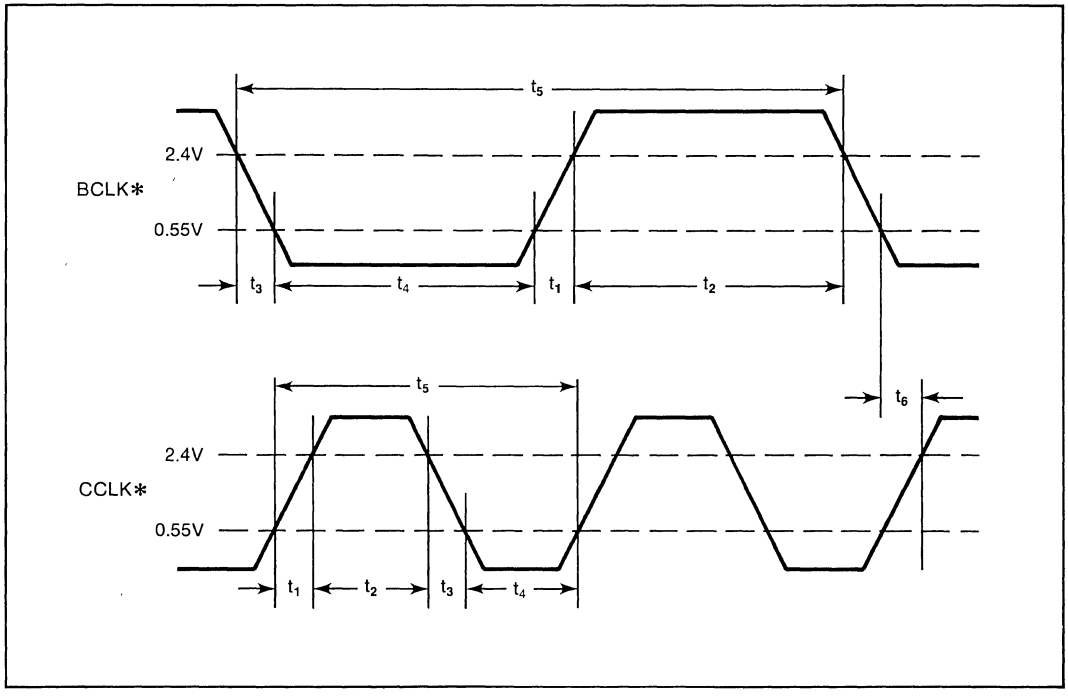


Figure 3-2. Clock Timing Relationships at CSM Connector P1

Table 3-2. CSM Cold Reset Timing Specifications

Parameter	Parameter Description	Min	Max	Units
t1	DC power setup to **DCLOW***	—	1.0	ms
t2	Cold reset duration	2.5	—	ms
t3	Warm reset duration	50.0	—	ms

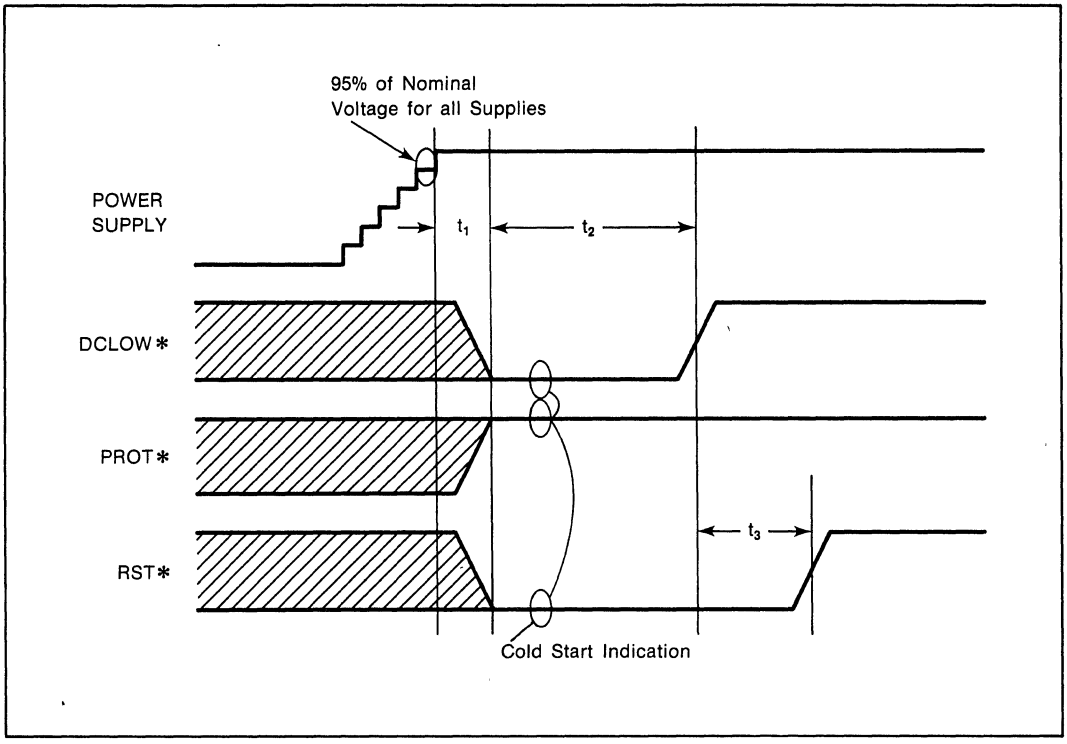


Figure 3-3. Cold Reset Timing on the PSB

Table 3-3. CSM Warm Reset Timing Specification

Parameter	Parameter Description	Min	Max	Units
t1	**RST*** pulse width	50.0	—	ms

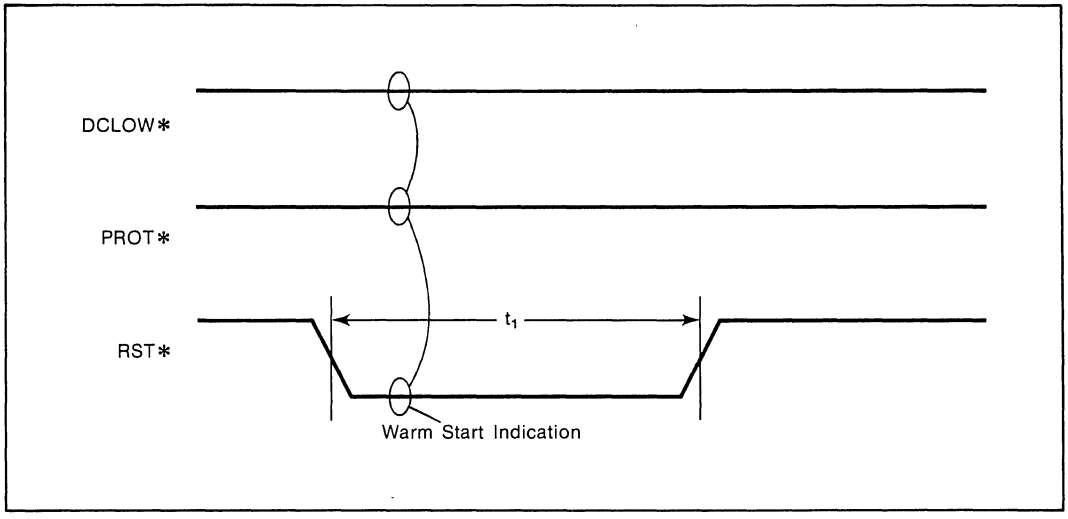


Figure 3-4. Warm Reset Timing on the PSB

Table 3-4. CSM Power Fail and Recovery Timing Specification

Parameter	Parameter Description	Min	Max	Units
t1	DC power hold from **DCLOW***	6.5	—	ms
t2	**PROT*** delay from **DCLOW***	6.0	6.25	ms
t3	DC power setup to **DCLOW***	1.0	—	ms
t4	**RST*** delay from **DCLOW***	6.5	7.0	ms
t5	**RST*** setup from **DCLOW***	0.5	—	ms
t6	**RST*** active from **PROT***	50.0	—	ms
t7	**DCLOW*** pulse width	7.5	—	ms
t8	**PROT*** hold from **DCLOW***	2.0	2.5	ms

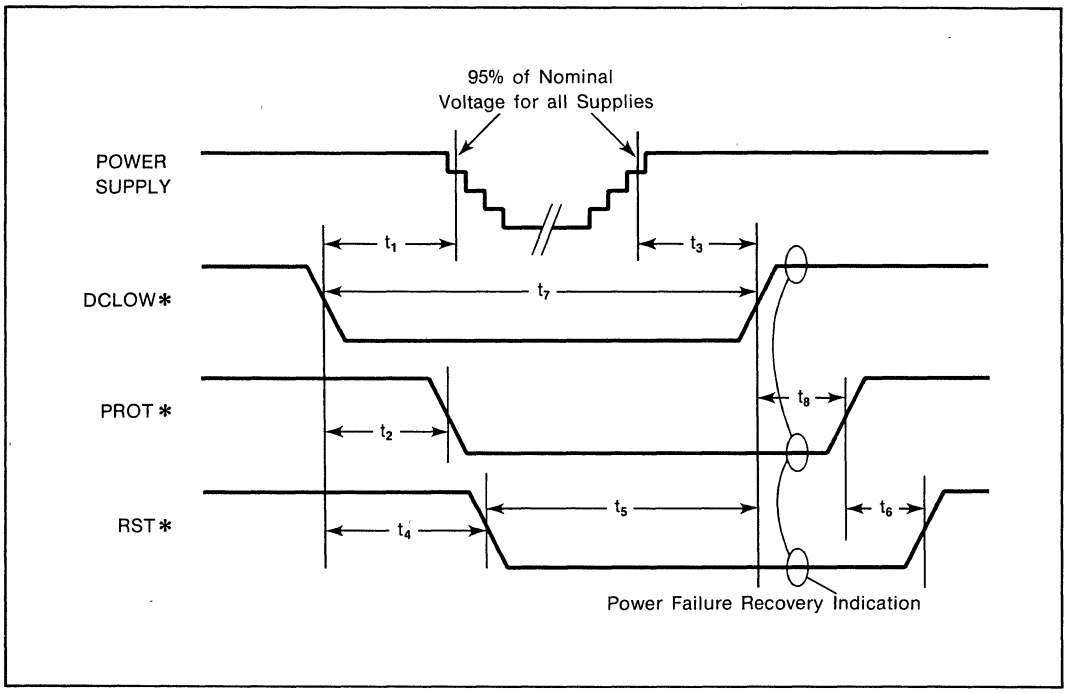


Figure 3-5. Power-Fail Recovery Timing on the PSB

3.3 Timeout Generator (TOGEN) Function

TIMOUT* signal generation for both the Transfer Cycle and the bus ownership cases will be considered. Since all input and output signals are synchronous, **BCLK*** is required.

3-3.1 TRANSFER CYCLE TIMEOUT

The IEEE 1296 specification defines a Transfer Cycle timeout period as 10,000-12,500 counts of **BCLK*** (nominally 1-1.25 ms). A timer or counter and control logic can be used to implement this function. If test frequencies are desired for **BCLK***, then a counter may be a more desirable solution so the Transfer Cycle timeout period (TOP) will be a function of **BCLK*** and not fixed at 1 ms.

Timing of the Transfer Cycle begins on the first clock of a request phase; indicated by **SC0*** active. Once initiated and unless one of the following conditions is satisfied, the Transfer Cycle TOP will have expired and the CSM must assert **TIMOUT***:

a. **SC2*** AND **SC4*** low AND **SC0*** high during a reply phase. This condition indicates requestor end of transfer (EOT) and replier ready handshake, which terminates Transfer Cycle timing.

b. **BUSERR*** low. This signal unconditionally initiates an Expection Cycle which ends the Transfer Cycle and stops the counter.

c. **SC3*** AND **SC4*** low AND **SC2*** AND **SC0*** high during a reply phase. This condition indicates handshake without EOT and the Transfer Cycle TOP needs to be restarted.

d. **RST*** low. This condition terminates all bus activity.

The state-flow diagram in figure 3-6 symbolizes the control logic necessary to assert **TIMOUT*** during a Transfer Cycle. The transition from one state to the next is assumed to be synchronous with **BCLK***.

In the IDLE state, Transfer Cycles are not in progress. The conditions for entering the IDLE state are: system reset (**RST*** low) OR exception cycle (**BUSERR*** low) OR EOT handshake in reply phase (**SC2*** AND **SC4*** low AND **SC0*** high).

The condition for transitioning to the START state is the start of a Transfer Cycle (**SC0*** low). The START state is used to initialize the Transfer Cycle TOP counter before transition to the WAIT state. The START state always transitions to the WAIT state.

In the WAIT state, either Transfer Cycle TOP expires or a condition where handshake without EOT occurs. If the Transfer Cycle TOP has expired, transition is to the TO state and the signal **TIMOUT*** is activated. If handshake without EOT occurs (**SC3*** AND **SC4*** low AND **SC2*** AND **SC0*** high), transition is back to the START state to reinitialize the Transfer Cycle TOP counter and then returns to the WAIT state.

The TO state always transitions back to the IDLE state. Thus, in this design, **TIMOUT*** is asserted for one **BCLK***.

3.3.2 BUS OWNERSHIP TIMEOUT

The IEEE 1296 specification identifies the bus ownership timeout as system defined. A timer or counter and control logic can also be used to implement this function.

Timing of bus ownership begins with the assertion of **BREQ*** low and ends when **BREQ*** high OR **RST*** low. If neither of these two conditions occur before the TOP expires; then the signal **TIMOUT*** is asserted.

The state-flow diagram in figure 3-7 symbolizes the control logic necessary to assert **TIMOUT***

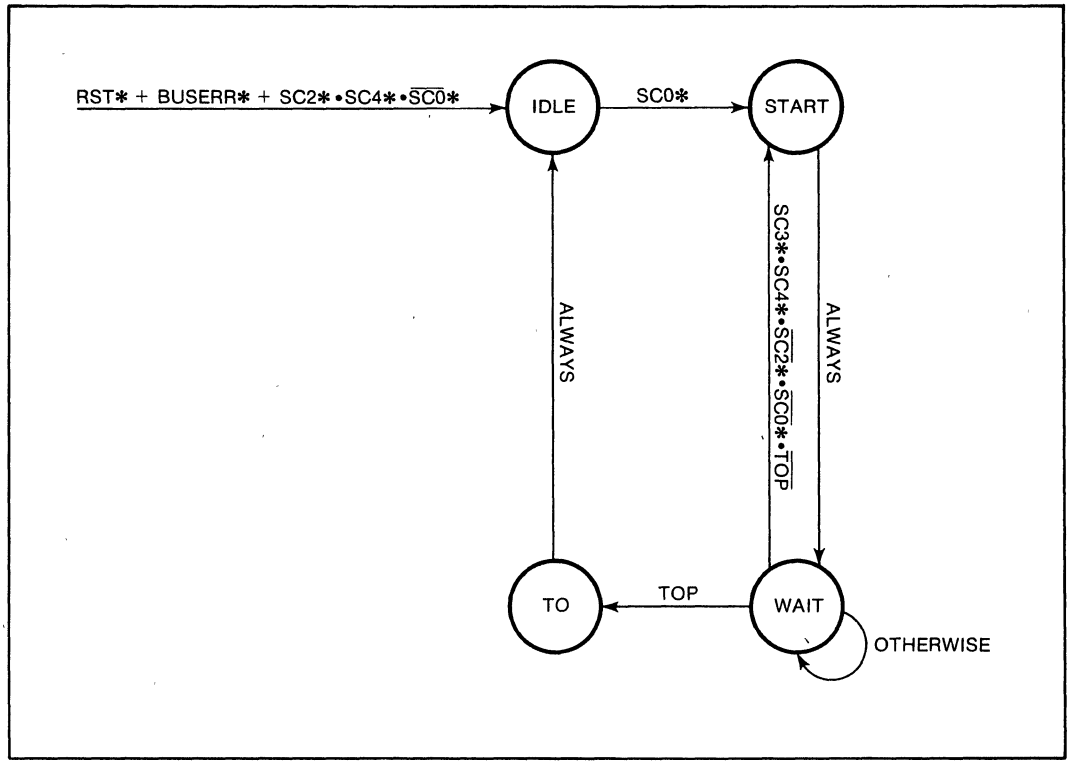


Figure 3-6. State-Flow Diagram for Monitoring Transfer Cycle Timeout

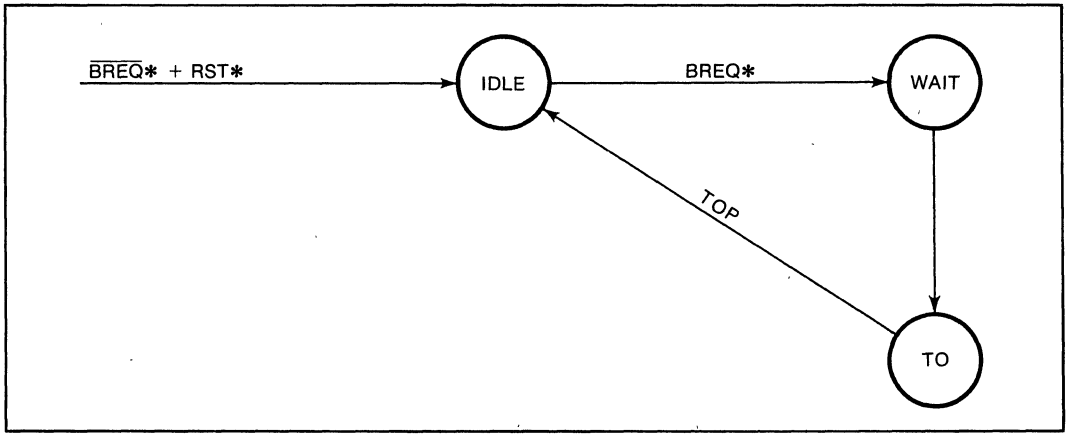


Figure 3-7. State-Flow Diagram for Monitoring Bus Ownership Timeout

for bus ownership. The IDLE state is transitioned to whenever **BREQ*** high OR **RST*** low. Once **BREQ*** low, a transition to the WAIT state is made. If **BREQ*** fails to go high before TOP expires, transition is to the TO state and the signal **TIMOUT*** is activated.

In implementing the Transfer Cycle and bus ownership timeout functions, it may be desirable to differentiate between the two signals via a status register in the agent's interconnect space. Requesting agents on the PSB bus could then determine which type of reset occurred.

3.4 Cardslot and Arbitration ID Assignment Generator (IDGEN) Function

The total number of cardslot and arbitration IDs to be assigned is determined by the number of agents in a system (up to a maximum of 21). Also, the implementation logic requirements are reduced when the system contains fewer agents, but for this discussion the maximum number of agents is assumed.

The default assignment of cardslot and arbitration IDs are listed in table 3-5. The CSM timing relationships shown in figure 3-8 are duplicated from the IEEE 1296 specification for reference. Not indicated in the table or figure is the requirement that each ID be setup one **BCLK*** before and held one **BCLK*** after the **BCLK*** in which **LACHn*** is active.

The **LACHn*** for each cardslot equals its corresponding **ADn*** and assuming the ID assignments will be made in ascending numerical order (**AD1*** . . . **AD20***), a shift register would be a satisfactory method for driving the Address/Data lines. The IDs themselves lend nicely to sequential logic or a table scheme. The remaining circuit requirements are control logic to provide at least eight counts of **BCLK*** delay following **RST*** going active before ID assignment begins (per IEEE 1296 specification), and to coordinate the Address/Data line shift register with the ID sequencer logic. The agent's Address/Data line buffer control logic must allow the CSM to enable the buffers on the PSB during ID assignment.

3.4.1 EIGHT COUNT BCLK* DELAY

A simple way to implement an eight count **BCLK*** delay before ID assignment begins is symbolized in the state-flow diagram shown in figure 3-9. Waiting until **DCLOW*** AND **PROT***

Table 3-5. Default Cardslot and Arbitration ID Values

Cardslot	**ADn***	Cardslot ID **ARB<5 . . 0>***	Arbitration ID **ARB<5 . . 0>***
0	—	LHHHHH	HLLLLL
1	1	LHHHHL	HLLLLH
2	2	LHHHLH	HLLLHL
3	3	LHHHLL	HLLLHH
4	4	LHHLHH	HLLHLL
5	5	LHHLHL	HLLHHL
6	6	LHHLLH	HLLHHH
7	7	LHHLLL	HLHLLL
8	8	LHLHHH	HLHHLL
9	9	LHLHHL	HLHHHL
10	10	LHLHLH	HLHHHH
11	11	LHLHLL	HHLLLL
12	12	LHLLHH	HHLLLH
13	13	LHLLHL	HHLLHH
14	14	LHLLLH	HHLHHH
15	15	LHLLLL	HHHLLL
16	16	LLHHHH	HHHLLH
17	17	LLHHHL	HHHLHH
18	18	LLHHLH	HHHHLL
19	19	LLHHLL	HHHHLH
20	20	LLHLHH	HHHHHL

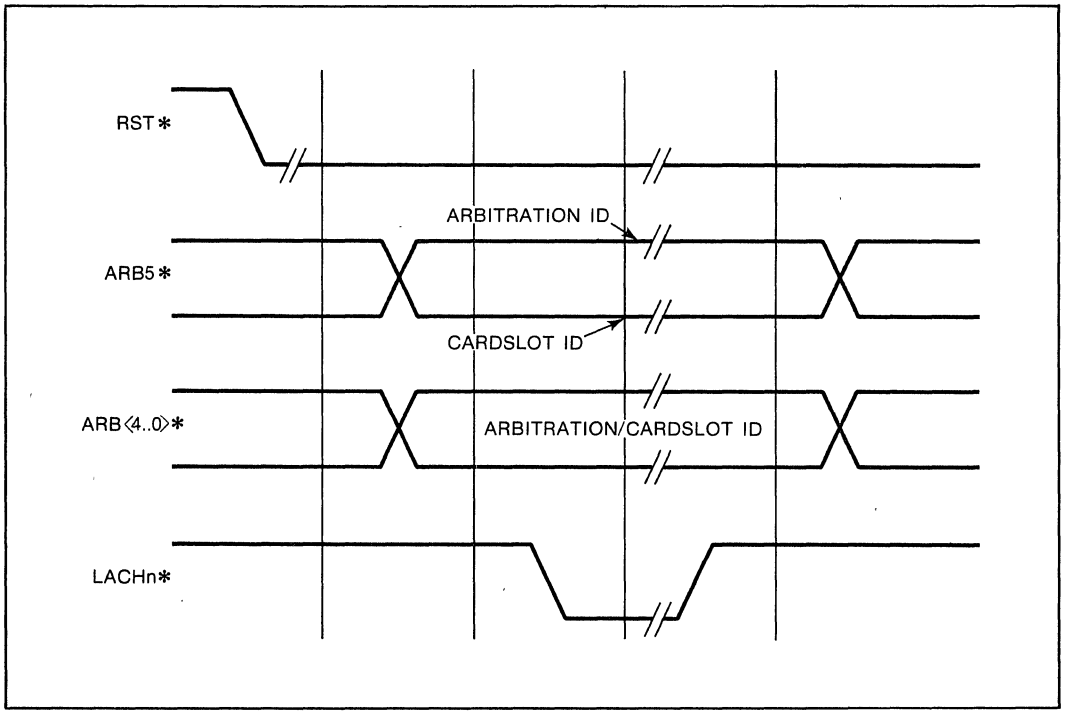


Figure 3-8. Cardslot and Arbitration ID Assignment Timing

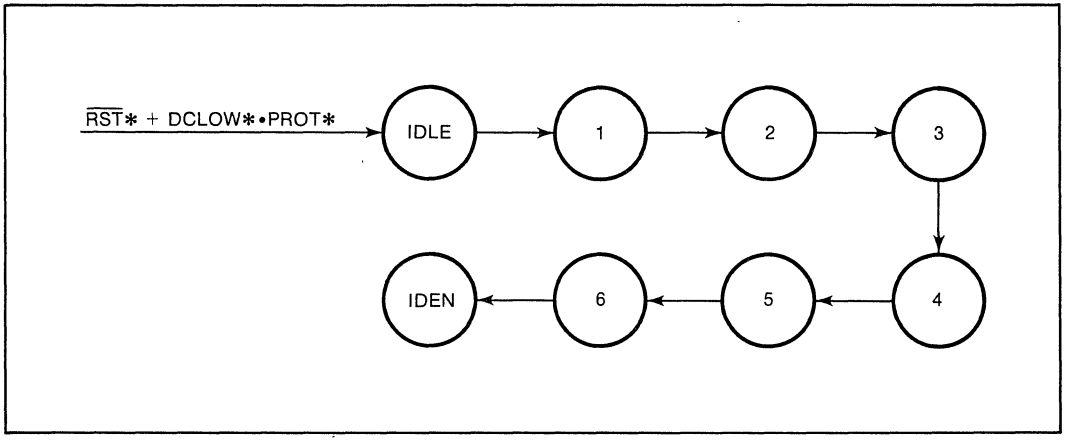


Figure 3-9. State-Flow Diagram for Delaying ID Assignment

are inactive before starting the counter ensures that the power source has stabilized following power-up and recovery resets. Since a warm reset occurs only during normal system operation and is indicated by **RST***, the **RST*** low condition is used to transition to the first count state.

By using **RST*** high to keep the counter in the IDLE state, the IDEN state can be used in Arbitration and Address/Data line PSB buffer control because the CSM is the only driver of these lines during a reset sequence. The implemented buffer control circuit will necessarily depend on the type of agent the CSM resides on and the type of buffers used.

3.4.2 ID SEQUENCER LOGIC

The state-flow diagram in figure 3-10 symbolizes the control logic necessary for controlling the ID sequencer. The sequencer is in an IDLE state until the **BCLK*** counter of figure 3-9 is in the IDEN state; then SETUP, LATCH and HOLD states are necessary for each ID. Assuming the Address/Data PSB buffers are enabled during the LATCH state, the shift register driving the Address/Data lines with the **LACHn*** signal could be timed in either the SETUP or HOLD states. The sequencer continues through the SETUP, LATCH, and HOLD states until all of the IDs have been assigned.

4.0 DESIGN EXAMPLE

The CSM functions previously described in the functional overview (excepting bus ownership timeout) have been implemented in the design example presented and described in the paragraphs to follow.

Because the agent hosting the CSM determines the type of line receivers and drivers used, the functional block diagram (figure 3-1) is modified to include a parallel system bus interface (PBI) function (see figure 4-1). The PBI function defines the buffer structure for CSM input/output operations, electrically isolates the other CSM circuits from the PSB interface, and further modularizes the design.

The circuits assembled to perform the CSM functions in the design example are shown schematically in figure 4-2. The remainder of this section describes signal processing for each of the major functional groups and references are made to figure 4-2 by sheet number only. For usability, the figure is located at the end of this section following figure 4-8. For simplicity, the decoupling capacities have been omitted from the schematic.

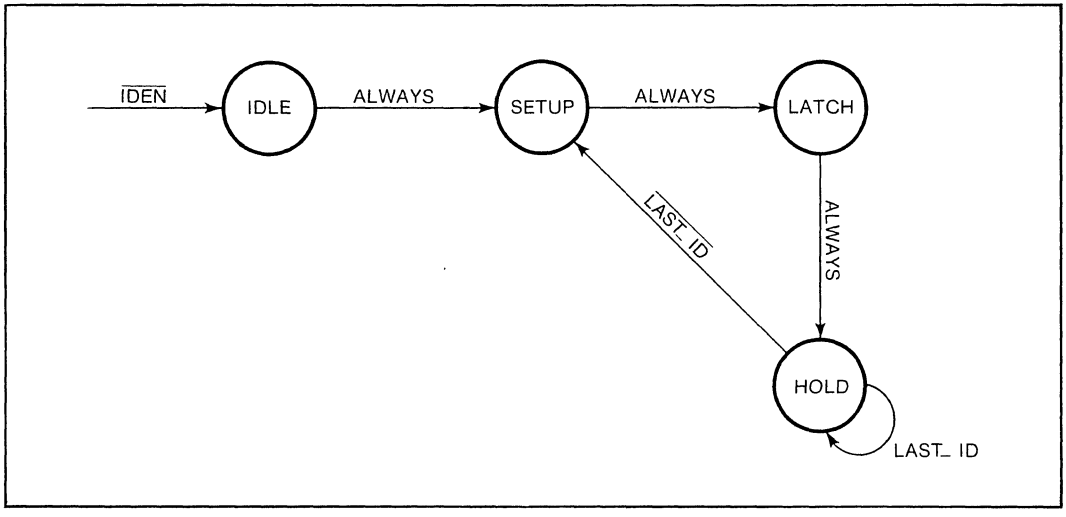


Figure 3-10. State-Flow Diagram for Controlling ID Assignment

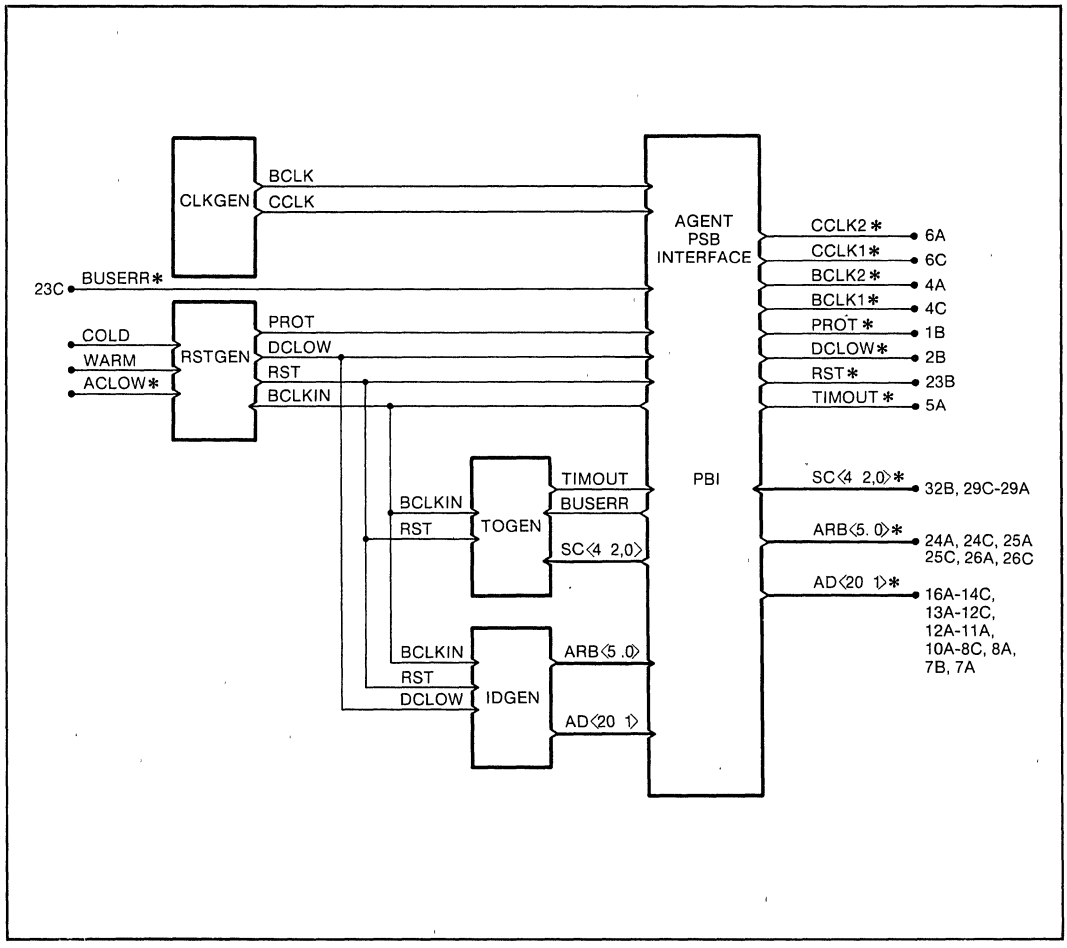


Figure 4-1. CSM Overall Block Diagram

4-1 PSB Buffer Interface (Sheet 1)

The majority of the PSB interface signals are buffered using 74F240 packages. The 74F240 circuitry meets the PSB loading and drive specifications and are satisfactory for this design example. The **ARB<5 . . 0>*** lines are specified to operate open-collector and the 74S38 gates used meet this requirement. The **BUSERR*** input is buffered through a spare 74AS1004 gate simply to keep the part count down.

Since the CSM is the only driver of the clock, reset and timeout signals, their buffer enables are tied active. The **AD<20 . . 1>*** lines are only driven by the CSM during reset sequences and the **ADEN*** signal from the IDGEN circuit is used to enable these lines. Also provided by the IDGEN circuit is the **IDEN** signal to enable the 74S38 gates during ID assignments. **BCLK1*** is buffered through a 74AS1004 gate as **BCLKIN** and provides the on-board timing for the synchronous logic.

4.2 CLOCK GENERATOR (CLKGEN) (Sheet 2)

The 40MHz oscillator, 74AS163 package and two 74AS1004 gates form the CLKGEN circuit. Dividing the 40MHz to 20MHz and 10MHz in the same package controls clock-to-clock skew. Since all clock signals are driven by the same 74F240 package (sheet 1) and assuming a less than 0.5ns same-package skew from both the 74AS163 and 74F240 packages, the worst case clock-to-clock is 1ns (ignoring trace layout considerations). The IEEE 1296 specification defined clock-to-clock skew is listed in table 3-1 as parameter t6. By using the 74AS1004 gates in the 20MHz path to delay **CCLK**, the clock-to-clock minimum and maximum times are met:

clock-to-clock min = 2 x mintpd 74AS1004 - worst case package skew
= 2ns - 1ns
= 1ns

clock-to-clock max = 2 x maxtpd 74AS1004 + worst case package skew
= 8ns + 1ns
= 9ns

Trace routing and loading on the clock signals are critical to proper CSM operation. The loading and trace layout should be kept as close to identical as possible to minimize skew. If analysis reveals that skew is greater than allowed, additional steps would need to be taken to reduce it.

Note that in the design example, the counter will reach a count of 0FH (15) after power-up before the circuit starts to produce the proper **BCLK*** and **CCLK*** waveforms.

4.3 Reset Generator (RSTGEN) (Sheet 2)

As described in paragraph 3.2, the RSTGEN function is influenced by the power supply used and system configuration. For simplicity, the design example assumes the following:

- No battery back-up capability
- Equal ramp-up time on all power supply levels
- **ACLOW*** input from the power supply signalling eminent power failure
- Two active high debounced inputs for cold and warm reset invocation by the system user.

The TL7705A and 74AS74 packages on sheet 2 and part of the PAL16R4B programmable logic device on sheet 3 form the RSTGEN circuit. The timing for **DCLOW** and **ARST** are provided by the CT inputs on the TL7705A power supply monitors, which are adjustable to meet different power supply ramp-up times. The timing provided at the CT input can be determined by adding the power supply ramp-up time from the monitor threshold at 90 percent VCC to the minimum pulse width of **DCLOW** or **ARST**. (Note that the pulse width for **ARST** should also include the pulse width for **DCLOW**.) For this example, a 5ms ramp-up from 4.5- to 4.75-Vdc was assumed.

The 74AS74 packages synchronize **SRST** (later developed into **RST***) for the PSB interface and **SDCLOW** for the state machine in the IDGEN circuit.

The **RIDCLOW*** and **RIRST*** signal inputs to the TL7705A packages are generated in the PAL16R4B simply to reduce the part count. The PAL16R4B equations, shown in figure 4-3, are basically that of OR gates.

4.4 Timeout Generator (TOGEN) (Sheet 3)

The 74S779 counters and part of the PAL16R4 implement the Transfer Cycle timeout function. Using the **TIMOUT**, **S1** and **CET*** outputs of the PAL16R4 as the state bits in figure 3-6, the state assignments provided in table 4-1 control the counters and assert **TIMOUT**.

In the configuration shown on sheet 3, the counters provide 10,240 counts of **BCLK*** and when combined with the state machine, yield a timeout period of 10,243 counts of **BCLK***. The count can be fine tuned by adjusting the inputs to the counters. The equations for the PAL16R4 are shown in figure 4-3.

4.5 Cardslot and Arbitration ID Generator (IDGEN)

The 63RA481A® PROM (ID sequencer) on sheet 3 and the 74LS164 AD* shift registers and PAL16R8 on sheet 4 form the IDGEN circuit. Note that to keep the part count down, the shift register for driving **AD<17 . . 20>*** is implemented in the PAL16R8. Also implemented in the PAL16R8 is the IDEN state machine depicted in figure 3-9. The implemented IDEN state machine only provides one **BCLK*** delay instead of eight, but the ID sequencer provides the additional counts

Table 4-1. TOGEN State Assignments

State	TIMOUT	S1	CET*
IDLE	0	0	1
START	0	0	0
WAIT	0	1	0
TO	1	1	1

```
chip name PAL16R4

BCLKIN SRST SC0 SC2 SC3 SC4 BUSERR /ACLOW /TOP GND
/OE /RIRST /RIDCLOW S1 /CET TIMOUT RST COLD WARM VCC

equations

/TIMOUT     :=  /TOP * /TIMOUT * CET
            + /TIMOUT * /S1
            + TIMOUT * S1 * /CET
            + /SC0 * SC2 * SC4
            + BUSERR
            + RST

/S1         :=  TIMOUT * S1 * /CET
            + /SC0 * SC2 * SC4
            + BUSERR
            + RST
            + /SC0 * SC3 * SC4 * /TOP * /TIMOUT * S1 * CET
            + /TIMOUT * /S1 * /CET

CET         :=  /RST * /SC2 * /BUSERR * /TOP * /TIMOUT * CET
            + /RST * SC0 * /BUSERR * /TIMOUT * /S1
            + /RST * /SC4 * /BUSERR * /TOP * /TIMOUT * CET
            + /RST * SC0 * /BUSERR * /TOP * /TIMOUT * CET
            + /RST * /SC4 * /BUSERR * /TIMOUT * /S1 * CET
            + /RST * /SC2 * /BUSERR * /TIMOUT * /S1 * CET

RIDCLOW :=  COLD
         + ACLOW

RIRST   :=  COLD
         + WARM

/RST    :=  /SRST
```

Figure 4-3. Equations for TOGEN PAL16R4B

required by sequencing through unused cardslot IDs. (The signal **IDEN2** is identical to **IDEN** and is used to furnish additional DC drive for the ID sequencer.)

Six outputs of the ID sequencer are used to drive the **ARB<5 . . 0>*** lines and the remaining two outputs are assigned the signal names **LATCH** and **HOLD**. The IDLE state, shown in figure 3-10, is indicated when all of the ID sequencer outputs are low; the SETUP state by the **ARB<5 . . 0>*** outputs changing to a new ID; the LATCH state when the **LATCH** output is active high and the **HOLD** output is inactive low; the HOLD state when the **HOLD** output is active high and the **LATCH** output inactive low. The ID sequencer remains in the last HOLD state until **IDEN** becomes inactive low and is then reset to the IDLE state.

The state-flow diagrams in figures 3-9 and 3-10 are modified as shown in figures 4-4 and 4-5. These modifications take advantage of the design implemented to supply the eight counts of **BCLK*** delay before assigning IDs and to initialize the 74LS164 AD* shift registers.

The AD* shift registers are clocked during the HOLD state of the ID sequencer. Two passes are needed through these registers, one to latch card-

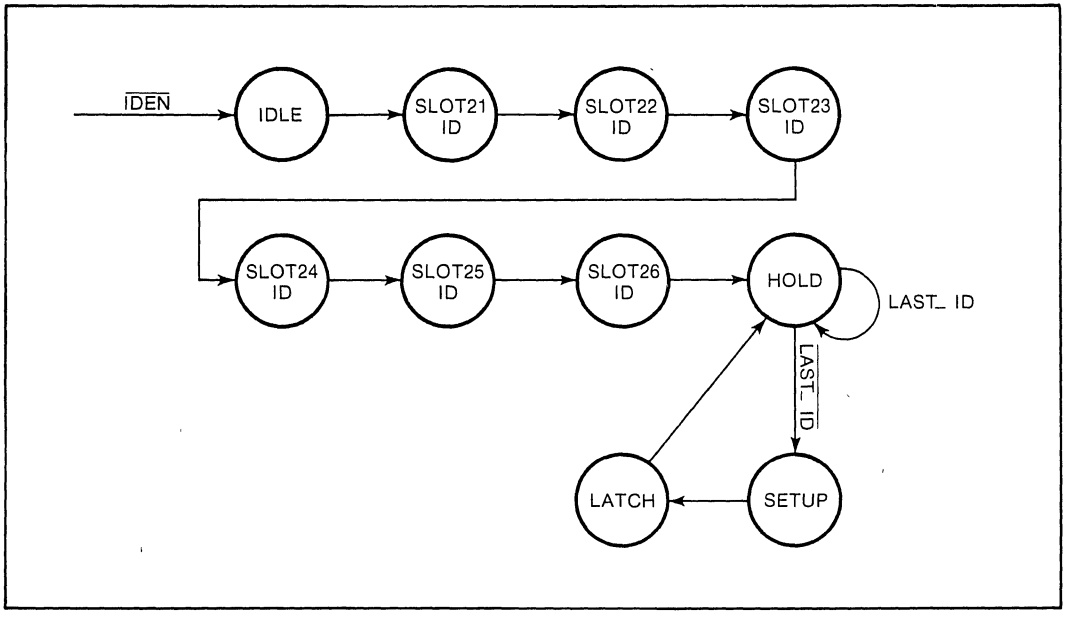


Figure 4-4. Modified State-Flow Diagrams for ID Assignment

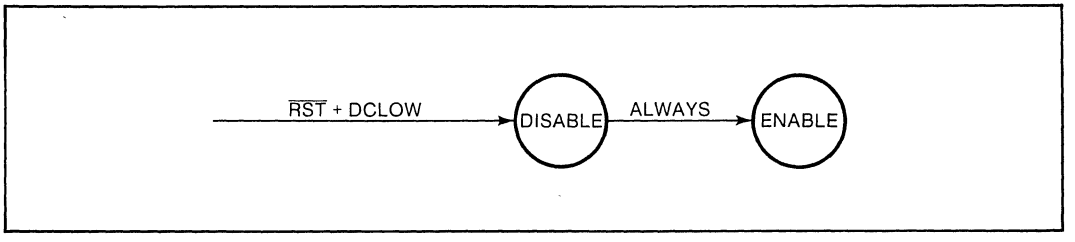


Figure 4-5. Modified IDEN State-Flow Diagram

slot IDs and one to latch arbitration IDs. The AD* shift registers are initialized at zero while **IDEN** is inactive low. During the HOLD state of unused cardslot ID 26 and the HOLD state of cardslot ID 20, the **SDATA** (Serial Data) output of the PAL16R8A is active high so that a one is shifted into the AD* shift registers at the beginning of each pass.

The **LATCH** output from the ID sequencer is inverted to produce **ADEN***. This signal enables the **AD<20 . . 1>*** buffers (sheet 1) onto the PSB bus during the LATCH state.

Figure 4-6 shows the timing produced by the IDGEN circuit. The equations for the PAL16R8 are shown in figure 4-7. The PROM ID code information is provided in figure 4-8.

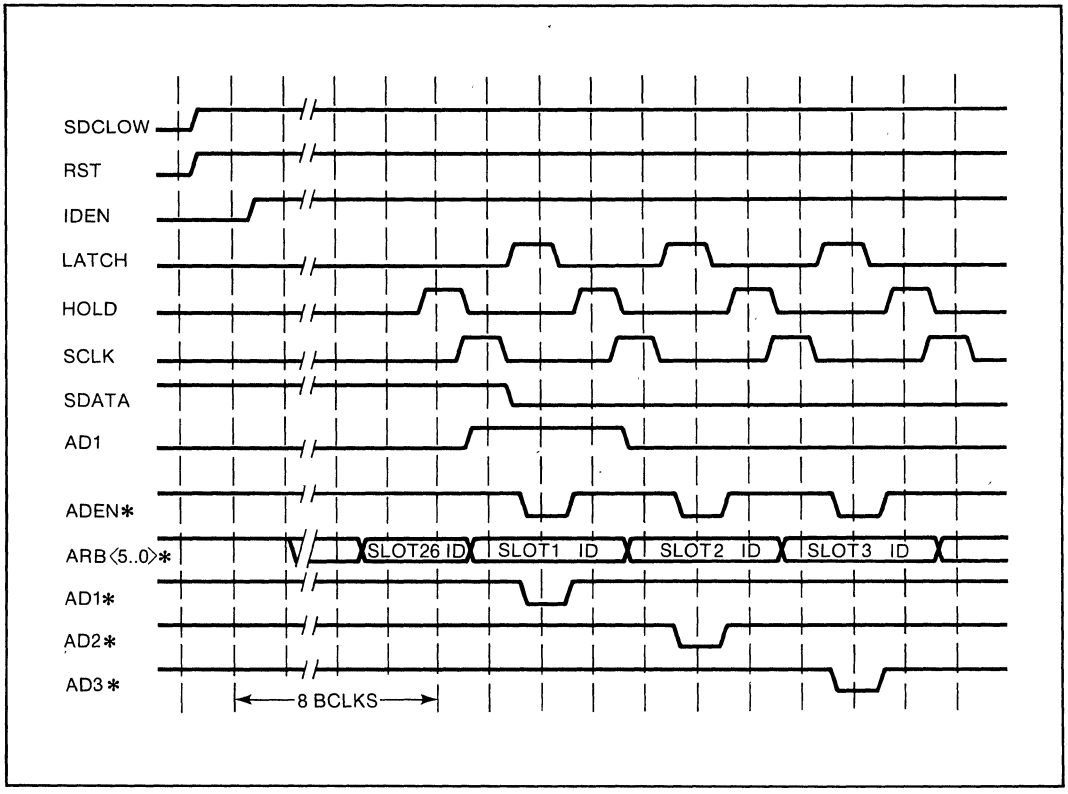


Figure 4-6. IDGEN Timing Diagram

```
chip name PAL16R8
BCLKIN RST SDCLOW HOLD LATCH AD16 IC7 IC8 IC9 GND
/OE AD17 AD18 AD19 AD20 SCLK SDATA IDEN2 IDEN VCC
equations
/IDEN     :=  SDCLOW
          + /RST

/IDEN2    :=  SDCLOW
          + /RST

/SCLK     :=  /HOLD
          + LATCH
          + SCLK
          + /IDEN

/SDATA     :=  /AD20 * IDEN * SCLK
           + IDEN * SCLK * SDATA
           + IDEN * /SCLK * /SDATA

/AD20     :=  HOLD * /LATCH * /AD19 * /AD18 * /AD17
          + /AD20 * /AD19 * /AD18
          + /AD20 * /AD19 * /AD17
          + LATCH * /AD20 * /AD18 * /AD17
          + /HOLD * /AD20 * /AD18 * /AD17
          + /IDEN

/AD19     :=  /AD20 * /AD19 * /AD18
          + /IDEN
          + HOLD * /LATCH * /AD20 * /AD18 * /AD17
          + /AD19 * /AD18 * /AD17
          + LATCH * /AD20 * /AD19 * /AD17
          + /HOLD * /AD20 * /AD19 * /AD17

/AD18     :=  /IDEN
          + /AD19 * /AD18 * /AD17
          + HOLD * /LATCH * /AD20 * /AD19 * /AD17
          + /AD20 * /AD18 * /AD17
          + LATCH * /AD20 * /AD19 * /AD18
          + /HOLD * /AD20 * /AD19 * /AD18

/AD17     :=  /IDEN
          + LATCH * /AD20 * /AD19 * /AD17
          + /HOLD * /AD20 * /AD19 * /AD17
          + /AD16 * /AD20 * /AD19 * /AD17
          + HOLD * /LATCH * /AD20 * /AD19 * /AD18 * AD17
          + /AD20 * /AD19 * AD18 * /AD17
          + /AD20 * AD19 * /AD18 * /AD17
          + AD20 * /AD19 * /AD18 * /AD17
```

Figure 4-7. Equations for IDGEN PAL16R8A

```
ADDRESS     DATA        COMMENTS
00000000    00110101    ;initial state
00110101    00110110    ;setup state for cardslot id 21
00110110    00110111    ;setup state for cardslot id 22
00110111    00111000    ;setup state for cardslot id 23
00111000    00111001    ;setup state for cardslot id 24
00111001    00111010    ;setup state for cardslot id 25
00111010    10111010    ;setup state for cardslot id 26
10111010    00100001    ;hold state for cardslot id 26
00100001    01100001    ;setup state for cardslot id 1
01100001    10100001    ;latch state for cardslot id 1
10100001    00100010    ;hold state for cardslot id 1
00100010    01100010    ;setup state for cardslot id 2
01100010    10100010    ;latch state for cardslot id 2
10100010    00100011    ;hold state for cardslot id 2
00100011    01100011    ;setup state for cardslot id 3
01100011    10100011    ;latch state for cardslot id 3
10100011    00100100    ;hold state for cardslot id 3
00100100    01100100    ;setup state for cardslot id 4
01100100    10100100    ;latch state for cardslot id 4
10100100    00100101    ;hold state for cardslot id 4
00100101    01100101    ;setup state for cardslot id 5
01100101    10100101    ;latch state for cardslot id 5
10100101    00100110    ;hold state for cardslot id 5
00100110    01100110    ;setup state for cardslot id 6
01100110    10100110    ;latch state for cardslot id 6
10100110    00100111    ;hold state for cardslot id 6
00100111    01100111    ;setup state for cardslot id 7
01100111    10100111    ;latch state for cardslot id 7
10100111    00101000    ;hold state for cardslot id 7
00101000    01101000    ;setup state for cardslot id 8
01101000    10101000    ;latch state for cardslot id 8
10101000    00101001    ;hold state for cardslot id 8
00101001    01101001    ;setup state for cardslot id 9
01101001    10101001    ;latch state for cardslot id 9
10101001    00101010    ;hold state for cardslot id 9
00101010    01101010    ;setup state for cardslot id 10
01101010    10101010    ;latch state for cardslot id 10
10101010    00101011    ;hold state for cardslot id 10
00101011    01101011    ;setup state for cardslot id 11
01101011    10101011    ;latch state for cardslot id 11
10101011    00101100    ;hold state for cardslot id 11
00101100    01101100    ;setup state for cardslot id 12
01101100    10101100    ;latch state for cardslot id 12
```

Figure 4-8. IDGEN PROM Content (Sheet 1 of 3)

ADDRESS	DATA	COMMENTS
10101100	00101101	;hold state for cardslot id 12
00101101	01101101	;setup state for cardslot id 13
01101101	10101101	;latch state for cardslot id 13
10101101	00101110	;hold state for cardslot id 13
00101110	01101110	;setup state for cardslot id 14
01101110	10101110	;latch state for cardslot id 14
10101110	00101111	;hold state for cardslot id 14
00101111	01101111	;setup state for cardslot id 15
01101111	10101111	;latch state for cardslot id 15
10101111	00110000	;hold state for cardslot id 15
00110000	01110000	;setup state for cardslot id 16
01110000	10110000	;latch state for cardslot id 16
10110000	00110001	;hold state for cardslot id 16
00110001	01110001	;setup state for cardslot id 17
01110001	10110001	;latch state for cardslot id 17
10110001	00110010	;hold state for cardslot id 17
00110010	01110010	;setup state for cardslot id 18
01110010	10110010	;latch state for cardslot id 18
10110010	00110011	;hold state for cardslot id 18
00110011	01110011	;setup state for cardslot id 19
01110011	10110011	;latch state for cardslot id 19
10110011	00110100	;hold state for cardslot id 19
00110100	01110100	;setup state for cardslot id 20
01110100	10110100	;latch state for cardslot id 20
10110100	00011110	;hold state for cardslot id 20
00011110	01011110	;setup state for arbitration id 1
01011110	10011110	;latch state for arbitration id 1
10011110	00011101	;hold state for arbitration id 1
00011101	01011101	;setup state for arbitration id 2
01011101	10011101	;latch state for arbitration id 2
10011101	00011100	;hold state for arbitration id 2
00011100	01011100	;setup state for arbitration id 3
01011100	10011100	;latch state for arbitration id 3
10011100	00011011	;hold state for arbitration id 3
00011011	01011011	;setup state for arbitration id 4
01011011	10011011	;latch state for arbitration id 4
10011011	00011001	;hold state for arbitration id 4
00011001	01011001	;setup state for arbitration id 5
01011001	10011001	;latch state for arbitration id 5
10011001	00011000	;hold state for arbitration id 5
00011000	01011000	;setup state for arbitration id 6
01011000	10011000	;latch state for arbitration id 6
10011000	00010111	;hold state for arbitration id 6
00010111	01010111	;setup state for arbitration id 7

Figure 4-8. IDGEN PROM Content (Sheet 2 of 3)

```
ADDRESS     DATE        COMMENTS
01010111    10010111    ;latch state for arbitration id 7
10010111    00010011    ;hold state for arbitration id 7
00010011    01010011    ;setup state for arbitration id 8
01010011    10010011    ;latch state for arbitration id 8
10010011    00010001    ;hold state for arbitration id 8
00010001    01010001    ;setup state for arbitration id 9
01010001    10010001    ;latch state for arbitration id 9
10010001    00010000    ;hold state for arbitration id 9
00010000    01010000    ;setup state for arbitration id 10
01010000    10010000    ;latch state for arbitration id 10
10010000    00001111    ;hold state for arbitration id 10
00001111    01001111    ;setup state for arbitration id 11
01001111    10001111    ;latch state for arbitration id 11
10001111    00001110    ;hold state for arbitration id 11
00001110    01001110    ;setup state for arbitration id 12
01001110    10001110    ;latch state for arbitration id 12
10001110    00001100    ;hold state for arbitration id 12
00001100    01001100    ;setup state for arbitration id 13
01001100    10001100    ;latch state for arbitration id 13
10001100    00001000    ;hold state for arbitration id 13
00001000    01001000    ;setup state for arbitration id 14
01001000    10001000    ;latch state for arbitration id 14
10001000    00000111    ;hold state for arbitration id 14
00000111    01000111    ;setup state for arbitration id 15
01000111    10000111    ;latch state for arbitration id 15
10000111    00000110    ;hold state for arbitration id 15
00000110    01000110    ;setup state for arbitration id 16
01000110    10000110    ;latch state for arbitration id 16
10000110    00000100    ;hold state for arbitration id 16
00000100    01000100    ;setup state for arbitration id 17
01000100    10000100    ;latch state for arbitration id 17
10000100    00000011    ;hold state for arbitration id 17
00000011    01000011    ;setup state for arbitration id 18
01000011    10000011    ;latch state for arbitration id 18
10000011    00000010    ;hold state for arbitration id 18
00000010    01000010    ;setup state for arbitration id 19
01000010    10000010    ;latch state for arbitration id 19
10000010    00000001    ;hold state for arbitration id 19
00000001    01000001    ;setup state for arbitration id 20
01000001    10000001    ;latch state for arbitration id 20
10000001    10000001    ;hold state for arbitration id 20
end
```

Figure 4-8. IDGEN PROM Content (Sheet 3 of 3)

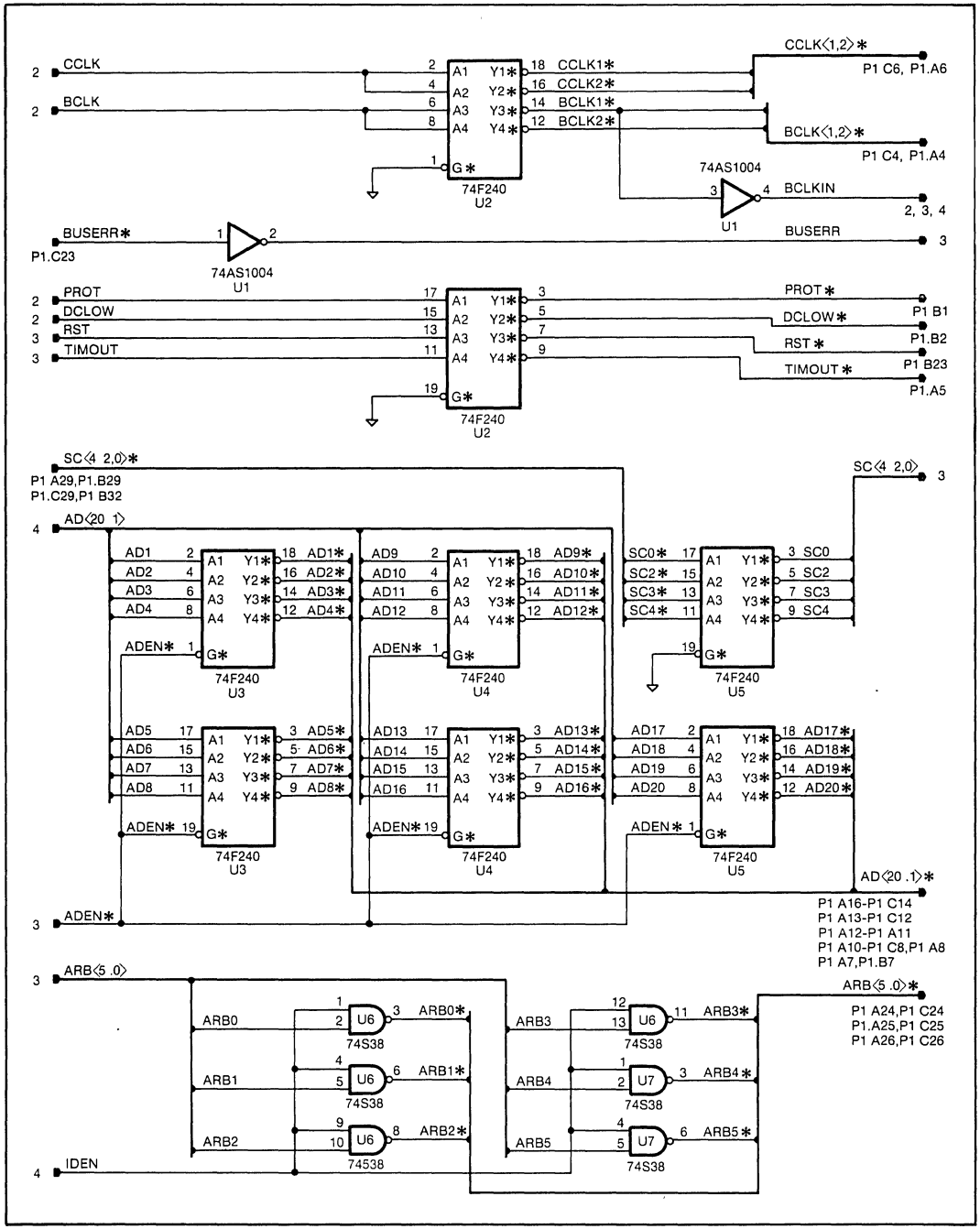


Figure 4-2. CSM Functional Schematic Diagram (Sheet 1 of 4)

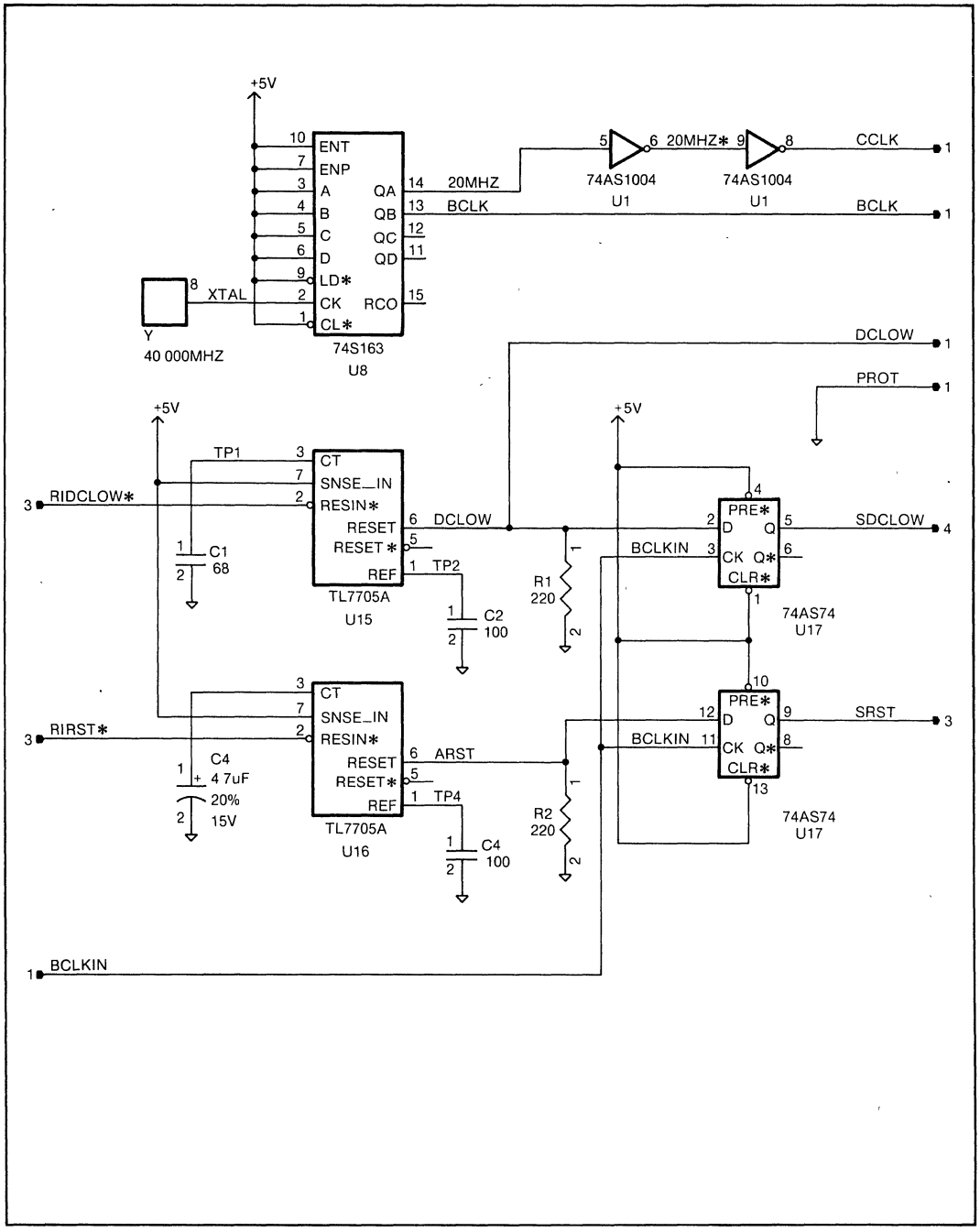


Figure 4-2. CSM Functional Schematic Diagram (Sheet 2 of 4)

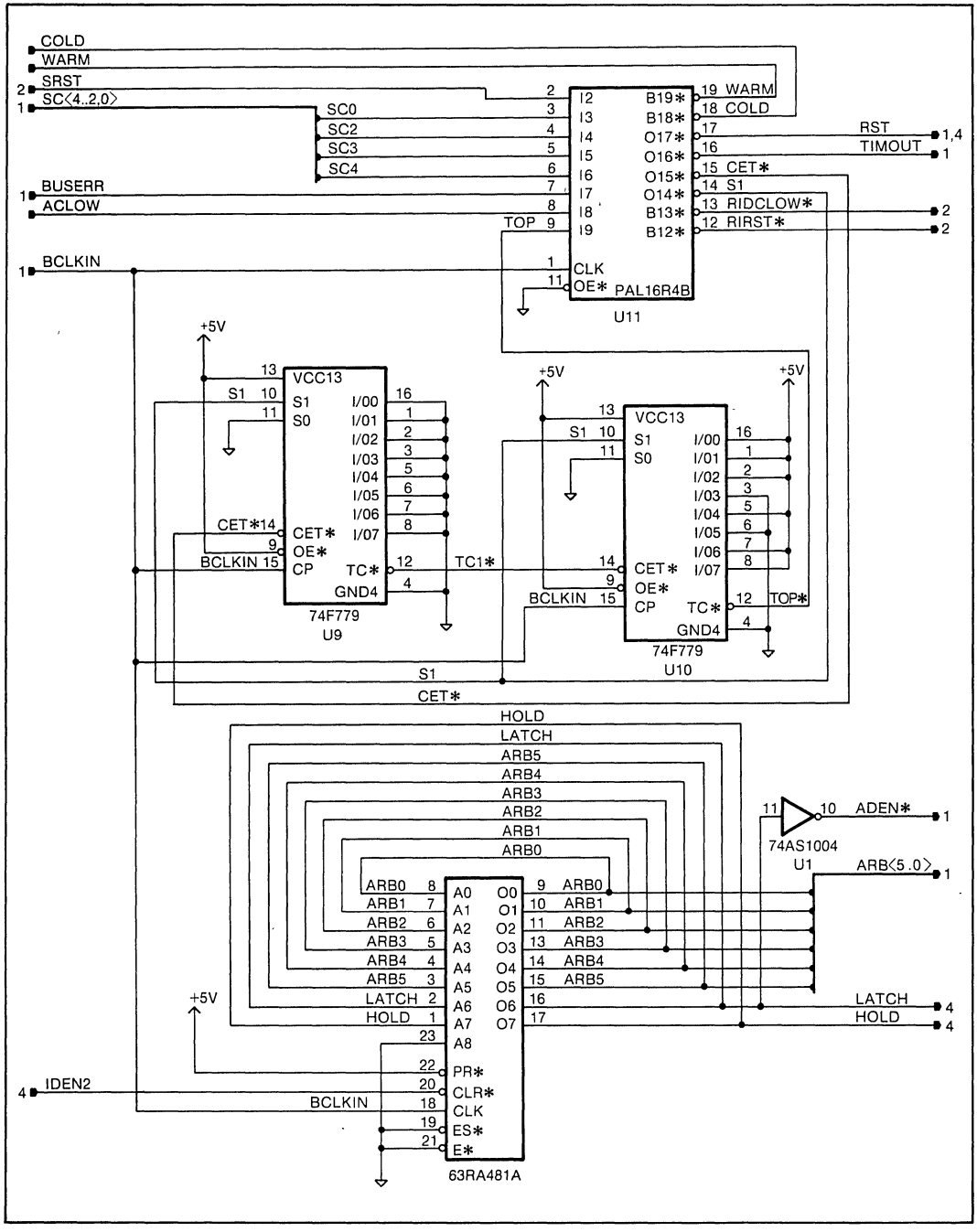


Figure 4-2. CSM Functional Schematic Diagram (Sheet 3 of 4)

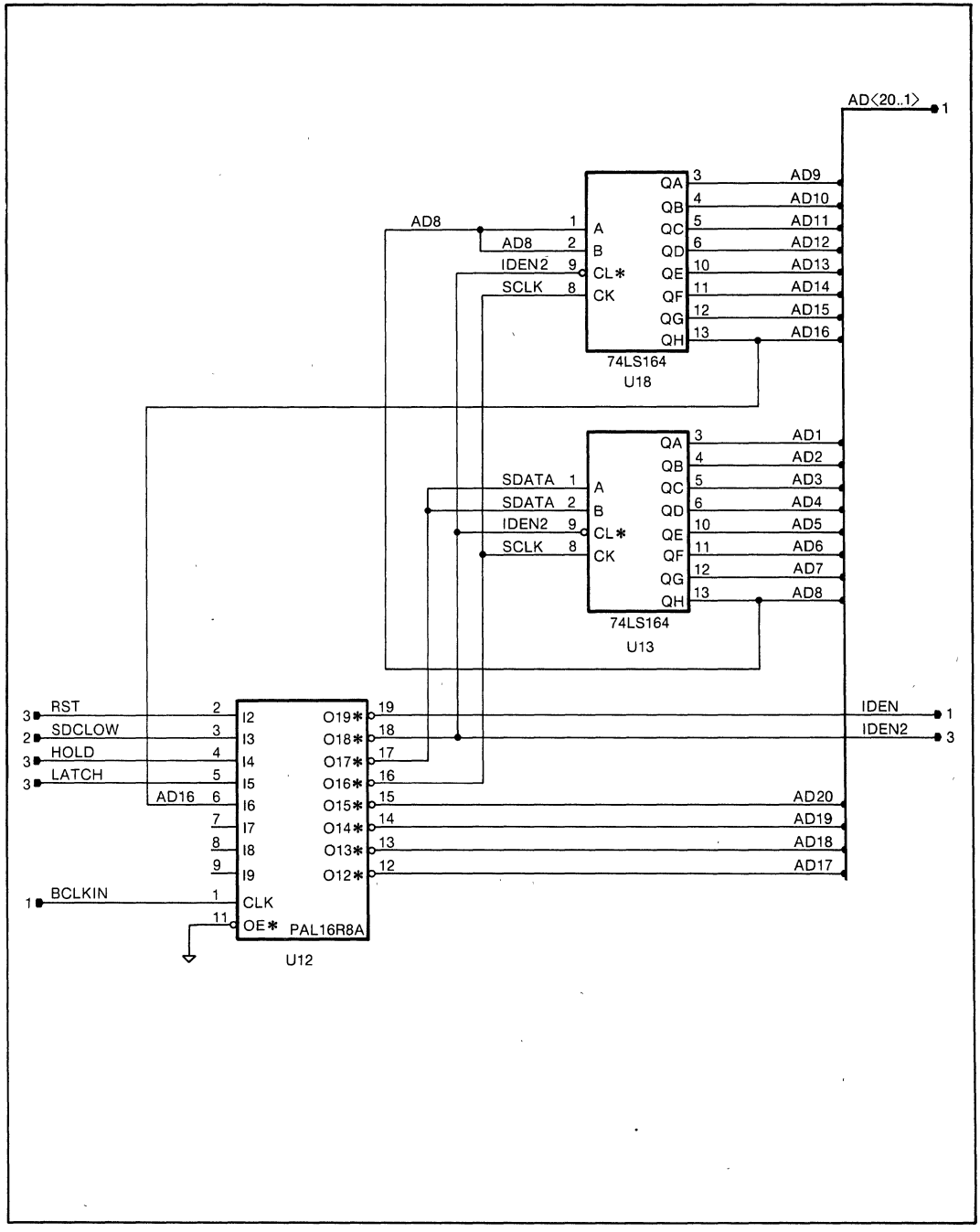


Figure 4-2. CSM Functional Schematic Diagram (Sheet 4 of 4)

5.0 DESIGN CONSIDERATIONS

Additional CSM design considerations are discussed in the following paragraphs. These are cardslot mobility and diagnostic possibilities.

5.1 Cardslot 0 Detection

If the agent hosting the CSM is desired to be cardslot mobile, i.e. operate the cardslots other than 0; then a cardslot 0 detection circuit is necessary to disable the CSM functions when the agent is not in cardslot 0. The 4A pin is 0-Vdc in all cardslots except 0 where it is used to bus clock signal **BCLK*** to the left half of a backplane containing more than 12 cardslots. A simple circuit to detect cardslot 0 is illustrated in figure 5-1.

The signals normally driven only by the CSM would now require buffering using a bi-directional device enabled with a **SLOT0** signal so the agent would receive these signals when not located in cardslot 0. Also, the clock lines will require jumpering as shown in figure 5-2, because the DC signal specifications do not permit driver loading of a signal receiver. The CLKGEN, RSTGEN, TOGEN, and IDGEN circuits would be enabled only when the cardslot 0 detection signal is true.

The **POR*** signal is a power-on reset having a duration guaranteeing stable power supply output levels. The **CLK** signal could be the 20MHz or 10MHz clock output from the CLKGEN circuit or be supplied from any other clock operational before the PSB clock drivers are enabled.

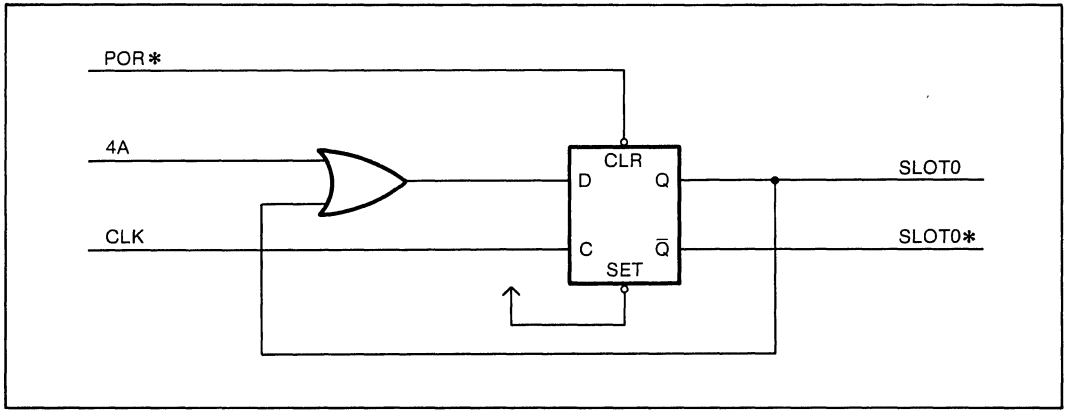


Figure 5-1. Circuit for SLOT0 Detection

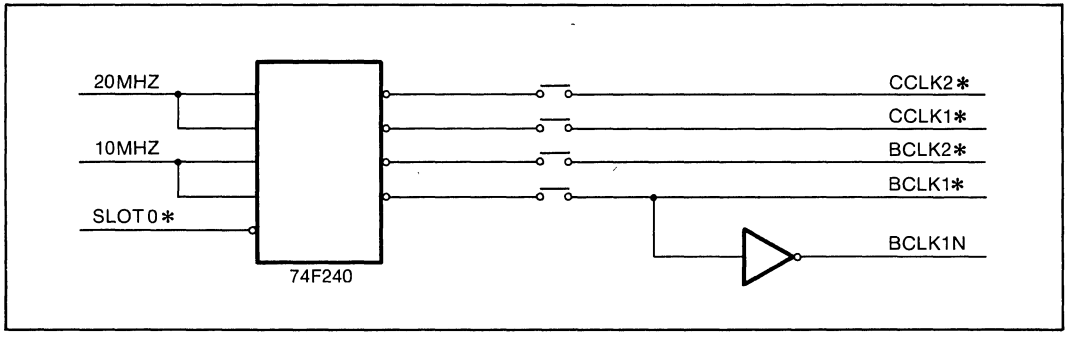


Figure 5-2. CSM Clock Jumper

5.2 System Diagnostic Options

For system test and diagnostic purposes, it may be desirable to provide for the capability of selecting other signal frequencies to exercise the clock drivers. For example, outputs from the frequency divider used in the design example could be selectively jumpered as inputs to the **BCLK*** and **CCLK*** line drivers.

A bus timeout disable function may also be desirable. Such a function can be implemented by providing an additional input to the TOGEN state machine driven by either the interconnect controller or jumper selectable.

5.3 CSM Functions on the Backplane

By removing the PBI function and interfacing directly to an agent's PSB interface, the design example in paragraph 4 could be added to almost any type of MULTIBUS II agent. As mentioned in paragraph 3.0, this agent might be a CPU board or a simple I/O replier device, which may contain other centralized system services.

The IEEE 1296 specification does not preclude putting slot 0 on the reverse side of the backplane. The minimal functionality described here will fit onto a small printed circuit card mounted on the reverse of the backplane — this does, of course, require a backplane designed for this application but if you are trying to squeeze an "extra" slot into a 19 inch rack, this can be accomplished.

intel®

APPLICATION NOTE

AP-423

January 1, 1988

The MULTIBUS®II Interconnect Design Guide

Jory Radke - Modules Development Engineer
Roger Finger - MULTIBUS® II Technical Marketing Manager

Order Number: 280640-001

PREFACE
THE ORIGINS OF INTERCONNECT

Interconnect arose out of a need to make complex technology easier to use. Advanced computer boards of the early 1980's were becoming more sophisticated and complex. Wiring options permitted interrupt routing, memory mapping, EPROM size, and the use of other user installed components. When the jumper count hit 300, it no longer made sense to waste this board space because an inexpensive controller or co-processor could be used to manage resources more intelligently. What evolved was a concept of self-configuration on a scale that had never before been attempted — the standardization of an entire industry.

Our Message Passing Coprocessor (MPC) design team was composed of a group of senior design engineers and architects at the Intel factory in Hillsboro, Oregon and a second group of engineers in Swindon, England who had experience with custom and semi-custom design of Application Specific Integrated Circuit (ASIC) components. The bus interface was partitioned into the primary functions of bus and DMA control, message passing, and interconnect. At Intel, this was implemented with the 82258 Advanced DMA Controller (ADMA) and an 8751 Microcontroller as "partner chips" to the MPC. In the vendor community, all sorts of creative solutions then began to appear. The simplest designs used PROMs or Programmable Array Logic (PAL) to implement the barest subset of the interconnect design specification. To reduce cost of the board, other interconnect designs were based on using the CPU as the interconnect controller. By far the most common approach with intelligent boards was to go with the 8751 (or equivalent) and the MPC. The direct interface between these two controllers predisposes one to selecting these components. But what about firmware?

Early endeavors in microcontroller firmware led to mixed results. Suddenly there were no jumpers to play with and if you wanted to change something, you had to reach inside the firmware. A new set of tools were needed. These tools turned out to be software utilities, operating system services, intelligent device drivers, debuggers, and a system confidence test. With each new board produced, we learned a little more about this amazing environment that had been created as a result of interconnect. An example is the Firmware Communication Record found on many Intel boards. It was originally provided as a "scratch pad" of register space with no dedicated function. To date, it has been used for downloading of code to remote agents, the passing of initialization parameters to I/O controllers, a backplane debugger console, and for issuing requests for bootstrap.

Gradually a core of firmware routines developed which would be reusable on many board types. Jory Radke had the responsibility to develop the firmware on several Intel boards during the period between 1985 and 1986. To make his job easier, he developed a set of table driven configuration routines that provide the basic core functions of interconnect. Being an avid (did I say rabid?) macro fan, Jory exploited every conceivable feature of the ASM-51 and RL51 development tools. The result is the firmware that you see today.

In the waning days of 1987, I was preparing for my Intel sabbatical when I first heard of Jory's work. I was so enthused by the possibility of a "universal" solution to the interconnect problem that Jory and I approached management about commissioning this project. John Hyde and Len Schulwitz obtained the necessary approvals and the writing began in earnest. Many Intel employees use their sabbatical time to write books, spend time with the family, or to travel to new places. I had already been selected to participate in a new "Intel China Ambassador" program and was about to travel all over the Asia Pacific region. Portions of this document were written in Alaska, Hawaii, Japan, and China — usually at an altitude of about 30,000 feet. The bulk was written in one marathon 10 hour layover in Tokyo's Narita airport. I wish to thank the many people throughout the world who lent me their personal computers, thereby enabling

this Application Note to get written. Also, thanks to the spec writers who gave us a reasonably clear view of what we were building, but left enough latitude for creativity.

Our foremost concern is for compatibility between vendor products. On February 22, 1988 this concept was put to the test. The occasion was the BUSCON Trade Show in Anaheim, California where 15 MULTIBUS® II vendors demonstrated their products — all operating in the same chassis! Both interconnect and message passing were proven to work between a wide selection of products. MULTIBUS II had achieved, in two years of production, what other busses have yet to accomplish.

As this firmware propagates its way into new vendor products and in-house designs, yet another generation of compatible products will be born. We hope you find it useful in your own designs.

Roger Finger
MULTIBUS® II
Technical Marketing Manager

Jory Radke
Modules Development Engineer

CHAPTER 1
INTERCONNECT ADDRESS SPACE ON MULTIBUS® II

1.1 WHO SHOULD READ THIS DOCUMENT?

The primary audience of this document consists of companies and individuals who are in the process of designing their own MULTIBUS II boards for use with other compatible products. It is assumed that the reader has already studied the Interconnect Interface Specification and has a good working knowledge in the operation of interconnect space on existing Intel products. In addition, portions of the IEEE 1296 specification and the MPC User's Guide are referenced in some detail. A complete bibliography of recommended reading material is contained in Appendix A.

This design package consists of two related items. The first item is the document you are now reading which is a user's guide to the overall design process. The second item, is a diskette containing copyrighted software to be used in generating new firmware for your interconnect subsystem. This package is not intended for users to change the content of microcontrollers already installed on Intel boards.

1.2 CONFIGURATION ISSUES IN MICROCOMPUTER BASED SYSTEMS

Over the past few years, microcomputer designs have progressed dramatically in capability and performance. In contrast, little progress has been made in enhancing ease-of-use. Until recently, board users have had to deal with the added complexity of modern single-board computers by wading through lengthy reference manuals and innumerable jumper options — often arriving at the final solution only by trial and error. System integrators often found that the firmware revision number penciled in on the EPROMs they installed did not match the device driver revision and consequently, nothing works! Worse yet, things might work for a little while and then fail; resulting in wasted time debugging the problem to determine what went wrong. Memory mapping options, arbitration priorities, interrupt levels, and scores of other "tunable" parameters contribute to the fray, leaving the system designer befuddled and confused. Often, the only way out of this mess was to locate a board that was already properly configured and copy off the jumper list.

Board designers build in numerous options so their products can be used in the broadest possible spectrum of applications. The number of options offered is not the core of the problem, but managing them is. MULTIBUS II addresses this problem with a special address space known as "interconnect". Now for the first time, system-wide configuration information has been made accessible to software; thereby opening opportunities for centralized control and coordination. In most cases, the end user of these products will be completely unaware of the configuration process. They simply remove the board from its shipping container, install the proper firmware, plug it into a free cardslot in the backplane, and apply power. Things work the first time around with no mess, no fuss, and no configuration errors.

Interconnect is great for end users; it eliminates most of the common configuration errors, speeds up the installation process, and facilitates diagnostics and repair. When considered in the context of an overall system architecture that includes message passing, interconnect is one of the foundation building blocks distinguishing MULTIBUS II as an environment capable of satisfying the most demanding of applications.

1.3 OVERVIEW OF INTERCONNECT ADDRESS SPACE

Interconnect address space is a fundamental part of the IEEE 1296 specification, which defines MULTIBUS II. Interconnect address space was included in the IEEE 1296 specification to solve three major problems: board identification, configuration, and diagnostics. The board identification registers are read-only locations containing board information such as type, manufacturer, compon-

ents installed, and other board specific functions. The configuration registers are read/write registers which allow the system software to set and change the configuration of many on-board hardware options. In most cases, hard-wired jumper options can now be eliminated in favor of software control. The diagnostic registers are used for the starting, stopping, and status reporting of self-contained diagnostic routines supplied with each board. These diagnostics are commonly known as Built-in Self Tests (BISTs).

1.3.1 Geographical Addressing

Interconnect is based on the fundamental principle that you can locate boards within a backplane using a system of cardslot numbering. This concept, known as geographical addressing, is a very useful tool during system-wide initialization. Each board in the system contains firmware which conforms to a standardized header format (figure 1-1).

At boot time, the system software will scan the backplane to locate its resources before loading in the device drivers. This approach eliminates the need for reconfiguring the software every time a new board is introduced into the backplane. It also solves the problem of how to configure multiple instances of controller and processor boards in large multiprocessing systems. Cardslot independence is achieved by having all boards in the system carry their own initialization and diagnostic functions on-board in firmware. Operating systems can generate a map of where resources are located during initialization and then use this map as a base address list for message passing.

1.3.2 Microcontrollers in the Bus Interface

Most MULTIBUS II designs are based on a highly integrated bus interface controller known as the Message Passing Coprocessor (MPC). Special provisions have been made in the bus interface silicon to enable board designers to implement intercon-

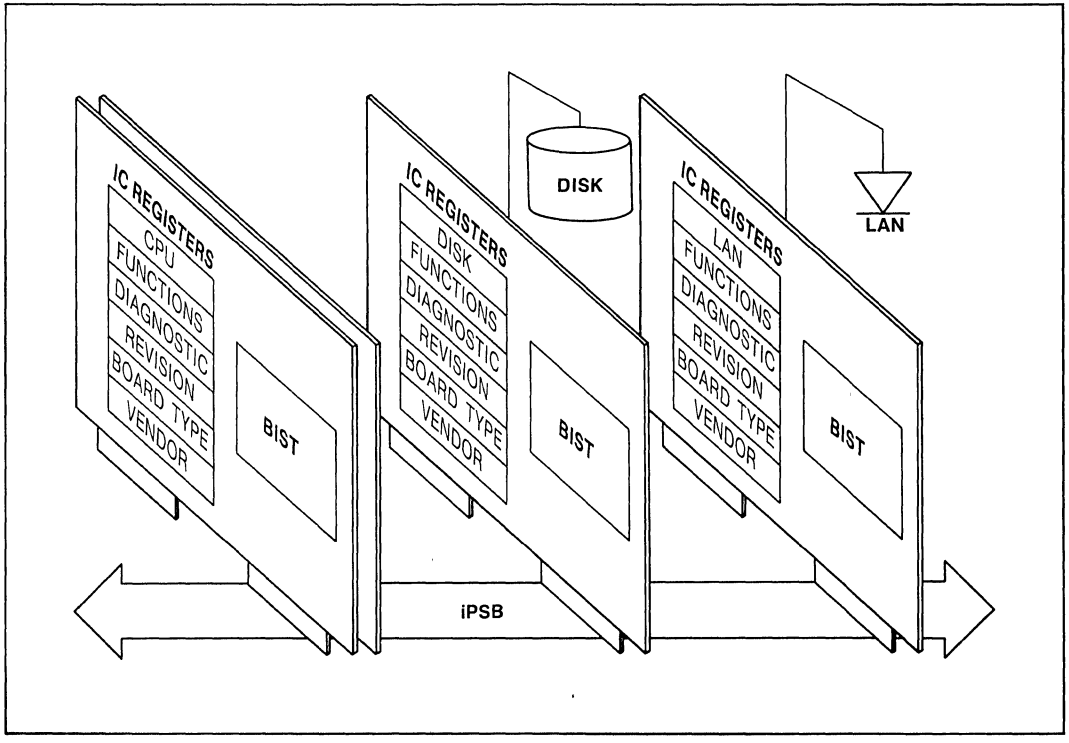


Figure 1-1. Interconnect Overview

nect in a cost effective manner. A typical MULTIBUS II interface consists of the MPC, a small number of bus transceivers, and a microcontroller such as the Intel 8751 or equivalent (figure 1-2). It is the microcontroller (in association with the MPC) that has the responsibility for all interconnect functions.

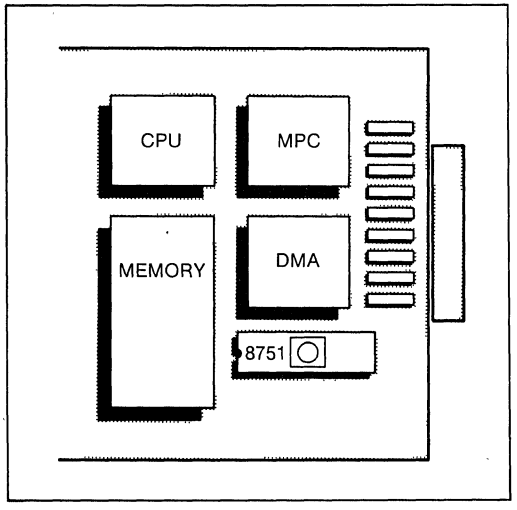


Figure 1-2. Hardware Support for Message Passing

Microcontrollers are ideally suited for this type of work because they are independent self-contained computing devices and require no external support chips outside of a clock crystal. Their architecture provides separate address spaces for on-chip ROM (4 kbytes) and RAM (128 locations), as well as three 8-bit bidirectional I/O ports. The ROM locations are used for program storage, constants, and read-only registers within the interconnect template. The RAM locations are used for read/write registers and as temporary storage. Port pins provide the interface to the real world; sampling test points, latching address terms into comparators, and controlling other devices on the board.

1.3.3 Addressing of Interconnect

Before discussing how to address interconnect registers on various boards, it is important to note that all interconnect implementations are dual-ported. Dual porting consists of an interface to the local CPU and to the Parallel System Bus (iPSB). Figure 1-3 shows that these two interfaces are addressed in slightly different ways. A complete interconnect address on the iPSB consists of a cardslot ID plus a register offset. These values are combined into a single 16-bit address field written to the iPSB by the MPC when an interconnect cycle is requested.

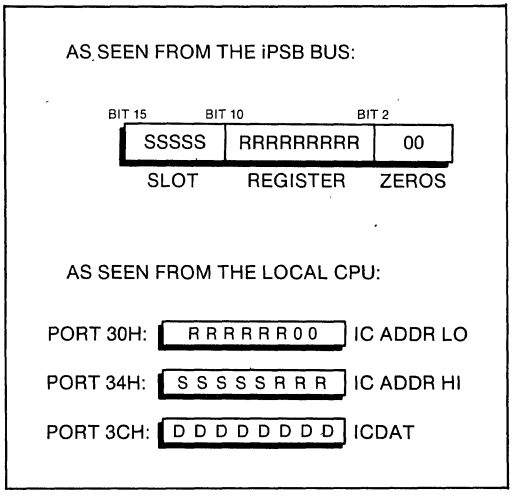


Figure 1-3. Interconnect Addressing

To generate an interconnect request, the local CPU writes the lower 8-bits of the interconnect address to a reserved I/O location (IC ADDR LO — usually 30H), and writes the upper 8-bits of the interconnect address to a second reserved location (IC ADDR HI — usually 34H). If it is an interconnect read operation, then the data can be read from a third reserved location (IDAT — usually 3CH). A write operation to the IDAT location will generate an interconnect request bus cycle on the iPSB.

One special case involves a CPU attempting to program its own on-board interconnect registers. As the CPU drives an interconnect address onto the bus, its transceivers wait for a handshake from the replier board. But since an CPU cannot handshake with itself, such a transaction would be invalid and an error generated. Whenever a CPU is programming its own interconnect registers, a cardslot address of 31(1FH) should be used. This instructs the MPC to pass the request directly to the local microcontroller without going through the iPSB interface.

Another special case is when sub-buses such as the Local Bus Extension (iLBX™), are attached to the primary agent. Interconnect facilities should be provided for these boards and the addressing on the sub-bus begins with cardslot number 24 (i.e. the primary agent), and proceed upwards to cardslot number 30.

1.3.4 Data Structures in Interconnect

The objective of interconnect address space is to allow higher level software to gain information about the environment in which they operate independent of who manufactured the board, what functions it contains, and what cardslot it resides in. To accomplish this goal, an Interconnect Interface Specification has been published and forms the basis for much of the information in this guide. If you have not yet read this document, you should do so before beginning your design effort.

Interconnect functions implemented on Intel's single board computers go beyond the requirements of the IEEE 1296 specification. This specification mandates that all conforming products include an Interconnect Header Record. The header record consists of information regarding board type, its manufacturer, what firmware is installed, and other relevant information. An example header record is shown in figure 1-4. In addition to the header record, the manufacturer may also supply additional function records which make other features of the board accessible through interconnect.

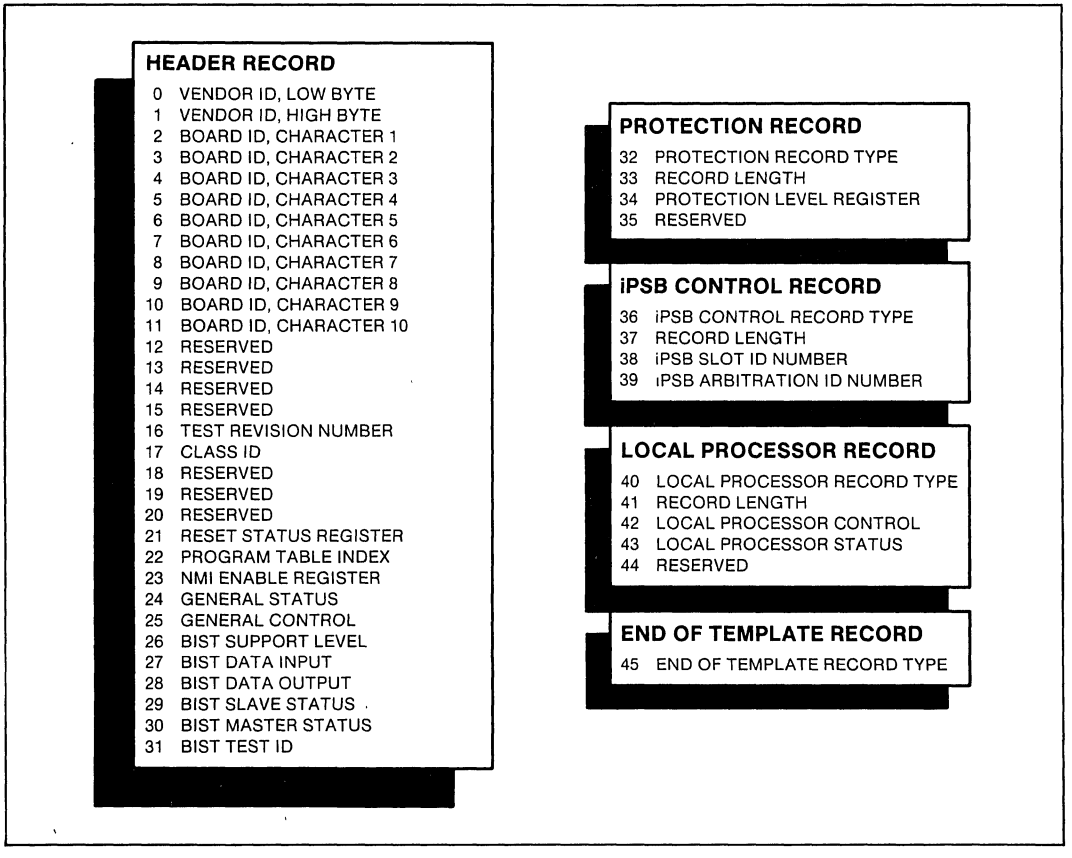


Figure 1-4. Function Records in the Interconnect Core Firmware

Function records begin with a byte specifying the record type followed by the number of bytes that make up the record. The manufacturer must publish a description of these function records in their reference documentation. Many types of function records have already been defined. Some examples include memory configuration, parity control, serial I/O, and other commonly used functions. If the existing record types do not adequately describe a function, a new record type can be defined — up to a maximum of 1020 different record types are allowed.

The system software initiates the search for a spespecific function record at interconnect register 32, which is the first function record following the header record. The program first checks the record type field and then counts bytes to the next record type field until either the correct function record is found or the End of Template (EOT) record (hex value 0FF) is encountered.

1.3.5 Access Rights and Protection Records

Because the interconnect registers are dual-ported, each has a set of static and dynamic access rights that determine which operations will be allowed on either the local or iPSB bus interface. All interconnect registers can be read from either bus, however, static access rights may place restrictions on whether a register can be written to from either interface. The term "static" is used because these access privileges are predetermined by the designer of the interconnect firmware and will not change during system operation. In other situations it may be desirable to allow a register to be modified during system initialization; then locked against further changes during normal system operation. This capability is essential since many of the functions contained in interconnect are so vital to correct system operation that some means of protecting them from malicious or inexperienced users is required.

Dynamic access rights are determined by protection records which are used to prevent other boards from modifying a local interconnect resource. When activated, all subsequent records become read only so other users can read from interconnect registers, but cannot write to them.

1.3.6 Diagnostic Philosophy of MULTIBUS® II

The diagnostic philosophy of MULTIBUS II is that each board should have the capability to test itself and report error status when problems exist. There are two occasions when diagnostic testing is invoked. A subset of the complete on-board diagnostics is run during power-on initialization and more extensive testing can be invoked from the operator's console. Following power-on, most boards go through a series of initialization checks where the basic functioning of the MPC and microcontroller are verified. Initialization is followed by a power-on test suite automatically invoked by each board. If a hardware failure is detected at this point, a yellow LED on the front panel will illuminate so that the failing module can be easily identified and replaced.

If further testing is desired, extended diagnostics can be invoked by placing a diagnostic request packet in the interconnect BIST registers. Usually one board acts as the Master Test Handler and requests services from other system boards functioning as Slaves when under test. A menu of tests is available via interconnect. This test philosophy can be applied on-site by the end-user, service representative, or remotely executed via modem from the regional repair center. In most cases, downtime is minimized by sending out a replacement board and thus avoiding an expensive repair call.

The firmware content of MULTIBUS II boards is much greater than that found on previous industry standard buses. In addition to the 8751 Microcontroller, MULTIBUS II boards normally host EPROMs that contain extended diagnostics (BISTs), test handlers, reset initialization sequencing, debug monitors, and many other functions. The location of diagnostic firmware on a board (figure 1-5) is dependent on code complexity and execution speed. For simple replier agents, the microcontroller's on-board EPROM may have enough program storage space for diagnostic functions as well as the interconnect firmware. In contrast, the majority of the requestor/replier boards (i.e. capable of becoming bus masters), are more complex and most diagnostic code is run on the CPU from on-board EPROM. In this case, the microcontroller primarily serves as the communication interface for the diagnostics.

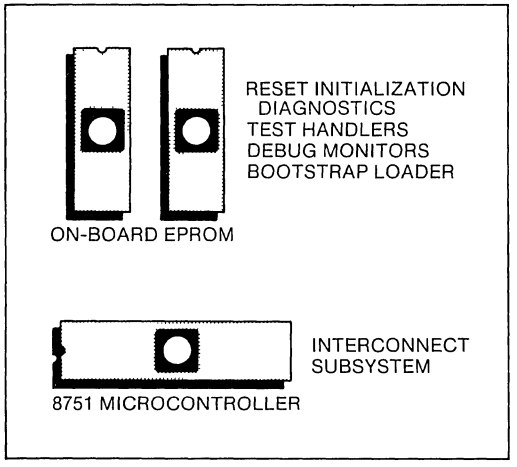


Figure 1-5. Firmware Content of MULTIBUS® II BOARDS

1.4 INTERCONNECT — THE MANUFACTURER'S PERSPECTIVE

From the perspective of a board designer, interconnect is a mixed blessing. The board manufacturer is certain to enjoy the benefits of reduced support costs, easier fault isolation in field repairs, and enhanced customer satisfaction — but these advantages do not come free. One would anticipate longer development time, increased on-board part count, and the firmware configuration to increase the amount of effort it takes to prepare a MULTIBUS II board for market. And indeed this is so. If a competent design team were to tackle the interconnect subsystem (including diagnostics, hardware, and firmware design), it would not be unreasonable to allow six man-months for the job.

Given that this represents an extraordinary investment for the manufacturer, the primary goal of this design package is to reduce the amount of time required to include interconnect in your design from six months to only six days! Another goal is to guarantee compatibility and interoperability of your products by placing common core functions in user extensible firmware in such a way that it is easy to customize the design to fit your own particular needs.

1.5 ASSUMPTIONS REGARDING YOUR OPERATING ENVIRONMENT

Although many implementations of interconnect are possible, it was necessary to restrict the scope of this guide to satisfying the broadest and more typical range of users — designs based on the MPC component in association with an 8751 Microcontroller (or equivalent). This design guide is intended as a generic solution that meets the needs of most of these users.

For software development, it will be necessary for you to obtain an IBM® PC (model XT, AT, or compatible) plus the appropriate Intel programming languages and a PROM programmer to transfer your code into the microcontroller. An in-circuit emulator is not required for this project; however, some users will find it expedient to make use of such a tool since it simplifies debugging and eliminates the need for PROM programming while the code is being developed. A complete list of hardware and software requirements may be found in Chapter 2. Any departure from the recommended development tools or practices is outside the scope of this document and may lead to unpredictable results.

1.6 DESIGN METHODOLOGY — AN OVERVIEW

The process of designing an interconnect subsystem invariably begins with a high level discussion of what function you intend to support. While the header record is quite easily defined, decisions as to what function records to include should be carefully considered in terms of how much flexibility to give your users, what functions they might be interested in, and how much external hardware will be required. Some of the function records listed in the Interconnect Interface Specification are already implemented in the core firmware and require minimal effort to support. Other functions may be quite complex and could potentially require extensive TTL circuitry external to the microcontroller. As with all engineering designs, you should spend a significant portion of your time making sure that you have a clean workable specification before proceeding into the implementation phase.

The second step in interconnect design is to determine what circuitry is required to gain access to the information in interconnect that you intend to

present to the user. This consists of locating all test points, control circuitry, latches, and transceivers external to the microcontroller. Most likely this determination will be made at a point where the overall schematic for the board is near completion and before you begin the layout and develop the prototype. At this stage, all dedicated I/O addresses will be defined and rudimentary PAL equations for the control points will be written.

The third step of interconnect design consists of evaluating your on-chip resource requirements based on the function records being implemented. At this stage, you will write the functional routines and identify the RAM, ROM, and port requirements for the microcontroller. If your original interconnect specification was over ambitious or inappropriately defined, you will discover at this point that you may be forced into external PROM, static RAM, or port expansion logic; and may wish to scale back your design or change over to the 8752 Microcontroller (having 8 kbytes of ROM and 256 bytes of on-chip RAM). Once you know your resource requirements and have written the functional routines, you are now ready to integrate your custom code with the core interconnect firmware.

The fourth step in the design process consists of loading the tables with data based on the interconnect template you specified in step one, plus the external declarations for the routines you wrote in step three. The object code supplied on the diskette with this guide contains a table driven collection of routines that provide the core interconnect functions. These give the user some commonly used function records (figure 1-4) and provide the opportunity for users to add their own routines to this core. Generating the firmware consists of assembling your code and then allowing the table generators to integrate this code into the core module through an ASM-51 macro expansion process. Once complete, the entire package is integrated using RL51 (a relocation linker) to resolve any external references and produce a unified object module for loading into the microcontroller EPROM.

The final step in the process is to program the microcontroller and test every imaginable function and event sequence within interconnect. It is at this stage that the use of an in-circuit emulator, logic analyzer, or oscilloscope may be desirable to help in localizing logic faults or timing related problems. In most cases, debug time is fairly short since the core routines are supplied already and are known to be good.

Once the interconnect subsystem is totally tested, the board can be forwarded to the device driver development team and/or system integrator for initialization software development and further functional testing.

CHAPTER 2
PREPARATION FOR USE

2.1 HARDWARE AND SOFTWARE REQUIREMENTS

Before proceeding with your interconnect design using this guide, the hardware, software, and optional equipment listed below are required (at a minimum).

Hardware Requirements:

- IBM® PC (model XT, AT, or compatible) configured with at least 640 kbytes of internal memory and a 10-Mbyte (or larger) hard disk.
- Intel PROM Programmer, model iUP 201, plus the 8751 Microcontroller Personality Module and a serial cable. The IBM PC must have a spare serial port to interface with the PROM Programmer. (Note that other brands of PROM Programmers can be used, but the batch files and object module produced by Intel's development tools are not guaranteed to be compatible.)

Software Requirements:

- DOS Operating System, version 3.0 or greater
- iPPS PROM Programming Software, version 2.2 or greater
- ASM-51 Macro Assembler version 2.2 or greater
- RL51 Relocation/Linkage package version 3.0 or greater.

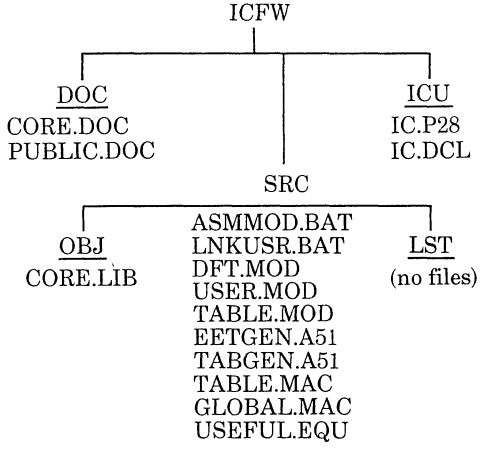
Optional Equipment:

- ICE™ 51 (or ICE 5100) In-Circuit Emulator with IBM PC Interface Card
- Oscilloscope.

2.2 LOADING SOFTWARE ONTO YOUR SYSTEM

Before starting, it is assumed that you will have already loaded the ASM-51 and RL51 program files in a common subdirectory of the hard disk and have indicated the route to that subdirectory in a PATH command entered into the AUTOEXEC.BAT file (in the root directory). This will allow you to call these program files from any point in the file structure without having to identify the directory search path to these files.

Install the software supplied with this design guide onto your hard disk by inserting the interconnect firmware diskette into drive A or B. After the C> prompt, type A (or B): to change the default drive. Next type INSTALL and then press **Enter**. The computer will read from the drive specified and immediately start executing the install.bat batch program. When this batch program completes processing, you will find the following added to the subdirectory structure of your hard disk:



Batch files are supplied to automate the firmware generation process. These files make some assumptions about your hard disk directory structure. It is

important to note that all batch files must be called from the \ICFW\SRC directory path, otherwise the DOS command processor will not look beyond your current directory in its search for a file name. Edit the path command in your AUTOEXEC.BAT file to include the \ICFW directory.

2.3 INTERCONNECT THE EASY WAY

While interconnect is a complex topic, there is an easy way to get a functional interconnect subsystem operational without detailed knowledge of the internal design. To do this, view the file \ICFW\SRC\TABGEN.A51 using the TYPE command or a text editor and notice that the data fields for vendor ID, board ID, hardware test revision, and class ID have been left blank (looking ahead, this is figure 5-1). Consult the Interconnect Architectural Specification to determine what information to place in these fields. Once you have obtained this information, perform these steps at your computer console:

a. Type CD and then press **Enter** to display the current directory. If \ICFW\SRC is not being displayed, then type CD\ICFW\SRC and press **Enter** to change to the correct directory path.

b. Run the following batch programs in the order listed:

```
ASMMOD  dft
ASMMOD  user
ASMMOD  table
LNKUSR  test
```

The result is a PROMmable object code file (TEST.LNK) that is placed in \ICFW\SRC\OBJ subdirectory. The TEST.LNK object code is ready to burn into the microcontroller EPROM and provides a complete interconnect header record as well as protection, iPSB control, and local processor records.

The core hardware design consists of the minimum interconnect implementation as shown in figure 4-1. This basic combination of hardware and firmware can be used during prototyping as the starting point for most interconnect designs.

CHAPTER 3
THEORY OF OPERATION

3.1 MPC TO MICROCONTROLLER HARDWARE INTERFACE

Most MULTIBUS® II designs use the Message Passing Coprocessor (MPC) component with an 8751 Microcontroller to implement the Parallel System Bus (iPSB) interface. This combination minimizes the number of devices required to implement a full-featured bus interface and provides flexibility in adapting the design to the broadest possible range of functional specifications. The hardware interface between the MPC and the microcontroller is shown in figure 3-1.

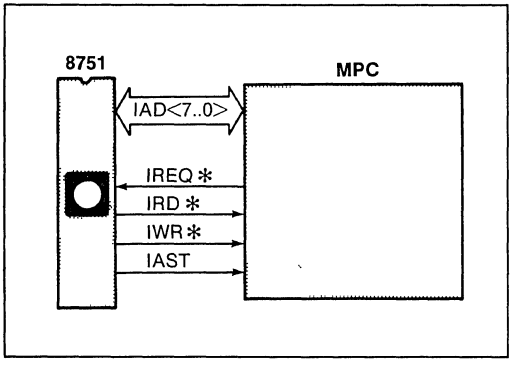


Figure 3-1. MPC to Interconnect Pathway

The MPC is designed to sit directly on the microcontroller's multiplexed Interconnect Address/Data bus (IAD<7..0>). When an interconnect cycle is initiated, the **IREQ*** signal from the MPC interrupts the microcontroller with a request for services. The microcontroller then performs a series of read and write operations to a group of MPC interconnect bus registers to satisfy the interconnect request and complete the operation. In terms of hardware control, the microcontroller acts as the bus master on this interface; generating the read and write signals, and supplying an Interconnect Address Strobe (**IAST**) based on its own Address Latch Enable (**ALE**) signal. The reader is requested to review chapters 4 and 5 of the MPC User's Manual before continuing further.

3.2 MPC INTERCONNECT BUS REGISTERS

The MPC component contains a set of special function registers that are only accessible via the IAD bus. These registers (figure 3-2) can be catagorized into five functional groups: interconnect reference registers, slot and arbitration ID registers, configuration registers, diagnostic registers, and the no access registers.

The MPC interconnect reference registers serve as the basic communications interface between the microcontroller and the MPC. Whenever the local CPU or iPSB agent generates an interconnect request cycle, the registers actually being accessed (IC ADDR HI, IC ADDR LO, IDAT) physically reside in the MPC rather than in the microcontroller. The MPC asserts the **IREQ*** signal to interrupt the microcontroller which responds by initiating a dialogue of read/write commands to the MPC interconnect reference registers.

The second functional group of MPC interconnect registers are concerned with the cardslot and arbitration ID assignments made by the Central Services Module (CSM) during reset initialization. Note that the Interconnect Interface Specification describes an iPSB Control Record which includes registers for both arbitration and cardslot ID. This allows a CPU to determine in what cardslot it is residing.

The third functional group of MPC interconnect registers control configurable features on the MPC such as dual-port address boundaries, arbitration priority, Reset-Not-Complete (RSTNC) control, error reporting, and fail-safe counter functions. In most implementations, these registers are passed through an interconnect function record to make them user accessible and configurable.

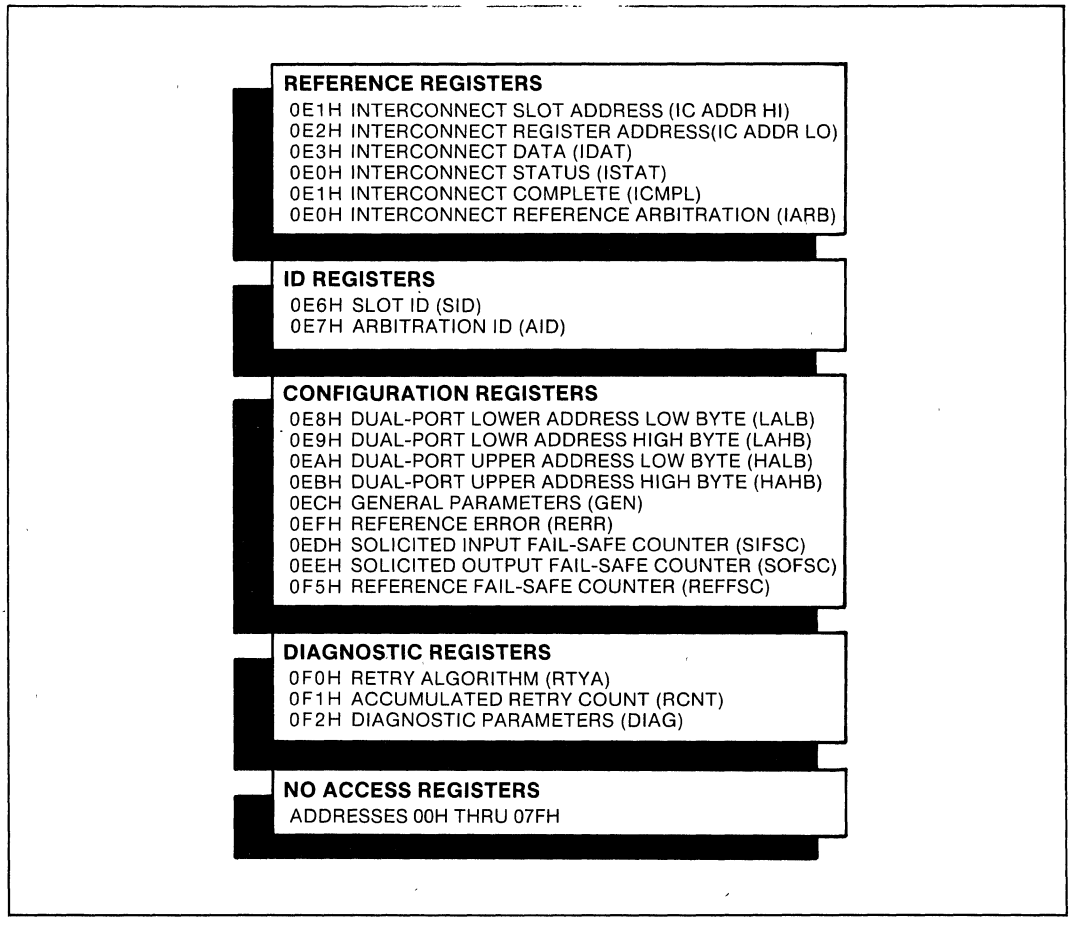


Figure 3-2. MPC Interconnect Registers

The fourth functional group of MPC interconnect registers control retry and diagnostic functions. These parameters can be used for performance tuning and confidence testing, but are not frequently accessed by users. In general, they can be programmed during initialization with default values and then ignored.

Finally, there is a group of addresses between 0 and 7FH for which the MPC guarantees a tristate condition (no access). Interconnect designers can take advantage of this feature by using these addresses to decode registers and latches residing directly on the IAD bus. This technique provides a convenient way to generate an 8-bit bidirectional bus using user defined addresses as chip enable terms. The reader is encouraged to read all of Chapter 7 in the MPC User's Guide before proceeding.

3.3 PARTITIONING OF FUNCTIONS BETWEEN THE MICROCONTROLLER, CPU, AND MPC

Because of the close association of the CPU and the microcontroller to the MPC, a number of interdependencies arise — especially during initialization and diagnostic testing. Some of the more complex functions such as dual-port memory control and fail-safe counters actually cross com-

ponent boundaries. Thus, one must view the complete interface as a functional subsystem. The following discussion highlights the more important interdependencies regarding the interconnect subsystem.

3.3.1 MPC Diagnostic Testing

A special feature of the MPC is its ability to simulate message passing operations to itself without presenting data to the iPSB bus interface. This mode is termed "MPC Loopback Testing" and is enabled by the iPSB Diagnostic Register in the iPSB Control Record. This causes the microcontroller to set the Reset-Not-Complete Out (RSTNCOUT) bit in the MPC General Parameter Register. The result is that any messages the local CPU loads into the MPC transmit buffers are routed directly to the MPC receive buffers. While this is happening, the MPC Buffered Address/Data bus (BAD<31..0>*) is active and it is necessary for the microcontroller to tristate the iPSB buffer logic by preventing the iPSB Transceiver Output Enable (**BTROE***) signal from going active low. Note that MPC loopback testing is only allowed while **RSTNC*** is being asserted by the host agent.

3.3.2 Dual-Port Memory Control

When another iPSB agent selects your board to participate as a replier in a memory reference, the address recognition function for dual-port memory is performed by the MPC based on the starting and ending addresses programmed into the MPC interconnect registers during initialization. When an address match is found, the MPC will drive the **SEL*** signal to your dual-port memory controller and wait for a **COM*** or **ERR*** signal to be returned before completing the cycle. During this transaction, the MPC provides all parity generation and checking, system control, and wait-state signal generation services to the iPSB bus interface. Valid address selection may occur on any 64-kbyte boundary within the 4-Gbyte memory space; however, it is advisable to include value checking in your microcontroller firmware to ensure that the user doesn't enable more memory than is physically present on the board. Only one bank of contiguous memory is supported when using the MPC dual-port functions.

3.3.3 Message Retry Operations

When message traffic is arriving faster than the local CPU can receive it, the MPC FIFO buffers will overflow and some form of flow control must be initiated. The MPC has a special retry mechanism for this condition that is controlled through registers accessible to the microcontroller. Retry is enabled by setting a bit in the MPC Diagnostic Parameters Register and the delay between retries is selected in the Retry Algorithm Register. The Accumulated Retry Count Register indicates how many Negative Acknowledge (NACK) errors have occurred for a given message attempt. The Accumulated Retry Count Register is used with the Retry Algorithm Register to tune system performance by selecting the most effective retry interval.

3.3.4 Fail-Safe Counter Functions

When the MPC issues an iPSB buffer request, there is no guarantee that a buffer grant will be returned in a reasonable amount of time. If enabled, the MPC Reference Fail-Safe Counter will cause an error interrupt if no buffer grant is received by the end of a timeout period (typically 1.5 seconds). Likewise, a similar fail-safe timeout exists for reference operations in the unlikely event that they are unable to acquire the iPSB bus due to arbitration or Bus Clock (BCLK) problems. In either case, the microcontroller acts as a programmable timebase by writing to the MPC fail-safe counter addresses on a periodic basis in response to an internal timer interrupt (figure 3-3). When the MPC starts the buffer request or reference operation, it enables the fail-safe timeout and waits for the operation to complete. If the microcontroller is able to write to a MPC fail-safe counter address four times before the bus cycle completes; then a timeout interrupt is asserted to alert the CPU to the problem. Note that fail-safe counter functions should be disabled during debugging since breakpoints set by human intervention may prevent the MPC from completing an operation before a timeout occurs.

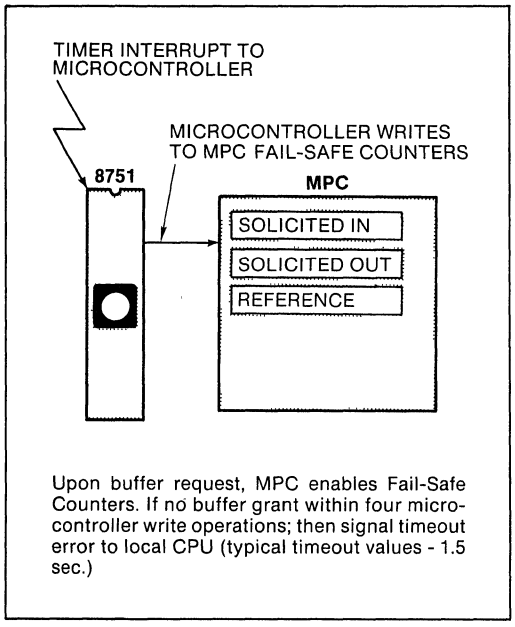


Figure 3-3. MPC Fail-Safe Counters

3.4 MULTIBUS® II RESET CONDITIONS

In a MULTIBUS II backplane, the **RST*** signal is used for the system-wide reset. Additionally, the **DCLOW*** signal designates power-fail indication and the **PROT*** signal designates an early warning battery back-up control. One of the options available to the board designer is to use combination logic in association with the microcontroller to further define three catagories of reset conditions: cold-start, warm-start, and local reset. In figure 3-4, the reset circuitry that gives a board the capability to distinguish between these events is shown. The reset control logic (in the PAL16R4B) signals a cold-start whenever reset is accompanied by a low power condition and signals a warm-start in all other cases. Local resets are generated by an interconnect operation to the microcontroller, which then pulses the CPU reset line.

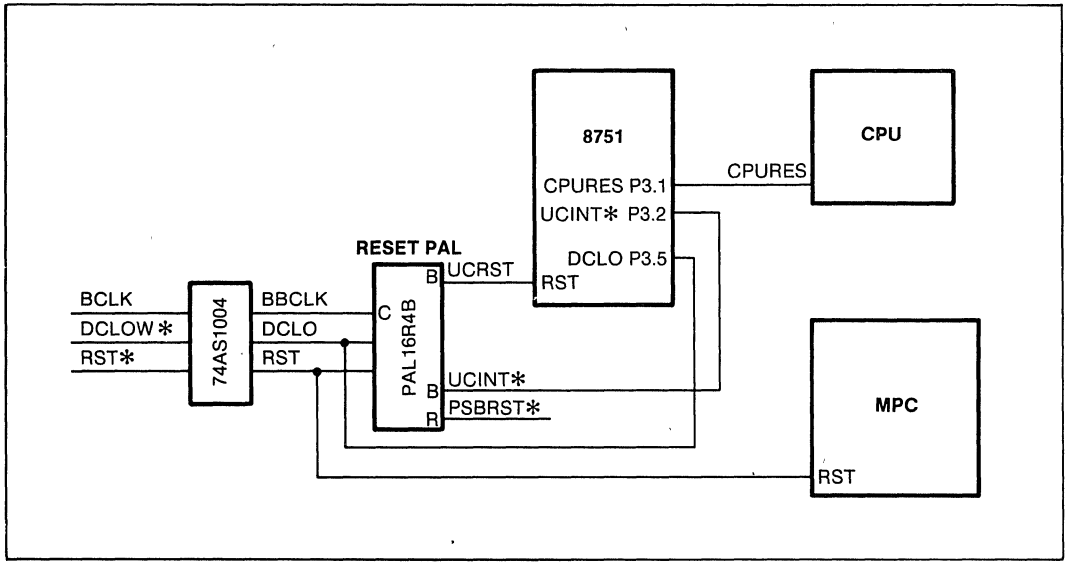


Figure 3-4. Reset Hardware

The equation for the RESET PAL in figure 3-4 is:

```
RSTCTL PAL
INTeL CORP
VERSION 001
chip name PAL16R4B

BCLK RST DCLO 4 5 6 7 8 9 GND
/OE 12 13 14 15 16 /PSBRST /UCINT UCRST VCC

EQUATIONS

PSBRST = RST
UCINT = PSBRST * /DCLO * /UCRST
/UCRST = /PSBRST + (/DCLO * /UCRST)
```

3.4.1 Cold-Start and Recovery Reset

A cold-start condition represents the state of the backplane following a power-on sequence, while the recovery reset represents the state of the backplane following power failure situations. The CSM will cycle through its normal sequence of assigning cardslot and arbitration IDs during a 50 ms period while the **RST*** signal is active low. The iPSB reset signal is buffered in on each agent and then routed directly to the MPC and the microcontroller. When reset is released from the iPSB, the microcontroller will force a reset to the local CPU and then execute some rudimentary BISTs (initialization checks). The interconnect template will be loaded with default values and the MPC gets initialized to its default state. The CPU is then released from reset to execute its power-on BISTs (some of which are likely to alter the contents of memory by overwriting data previously present).

In the firmware supplied with this guide, the CPU must clear the RSTNC* bit in interconnect within 30 seconds after power-on or else the microcontroller will again assert reset to the CPU on the assumption that something is seriously wrong with the board. (This is done to ensure that the failure of a single system board will not prevent the remainder of the system from coming alive due to **RSTNC*** being held active low.)

3.4.2 Warm-Start Reset

A warm-start condition is normally the result of a human operator pressing a front panel reset button or programming the CSM for a system-wide reset via interconnect. The sequence of events is similar to the cold-start except that memory will remain unchanged, BISTs are not run, and all interconnect configuration registers remain unchanged. Warm resets are very useful for recovering from a hung system condition during debugging because the system cycles through reset much more quickly and retains all configuration and error status information.

3.4.3 Local Reset

Local resets selectively return all circuitry behind the bus interface on a single agent to a known condition without effecting any other system boards. Local reset is generated through interconnect programming and does not disturb the contents of local memory or the interconnect configuration registers. The microcontroller BIST diagnostics are not run and the agent returns quickly to a known condition. Local resets are often used to recover from a partial system failure condition caused by improper programming or single agent failures.

There are two types of local reset supported: self-toggling and non-toggling. The Interconnect Interface Specification describes a local reset control (bit 7) of the General Control Register in the Interconnect Header Record of each board. This is a non-toggling reset that holds the agent inactive until specifically cleared by further interconnect programming. This feature allows an external agent to clamp reset to that board, thereby taking it off-line until further notice. Note that the board will not respond to other resets (warm or local) until this bit is cleared. This type of local reset is not suitable if the intent is for the agent to reset itself since the board is incapable of returning from the reset condition to clear the bit.

For this reason, many MULTIBUS II boards offer a self-toggling reset control in the Board Specific Record that will clear itself soon after being set (typically 20 ms). One might be tempted to use this register for all local resets in preference to the General Control Register, however, it is not supported on all MULTIBUS II boards and its absolute location in the interconnect template will depend on how many other function records precede it. In common practice, the local reset bit in the General Control Register is a better solution in all cases except self-toggling resets.

3.5 INITIALIZATION OF THE MICROCONTROLLER FOLLOWING RESET

Immediately following a cold or recovery reset, all bus interface components must be initialized to a

known state. The microcontroller is normally the first to undergo initialization since it will be released from reset before the CPU and has the added responsibility for the MPC interconnect registers. Flowcharts of the program flow in the microcontroller are provided in Appendix B.

The first priority of the microcontroller is to establish its own environment by creating a stack and loading the Special Function Registers (SFR's), which control the on-chip resources such as the ports, timers, and interrupt priorities. Next, the microcontroller initialization checks are run as a basic "sanity check" of itself. There are three BISTs in the initialization sequence:

a. Microcontroller RAM — A simple read/modify/write test to on-chip RAM locations.

b. Microcontroller ROM — A check sum test on the on-chip ROM.

c. IAD Bus and MPC Accessibility — A "walking ones" pattern is written to several MPC registers and read back to verify functionality of the MPC interconnect registers and the IAD bus.

The interconnect registers that reside in RAM (i.e., those that are R/W), are now loaded with their default values. These default values are copied from a ROM-based default table (the DFTABLE), specified at configuration time. Next, the microcontroller on-chip timer/counters and the MPC interconnect registers are initialized. If the user has any unique initialization requirements, a call to the INIT_USER procedure is made at this time. Up to this point, the local CPU has been held in reset and the iPSB transceivers are being held tri-state. Now all of these resources are released and timer/counters startup and the CPU has 30 seconds to clear the RSTNC* bit in interconnect. The microcontroller then enters the mainline routine of the firmware.

3.6 INTERRUPT DRIVEN FIRMWARE OPERATIONS

Figure 3-5 illustrates the various interrupt sources which drive the interconnect functions on the microcontroller. The mainline code consists of very simple polling operations that execute with interrupts disabled. These operations consist of polling for the Non-Maskable Interrupt (NMI), BIST, and reset conditions, plus updating the front panel LEDs. At the end of each pass through the mainline code, interrupts are enabled prior to executing the jump instruction that starts the next pass through the mainline code.

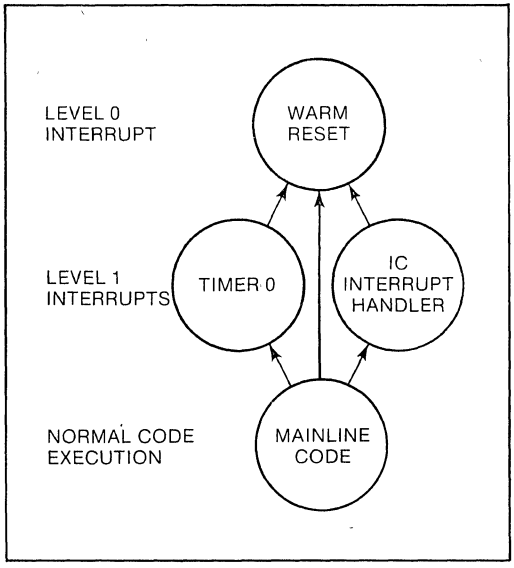


Figure 3-5. Microcontroller Interrupt Sources

Nearly all of the important operations within interconnect are interrupt driven. These include the IC interrupt handler (which services the MPC **IREQ*** signal), a reset interrupt handler, and a group of timer based functions. A general description of these operations follow.

3.6.1 Interconnect Interrupt Handler

The IC interrupt handler is called whenever the MPC signals to the microcontroller that a request for interconnect service is pending (**IREQ*** has been asserted). The microcontroller responds by reading the MPC's interconnect reference registers to determine which register in interconnect has been requested; whether this is a read or write operation; and whether this is the local CPU or an iPSB agent generating the request. If the register number is beyond the End of Template (EOT) record; then no error status is posted and the

interconnect operation is terminated. Otherwise the microcontroller begins a table driven look up of that register and completes the interconnect cycle.

3.6.2 Interconnect Read Operations

Interconnect read operations are relatively simple because no need exists for checking the access rights or data values (see figure 3-6). The register number (REGNUM) is used as an 8-bit offset into the read vector table (RVTABLE), which then supplies an offset into the read jump table (RJTABLE) containing a long jump instruction to the address of the read routine. This indirect look up technique is used to allow several different registers to share the same read routine and to improve code compaction. Indirect jump tables also allow the user to add their own read routines to the template following the interconnect core functions.

Once the read operation is complete, the results are available to the IC interrupt handler in the ICDATA register. The data is copied to the MPC IDAT register and completion status is posted. From the IC interrupt handler, control passes back into the mainline code. Note that interrupts were enabled within the IC interrupt handler (just prior to executing the long jump instruction), and that the reset interrupt has a higher priority.

3.6.3 Interconnect Write Operations

When an interconnect write operation is requested, the firmware checks whether the requestor has permission to write to that register. The IC interrupt handler examines the static and dynamic access rights of the register. If access permission is denied, then the appropriate error status is posted in the General Status Register of the Interconnect Header Record.

In addition to checking access rights, some registers will need value checking as well to determine if the data being written is within legal range for that function. For example, it would be advisable to subtract the memory starting address from the ending address to determine if memory size limitations are being exceeded. Other registers contain bit mapped functions and you should signal a value error if a Reserved for Future Use (RFU) bit was set.

Figure 3-7 shows the table look up technique used during interconnect write operations. In this implementation, value checking is performed by a collection of edit routines. The mechanism for selecting an edit routine is similar to the indirect jump method previously described for read routines. The REGNUM serves as an 8-bit offset into the edit vector table (EVTABLE). This table sup-

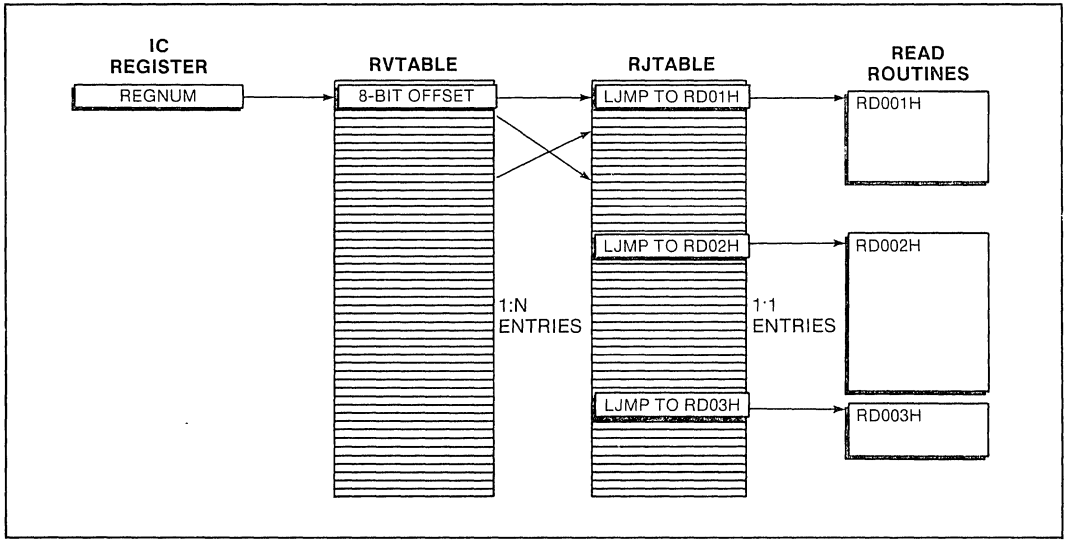


Figure 3-6. Indirect Jump Tables for Read Routines

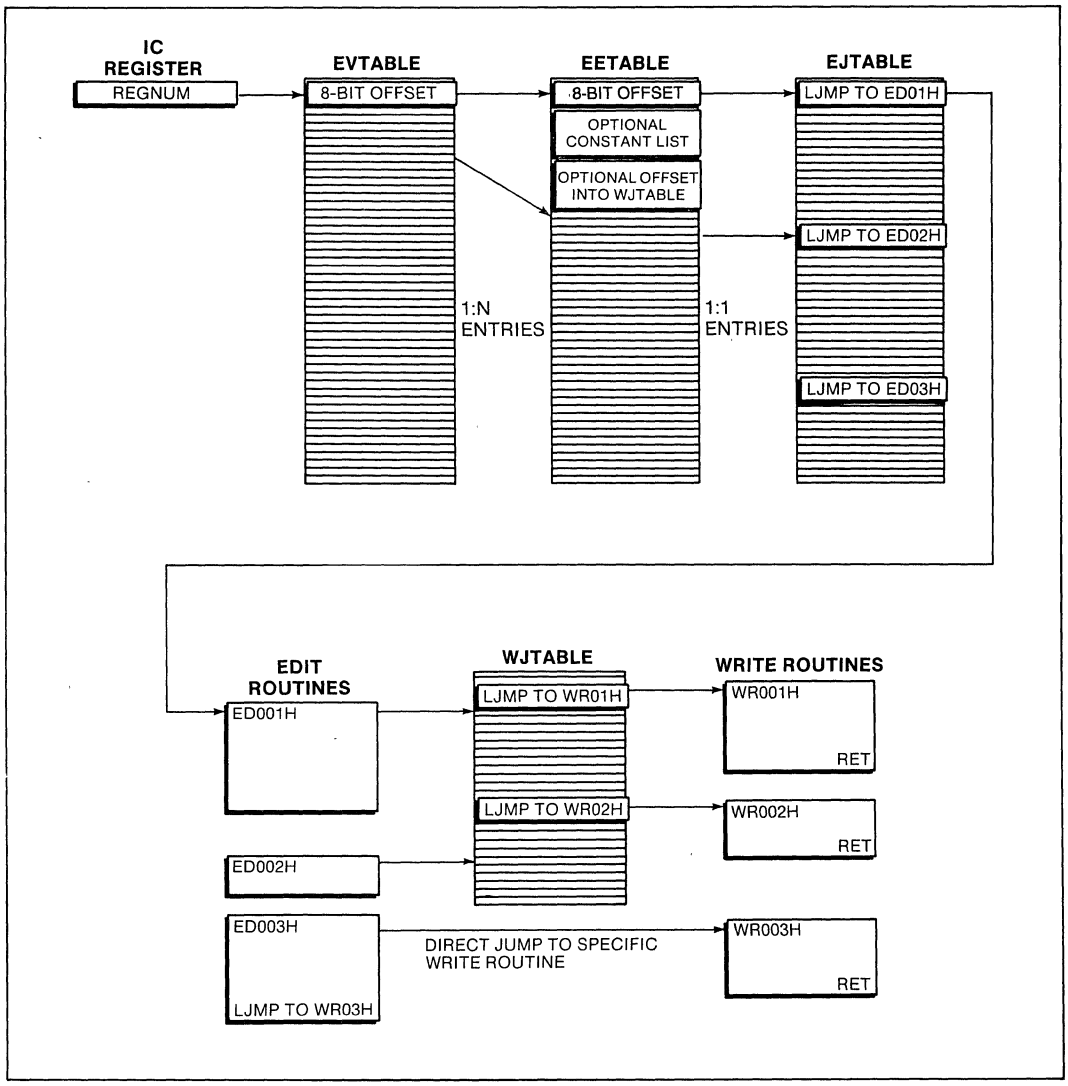


Figure 3-7. Indirect Jump Tables for Edit and Write Routines

plies another 8-bit value called EET_OFF, which is the offset of a data structure from the base of the edit entry table (EETABLE). This data structure contains an index into the edit jump table (EJTABLE), plus an optional list of constants to be used in value checking. If no constants are supplied, the edit routine will perform value checking using its own constants and no parameter passing is required. This has the effect of making the edit routine specific to a given function as opposed to generic routines, which receive their parameters by looking them up in table EETABLE.

Table EETABLE has another optional field for the user to supply an offset into the write jump table

(WJTABLE). If omitted, the edit routine will jump directly to the write routine without accessing table WJTABLE. Use of table WJTABLE has the advantage of being able to use various combinations of generic edit and write routines for better efficiency and code compaction.

The indirect jump method of read, write, and edit services requires slightly more code space and software overhead than direct calling of register service routines, but offers a simple and extensible mechanism for users to form their own interconnect templates without needing to know the details contained within the core interconnect module. Instructions on how to load these tables and generate the final object code are presented in Chapter 5.

3.6.4 Reset Interrupt Handler

In contrast to the power-on reset (which is a cold start sequence), the reset interrupt is used exclusively for warm resets generated when **UCINT*** goes active. The reset interrupt handler asserts the reset signal to the CPU; then disables NMI to protect itself from losing program flow (yes — the non-maskable interrupt is indeed maskable on the 8751 Microcontroller!). The reset interrupt handler then reinitializes the microcontroller and clears the BIST registers. It next initializes the MPC and provides a call to the USER RESET routine (just in case you want to alter the default values in the MPC registers). Remember that any values previously in the interconnect registers will remain unchanged throughout a warm reset sequence.

3.6.5 Timer Based Functions

There are a number of functions within interconnect that need to be performed periodically. These include writing to the MPC fail-safe counters, checking the RSTNC* counter, and optionally, user-defined timer based functions. The elapsed time between timer/counter interrupts is 1 ms based on use of an 11-MHz crystal oscillator for the microcontroller. This results in a fail-safe timeout period of 1.5 seconds. It is strongly recommended that you use this clock frequency and do not attempt to change the period of the timer interrupt because of the impact on the fail-safe count and RSTNC timing. Other unforeseen effects on the interconnect operations may also occur.

In the microcontroller, timer 0 is an internal interrupt source and has a higher hardware-implemented priority than does the IC interrupt handler even though they both are operationally level 1 interrupts. For this reason, the timer 0 interrupt is masked while the IC interrupt handler is executing.

3.7 USE OF MULTIBUS® II BOARDS WITH IN-CIRCUIT EMULATORS

In theory, in-circuit emulation of MULTIBUS II boards is no different than working with any other CPU based design. However, due to the indirect reset technique used within the interconnect subsystem, problems with reset initialization of the emulator probe are not uncommon. Most of these can be easily resolved by having the emulator issue port I/O commands to clear the RSTNC* bit in interconnect before a timeout occurs (typically 25 seconds). In some cases even this action will be inadequate because many ICE™ products will not tolerate a condition where reset is held continuously active. The solution to this problem is to disable the RSTNC* timeout function in the microcontroller. The core firmware design includes a jumper selectable option to allow in-circuit emulators to operate without reset problems.

3.8 AVOIDING iPSB TIMEOUTS

The reader will by now appreciate that interconnect bus cycles can be potentially long due to the extensive amount of processing a microcontroller completes to satisfy an interconnect reference. Performance within the interconnect address space is not an issue since interconnect bus cycles are run during system initialization and diagnostic tests, but cycles are rarely run during normal system operation because of their affect on the bus bandwidth. If, for any reason an interconnect bus cycle on the iPSB cannot complete within 1 ms; then an iPSB timeout will cancel the transaction and an error status posted. Therefore, users are cautioned not to attempt extensive processing within their interconnect read, write, and edit routines. Worst case timing analysis should be run on all new interconnect function records to avoid unexpected timeouts. More information on this topic is provided along with examples in Chapter 5.

CHAPTER 4
INTERCONNECT HARDWARE DESIGN

4.1 MICROCONTROLLER PIN ASSIGNMENTS

The 8751 Microcontroller is well configured with I/O port facilities. In addition to the bidirectional Interconnect Address/Data bus (IAD<7..0>) connected to Port 0 (P0), there are 24 additional bidirectional lines and a number of port expansion techniques that can be employed. The core interconnect design will consume some of these resources, but beyond that, the user is free to dedicate the remaining I/O lines in any desired way. A variety of hardware design examples are presented here to satisfy the more typical requirements.

4.1.1 The Interconnect Address/Data Bus

The initial design of the Message Passing Coprocessor (MPC) anticipated that the interconnect subsystem would be partitioned in a microcontroller of the 8751 or 8749 family. For this reason, the MPC provides a directly compatible multiplexed IAD bus requiring no external logic to the microcontroller. From a hardware viewpoint, the MPC is a slave to the microcontroller. The MPC supplies an external interrupt (**IREQ***) whenever it needs service and the microcontroller satisfies the request. The eight multiplexed IAD lines are taken directly from P0 of the microcontroller and interface timing is based on the microcontroller's Interconnect address strobe (**IAST**), Read (**IRD***), and Write (**IWR***) control signals. While the primary purpose of the IAD bus is to provide the communications path between the MPC and interconnect subsystem, it should be noted that the MPC will remain tristate for addresses 0 through 7FH. This condition presents an opportunity for I/O port expansion and is discussed later in this chapter.

4.1.2 Reserved Pins on the 8751

In addition to the IAD bus on P0, the eight lines of Port 3 (P3) and two lines on Port 2 (P2) are reserved in support of the core interconnect design. (P3 is completely dedicated to hardware functions.) Figure 4-1 is a simplified schematic of the basic hardware requirements that guarantee proper functioning of the interconnect core. This includes the front panel LED, reset inputs and outputs, interrupt sources, and provisions for MPC loopback testing (discussed in Chapter 3).

Port function assignments are based on a series of equate statements contained in the program file DFT.MOD. If your interconnect design maintains the configuration shown in figure 4-1 and defined in DFT.MOD; then there will be no need to modify the equate statements. A total of 14 port pins are available for user-defined I/O schemes. If necessary, the microcontroller pin assignments can be changed by modifying DFT.MOD (contained in the \ICFW\SRC subdirectory).

4.2 EXTERNAL TTL OPTIONS

Having completed the hardware design requirements, it is time to include the user hardware support options you have decided on. If your board requires only the core functions, then the hardware design is complete and prototyping can begin. Continued reading of this chapter may give you additional ideas for features you may wish to add. Other readers will already have a well defined functional objective and should be able to satisfy their requirements using one or more of the techniques given below. Bear in mind that the application examples described in this chapter are only suggestions and a combination of techniques are used. This underscores the need for good engineering discipline — always perform the AC/DC timing analysis and loading calculations!

The range of possibilities is limited only by your own imagination, but practical considerations will prevail. Keep aware of board real estate and bus loading requirements, expected implementation costs, and ask yourself if the feature is really worth the effort. With these considerations in mind, we will now examine some of the many options available for hardware design.

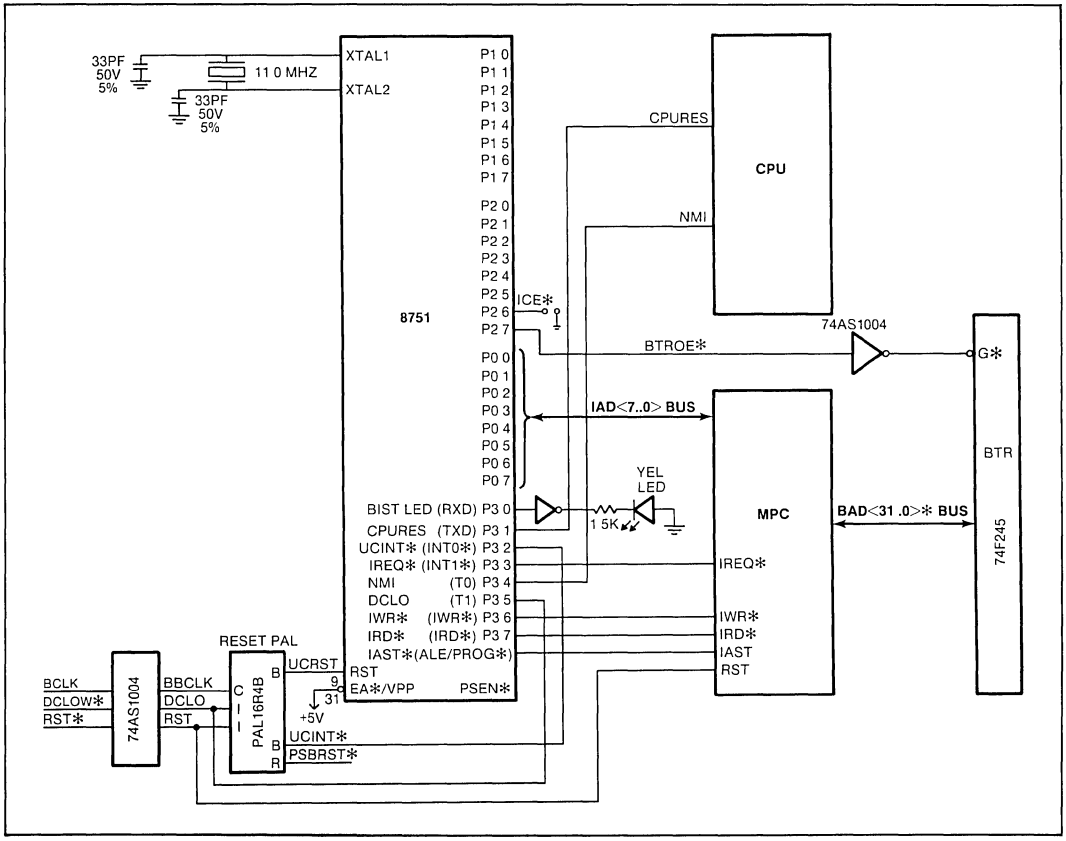


Figure 4-1. Interconnect Core Hardware Design

4.2.1 Microcontroller Input Options

On boards that contain user-supplied optional devices, it is desirable to include an interconnect status register to report whether or not the device is present. If present, system software can program the device with the appropriate driver. Some common examples include the Single Board Extension Bus (iSBX™) MULTIMODULE™ compatible board products, numeric processors, and DMA controllers.

A technique used to detect a board's presence is to identify a port pin that ties to ground and then have the microcontroller read that pin. The P1 and P2 lines of the 8751 Microcontroller are internally pulled up, which makes module not present for these lines always read a logical "1". Verifying if a chip is present is more difficult since the component needs the ground pin for its own power consumption. The preferred technique here is to ask the user to install a jumper and in that way, the microcontroller can report correct status.

Figure 4-2 schematically diagrams various techniques used to input to an 8751 Microcontroller. A typical technique is to require the microcontroller to read jumper inputs. (Although MULTIBUS® II has reduced the number of jumpers required, it hasn't eliminated them.) In cases where jumpers are unavoidable (such as chip select jumpers and component present indicators), it would be helpful to report the state of those jumpers with an interconnect status register. An example is the EPROM size register found on many Intel boards. If you have only a small number of jumpers, then a direct connection to one of the microcontroller's port pins will suffice. For boards with a larger number of

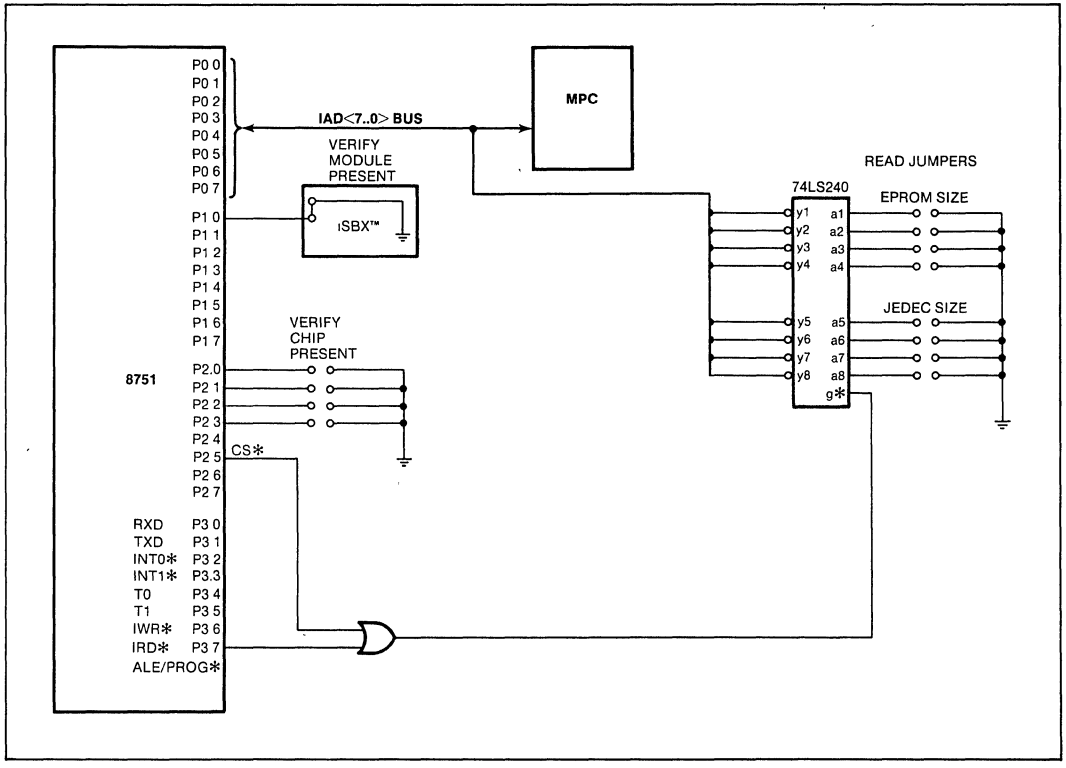


Figure 4-2. Microcontroller Input Options

jumpers, buffering will be necessary. The buffer output enable term is generated by a port pin in association with the microcontroller's **IRD*** signal. In the example shown, a read instruction to P2.5 will input data from the jumpers. This could be coded in ASM-51 as:

```
clrb        P2.5        ;activate CS for jumper
                        ;input buffer
mov   R0,   00h         ;load dummy address
movx  A     @R0,        ;read jumper inputs
setb        P2.5        ;deselect buffer
```

4.2.2 Microcontroller Output Options

Outputs from the microcontroller are used at various control points throughout MULTIBUS II boards. Simple functions such as the LED's can use a direct connection to a microcontroller port pin. More complex functions will require a connection to the buffered IAD bus. As a general rule, whenever more than two loads are on the IAD bus, it will need buffering. The MPC must be connected directly to the IAD bus to ensure proper timing. The schematic in figure 4-3 diagrams some typical applications.

4.2.3 LED Outputs

Nearly all MULTIBUS II boards contain one or more LED indicators on the front panel. These provide a visual indication of board activity and status. Typically, the red LED is provided as a user programmable indicator and is illuminated by the setting or clearing of a bit in a control register. A green LED is often used to indicate CPU activity. If the green LED is present, drive it with an Address Latch Enable (**ALE**) signal or equivalent. The yellow LED, if present, is lit during diagnostic testing and represents the ORed condition of the following bits in the BIST Slave Status Register:

BIST running + BIST failed + RSTNC timeout.

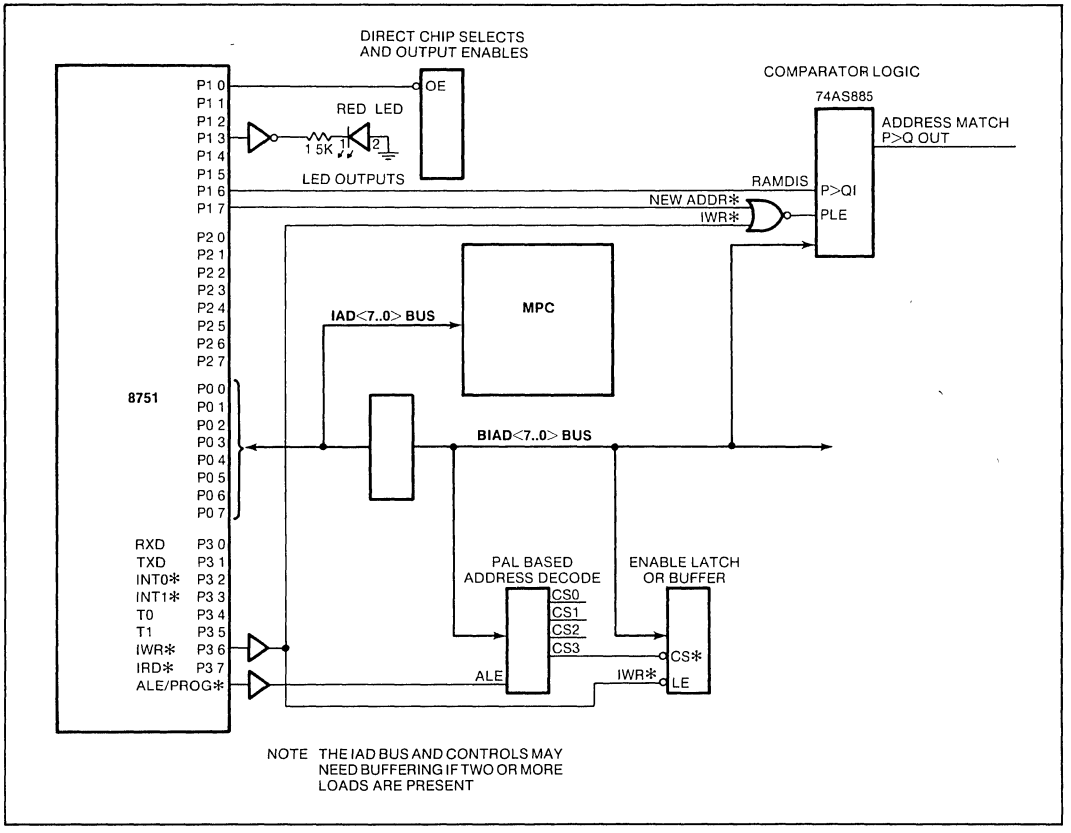


Figure 4-3. Microcontroller Output Options

4.2.4 Output Enables and Chip Selects

A microcontroller port pin can be used for a bus transceiver, **ALE**, or a direct chip select enable. When used as a direct chip select, care must be taken to guarantee proper timing and maintaining the state relationships with the other board occupants. A common application of this technique is a memory enable and disable signal. Direct chip selection is also used to enable the iPSB bus transceivers (via signal **BTROE***) in the core interconnect design.

4.2.5 Address Decode on the IAD Bus

The IAD bus address can be mapped to generate chip selects using simple "1 of n" decoders gated by a **ALE** signal. As an alternative, a PAL-based decoder can be used. A variety of chip select signals can be generated using this technique. Some designs use the chip selects to enable secondary latches and buffers. The ASM-51 coding for the circuit shown in figure 4-3 could look something like this:

```
mov   A,     ICDATA       ;Get value being written
mov   R0,    Latch_addr   ;Load the address of the
                          ;latch
movx  @R0,   A            ;Generate chip select and
                          ;write data
```

4.2.6 Comparators

Comparators are often used for address boundary checks in association with Local Bus Extension (iLBX™) or local memory. A comparator logic example is shown in figure 4-3. A select term can be generated on greater than, less than, or equal to

the data placed on the IAD bus. The set point for the comparison is latched-in during the data phase of the microcontroller's P0 bus. In this way, a port pin (the **NEW ADDR*** signal) can combine with the microcontroller's **IWR*** to control when the address comparison is enabled. The ASM-51 code to load a new address into the comparator could look like this:

```
mov   A,     ICDATA      ;Get the new setpoint
clrb         NEW_ADDR    ;This is the PLE signal
mov   R0,    00h         ;Dummy address
movx  @R0,   A           ;Output the new address
setb         NEW_ADDR    ;Latch it in
```

4.3 BIDIRECTIONAL I/O ON THE IAD BUS

Even a modest implementation of interconnect can easily exceed the available port resources of the microcontroller, especially when a byte wide data path is required. For these applications, numerous port expansion techniques exist. Your choice of which technique is best in your particular design will be based on such factors as on board space requirements, cost, bidirectionality, and ease of programming. An example of interfacing a complex peripheral to the IAD bus is shown in figure 4-4. The ASM-51 coding for this circuit is very straight forward since device selection is based on an IAD bus address.

```
mov   R0,    IO_ADDR     ;This is the address of
                         ;the device
movx  A,     @R0         ;Input from the device
mov   ICDATA, A          ;Save the data

mov   R0,    IO_ADDR     ;This is the address of
                         ;the device
mov   A,     ICCNTR      ;Load a control value for
                         ;output
movx  @R0,   A           ;Output to the device
```

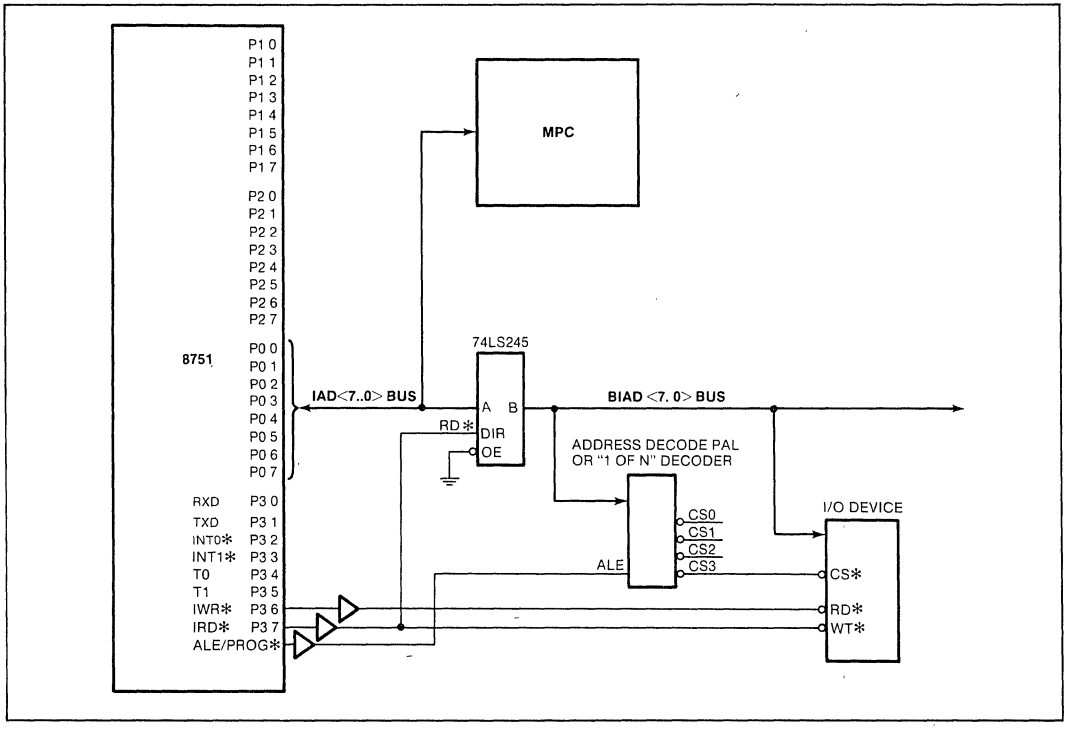


Figure 4-4. Bidirectional I/O on the IAD Bus

CHAPTER 5
DEFINING THE INTERCONNECT TEMPLATE

5.1 WHAT GOES INTO INTERCONNECT?

The design implementation phase of your interconnect subsystem begins with a careful evaluation of what functions to include. Common core functions are supplied on the accompanying diskette and you can add supplemental records to the interconnect template for board-level features you intend to support. Many frequently used function records are already defined in the Interconnect Interface Specification and the reader is requested to review this document before proceeding. If none of the furnished function records meet your needs, you can to create your own function record using one of the record types available for vendor definition or board specific functions.

If your function record is likely to have broad industry appeal and could be used by other vendors to support similar functions, you may wish to fill out the petition application included with the reader comment sheet at rear of this guide. Your petition will be evaluated based on suitability for a general class of hardware, industry standardization, and feedback from the MULTIBUS® II user community, and if accepted, published in the next revision to Interconnect Interface Specification.

5.1.1 Core Functions Supplied

The interconnect firmware supplied with this guide contains a group of records common to most MULTIBUS II boards containing a CPU, regardless of their specific function. These include an Interconnect Header Record, a Protection Record, the iPSB Control Record, a Local Processor Record, and an End of Template (EOT) Record. Some users may wish to generate a minimal interconnect template on the first pass to verify the operation of their base hardware before adding their own advanced functions. This approach simplifies debugging and allows new features to be added incrementally.

5.1.2 Getting Started: The Interconnect Worksheet

Unless you are using the supplied interconnect template without modification, it is a two-step process to complete the configuration tables. In the first step, you will fill-in a worksheet with information about default values, RAM usage, read/write and edit routine numbers, and other details. Figure 5-1 is an example of such a worksheet. While completing these tables, you might notice opportunities to reduce code size by making use of generic read, write, and edit routines. In the second step, the information in the worksheet is copied into a series of tables used by the configuration macros.

5.2 THE INTERCONNECT HEADER RECORD

The interconnect template begins with a standard Interconnect Header Record. This record contains fields that require specific values supplied by the user. You should fill-in this information in as we proceed through this chapter. For a detailed discussion concerning each register, consult Appendix A of the Interconnect Interface Specification.

5.2.1 Vendor ID Register (0-1)

Licensed MULTIBUS II vendors are assigned two vendor ID numbers — an odd number for conforming templates and an even number for templates not conforming to the IEEE 1296 specification. You should enter your odd vendor ID number, since this template will be conforming. Nonlicensed users who are building custom MULTIBUS II boards should use 65533 (0FFFDH) as their vendor ID.

REGISTER NAME	GAR	RAM	DFT	RRT	ERT	WRT	CONST	EET	VAR
HEADER RECORD									
VENDOR ID, LOW BYTE	0	0	____	000H	000H	?	?	000H	?
VENDOR ID, HIGH BYTE	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 1	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 2	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 3	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 4	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 5	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 6	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 7	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 8	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 9	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 10	0	0	____	000H	000H	?	?	000H	?
INTEL RESERVED	0	0	000H	000H	000H	?	?	000H	?
INTEL RESERVED	0	0	000H	000H	000H	?	?	000H	?
INTEL RESERVED	0	0	000H	000H	000H	?	?	000H	?
INTEL RESERVED	0	0	000H	000H	000H	?	?	000H	?
HARDWARE TEST REV NO.	0	0	____	000H	000H	?	?	000H	?
CLASS ID	0	0	____	000H	000H	?	?	000H	?
RFU	0	0	000H	000H	000H	?	?	000H	?
RFU	0	0	000H	000H	000H	?	?	000H	?
RFU	0	0	000H	000H	000H	?	?	000H	?
RESET STATUS RGTR	0	1	003H	001H	000H	?	?	000H	RST
PROGRAM TABLE INDEX	1	1	000H	001H	001H	000H	00H	001H	PTI
NMI ENABLE RGTR	1	1	000H	001H	001H	000H	38H	002H	NMI_EN
GENERAL STATUS	0	1	000H	003H	000H	?	?	000H	GEN_STS
GENERAL CONTROL	1	1	000H	001H	001H	002H	78H	003H	GEN_CTL
BIST-SUPPORT-LEVEL	0	1	000H	001H	005H	000H	18H,07H,04H	004H	BIST_SL
BIST-DATA-IN	1	1	000H	001H	001H	000H	00H	001H	BIST_DI
BIST-DATA-OUT	0	1	000H	001H	001H	000H	00H	001H	BIST_DO
BIST-SLAVE-STATUS	0	1	010H	001H	001H	000H	00H	001H	BIST_SS
BIST-MASTER-STATUS	1	1	020H	001H	001H	000H	50H	005H	BIST_MS
BIST-TEST-ID	0	1	000H	001H	002H	000H	01H,0FEH	006H	BIST_TID
PROTECTION RECORD									
PROTECTION RECORD TYPE	0	0	00BH	000H	000H	?	?	000H	PROT_OFF
RECORD LENGTH	0	0	002H	000H	000H	?	?	000H	?
PROTECTION LEVEL RGTR	0	2	000H	004H	001H	003H	0FEH	007H	UACRS
RFU	0	0	000H	000H	000H	?	?	000H	?
PSB CONTROL RECORD									
PSB CONTROL RECORD TYPE	0	0	006H	000H	000H	?	?	000H	PSB_CTL_OFF
RECORD LENGTH	0	0	006H	000H	000H	?	?	000H	?
PSB SLOT ID NO.	0	0	000H	002H	000H	?	?	000H	?
PSB ARBITRATION ID NO.	1	0	000H	002H	003H	?	?	008H	?
PSB ERROR RGTR	1	0	000H	002H	001H	001H	0FFH	009H	?
PSB CONTROL/STATUS RGTR	1	1	01FH	005H	001H	004H	63H	00AH	PSB_CS
PSB DIAGNOSTICS RGTR	1	1	000H	002H	004H	?	?	00BH	PSB_DIAG
RFU	0	0	000H	000H	000H	?	?	000H	?

Figure 5-1. Interconnect Configuration Worksheet (Sheet 1 of 2)

REGISTER NAME	GAR	RAM	DFT	RRT	ERT	WRT	CONST	EET	VAR
-----LOCAL PROC RECORD -----									
LOCAL PROC RECORD TYPE	0	0	013H	000H	000H	?	?	000H	?
RECORD LENGTH	0	0	003H	000H	000H	?	?	000H	?
LOCAL PROC CONTROL	0	0	000H	006H	001H	006H	0FEH	00CH	?
LOCAL PROC STATUS	0	0	000H	007H	000H	?	?	000H	?
RFU	0	0	000H	000H	000H	?	?	000H	?
----- END OF TEMPLATE RECORD -----									
EOT RECORD TYPE	0	0	0FFH	000H	000H	?	?	000H	EOT_OFF

Figure 5-1. Interconnect Configuration Worksheet (Sheet 2 of 2)

5.2.2 Board ID Registers (2-11)

These 10 registers contain the ASCII product code as described in your user's manual. You should avoid the use of nonprinting ASCII characters so the system software can display the product code verbatim on a CRT screen. If the product code is less than 10 characters, pad the remaining register fields with ASCII null characters (zeros).

5.2.3 Hardware Test Revision Number Register (16)

This register is part of a system of control for matching the firmware revision to the automatic test equipment used with that board. The master or local test handler software uses this as a mechanism for revision control of diagnostics.

5.2.4 Class ID Register (17)

Determine your class ID from the list supplied in the Interconnect Interface Specification. The class ID occupies the most significant nibble (4 bits) of the register and the subclass ID occupies the least significant nibble.

5.2.5 BIST Support Level Register (26)

The BIST Support Level Register defines the support level of the diagnostics on-board. This will determine which commands can be issued to your board via the BIST Data Input Register.

5.3 OPTIONAL USER FUNCTIONS

There is virtually no limit to the number and variety of function records that users can add to the core firmware — provided the following common sense guidelines are met:

a. A maximum of 256 interconnect registers are allowed due to the page limit length of the configuration tables. Most boards will need fewer than 100 registers.

b. The microcontroller you are using has certain resource limitations — the most pressing of which is on-chip RAM. Nested stack operations also require RAM, so you must avoid recursive procedures and nesting of calls beyond four levels in any of the USER code modules. Normally, it is preferable to pass parameters in registers rather than on the stack.

c. An interconnect request must complete operations within 1-ms to avoid an iPSB timeout. The actual time available to complete a USER routine is less than 0.5-ms because interconnect is dual ported and a local access can temporarily lockout an iPSB request. A timing analysis of your design will be done prior to PROMming the code (refer to paragraph 5.8.2).

d. Caution must be exercised whenever the potential for deadlock of a resource exists. Solutions to this problem include prioritization, reporting of error status, or timeout to guarantee that system hang does not occur.

e. User read, write and edit routines must follow the naming convention described in this chapter to

guarantee that the macro expansion works properly. Any global variable names used within existing core modules are considered reserved and should not be used in your own procedures (ASM-51 and/or RL51 will report duplicate symbol errors). A list of existing public variables may be found in program file ICFW\SRC\USER.MOD.

5.4 COMPLETING THE INTERCONNECT CONFIGURATION WORKSHEET

If you decided that a user-defined function record is to be incorporated in the interconnect template, then you must define the access rights, default values, and complete the remaining worksheet entries needed by the microcontroller firmware to support those features. You then write the functional read, write, and edit routines and tag them with appropriate labels. The new function record(s) must be inserted ahead of the EOT record of the core interconnect template.

As discussed in Chapter 3, the interconnect firmware uses a series of tables to associate a register with the correct read, write, and edit routine numbers. The information you supply here will be loaded into those tables during the code assembly process. In figure 5-2, an example of a completed interconnect configuration worksheet for the core firmware is shown. The parameters that must be supplied for each interconnect register identified on the configuration worksheet are described below.

5.4.1 Global Access Rights

All interconnect registers are readable from both the local and the iPSB bus. The state of the Global Access Rights (GAR) bit determines whether the register can be written to by another iPSB agent. If this bit is 0, then the register is read only on the iPSB bus and edit routine ED00H is used. A register will be locally read/writable only if an edit routine other than ED00H is supplied for this entry. If the GAR bit is 1, then this register is read/writable on both interfaces and an edit routine must be supplied.

5.4.2 RAM Usage

The microcontroller RAM is a precious resource and must be managed as efficiently as possible. Some registers in interconnect may contain static values that never change. These values should be stored in microcontroller ROM. In cases where the contents of an interconnect register are variable, either a byte or a bit of RAM must be allocated to store data, unless the data can be read directly from a port. The microcontroller's bit manipulation features can be used when a full 8-bit register is not required. The RAM entry in the worksheet must specify whether to reserve a bit location (RAM = 2), a byte location (RAM = 1), or no memory at all (RAM = 0). If RAM memory is requested, then a name for the public variable for that location must be supplied (refer to paragraph 5.4.9).

5.4.3 Default Value

The Default Value (DFT) is a hex number placed in this register during a cold or recovery reset. If this is a static register, then the default value represents the permanent contents of that register. The default value is also placed into dynamic registers following a cold reset; however, the register contents may subsequently change due to reprogramming or environmental changes (e.g., reading a value from a port).

5.4.4 Read Routine Entry

Read routines contain the code that actually perform the work in satisfying an interconnect read request. Eight read routines are used in the interconnect core firmware. These are listed below and all are user callable. Read routines are sequentially numbered from 00H to nnH and contain no numbering gaps. A single read routine may be referenced by several different interconnect registers. If you are adding a new read routine, the first available Read Routine (RRT) number would be RD08H. The value supplied for the RRT is the two digit hex portion of the read routine number.

```
RD00h  Load Default Value into ICDATA    Generic
RD01h  Get RAM Access Rights             Generic
RD02h  PSB Control/Status Register       Specific
RD03h  General Status Register           Specific
RD04h  Protection Level Register         Specific
RD05h  MPC General Parameter Register    Specific
RD06h  Local Processor Control Register  Specific
RD07h  Local Processor Status Register   Specific
```

```
;*******************************************************
;
;  NAME:          TABGEN  (TABLE GENERATION)
;
;  DESCRIPTION:   USES ENTAB MACRO TO ASSOCIATE EACH IC
;                 REGISTER WITH:
;                        GAR - GLOBAL ACCESS RIGHTS
;                        RAM - RAM USEAGE (WHETHER OR
;                              NOT UC RAM IS NEEDED)
;                        DFT - DEFAULT VALUE
;                        RRT - READ ROUINTE NUMBER
;                        EET - EDIT ENTRY NUMBER
;                        VAR - SYMBOL FOR RAM/BIT
;                              VARIABLE
;
; UPDATE HISTORY: 5-14-87 JR UPDATE FOR IDX
;
;*******************************************************
;       G R
;       A A
;       R M  DFT   RRT   EET   VAR
;-------------- HEADER RECORD -----------------------
%ENTAB(0,0,001H,000H,000H,?)       ;VENDOR ID, LOW BYTE
%ENTAB(0,0,000H,000H,000H,?)      ;VENDOR ID, HIGH BYTE
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 1
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 2
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 3
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 4
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 5
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 6
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 7
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 8
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 9
%ENTAB(0,0,000H,000H,000H,?)    ;BOARD ID, CHARACTER 10
%ENTAB(0,0,000H,000H,000H,?)           ;INTEL RESERVED
%ENTAB(0,0,000H,000H,000H,?)           ;INTEL RESERVED
%ENTAB(0,0,000H,000H,000H,?)           ;INTEL RESERVED
%ENTAB(0,0,000H,000H,000H,?)           ;INTEL RESERVED
%ENTAB(0,0,001H,000H,000H,?)     ;HARDWARE TEST REV NO.
%ENTAB(0,0,000H,000H,000H,?)                 ;CLASS ID
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
;       G R  DFT  RRT  EET  VAR
%ENTAB(0,1,003H,001H,000H,RST_STS)  ;RESET STATUS RGTR
%ENTAB(1,1,000H,001H,001H,PTI)     ;PROGRAM TABLE INDEX
%ENTAB(1,1,000H,001H,002H,NMI_EN)      ;NMI ENABLE RGTR
%ENTAB(0,1,000H,003H,000H,GEN_STS)      ;GENERAL STATUS
%ENTAB(1,1,000H,001H,003H,GEN_CTL)     ;GENERAL CONTROL
%ENTAB(0,1,000H,001H,004H,BIST_SL) ;BIST-SUPPORT-LEVEL
%ENTAB(1,1,000H,001H,001H,BIST_DI)        ;BIST-DATA-IN
```

Figure 5-2. ENTAB Macro Source Code Listing (Sheet 1 of 2)

```
%ENTAB(0,1,000H,001H,001H,BIST_DO)      ;BIST-DATA-OUT
%ENTAB(0,1,010H,001H,001H,BIST_SS)  ;BIST-SLAVE-STATUS
%ENTAB(1,1,020H,001H,005H,BIST_MS) ;BIST-MASTER-STATUS
%ENTAB(0,1,000H,001H,006H,BIST_TID)      ;BIST-TEST-ID
;---------- PROTECTION RECORD ----------------------
;      G R  DFT  RRT  EET  VAR
%ENTAB(0,0,00BH,000H,000H,PROT_OFF)  ;PROTECT REC TYPE
%ENTAB(0,0,002H,000H,000H,?)             ;RECORD LENGTH
%ENTAB(0,2,000H,004H,007H,UACRS)   ;PROTECT LEVEL RGTR
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
;---------- PSB CONTROL RECORD ---------------------
;      G R  DFT  RRT  EET  VAR
%ENTAB(0,0,006H,000H,000H,PSB_CTL_OFF)    ;PSB CR TYPE
%ENTAB(0,0,006H,000H,000H,?)             ;RECORD LENGTH
%ENTAB(0,0,000H,002H,000H,?)           ;PSB SLOT ID NO.
%ENTAB(1,0,000H,002H,008H,?)            ;PSB ARB ID NO.
%ENTAB(1,0,000H,002H,009H,?)            ;PSB ERROR RGTR
%ENTAB(1,1,01FH,005H,00AH,PSB_CS)  ;PSB CONT/STAT RGTR
%ENTAB(1,1,000H,002H,00BH,PSB_DIAG)     ;PSB DIAG RGTR
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
;------------ LOCAL PROC RECORD ---------------------
;      G R  DFT  RRT  EET  VAR
%ENTAB(0,0,013H,000H,000H,?)   ;LOCAL PROC RECORD TYPE
%ENTAB(0,0,003H,000H,000H,?)             ;RECORD LENGTH
%ENTAB(0,0,000H,006H,00CH,?)        ;LOCAL PROC CONTROL
%ENTAB(0,0,000H,007H,000H,?)         ;LOCAL PROC STATUS
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
;--------- END OF TEMPLATE RECORD -----------------
;      G R  DFT  RRT  EET  VAR
%ENTAB(0,0,0FFH,000H,000H,EOT_OFF)    ;EOT RECORD TYPE
$EJECT
```

Figure 5-2. ENTAB Macro Source Code Listing (Sheet 2 of 2)

5.4.5 Edit Routing Table

The purpose of the Edit Routine Table (ERT) parameter in the worksheet is to associate each register with the correct edit routine number. There are six edit routines supplied within this firmware:

ED00h	Used for Read-Only Registers	Generic
ED01h	Check that RFU Bits Match a Mask	Generic
ED02h	Range Checking Between Two Constants	Generic
ED03h	PSB ARB ID Register	Specific
ED04h	PSB Diagnostic Register	Specific
ED05h	Greater Than Check (>)	Generic

5.4.6 Write Routine Number

The Write Routine (WRT) parameter specifies which write routine number to jump to once access rights and value checking are complete. A given edit routine can jump to any write routine (there is no requirement for the edit routine number to match the write routine number). In the generic routines, several edit routines can reference the same write routine using different constant lists based on register number. In the specific routines, it is sometimes more expedient to jump directly to the edit routine rather than going through the

write jump table. In this case, a "?" should be entered for the WRT. Seven write routines are supplied, therefore, the first available number assignment for user code is WR07H.

```
WR00h  Write to a RAM Based Register          Generic
WR01h  Write to a MPC Based Register          Generic
WR02h  Write to PSB Control Register          Specific
WR03h  Write to Protection Level Register     Specific
WR04h  Write to PSB Control/Status Register   Specific
WR05h  Write to PSB Diagnostic Control        Specific
WR06h  Write to Local Processor Control       Specific
```

5.4.7 Constants for Value Checking

This column contains a list of constants (CONST) used by the edit routines in value checking register contents. During an edit routine, illegal values cause error reports to the General Status Register and the IC handler will exit without performing the write operation. The advantage of using the CONST is that edit routines can be made "sharable" because a different list of constants is used for each register. If no constants are required, a "?" should be entered. This implies that your value checking algorithm will use constants supplied by the edit routine (i.e., specific to that register).

5.4.8 The Edit Entry Table

The Edit Entry Table (EET) number is a value used as the index into the edit entry table during table look up operations. The order of how EET numbers get assigned is not important, however, numbering must start at 0H and run in sequence until every register has an assigned EET number. Two or more registers having the same attributes share the same EET number. This is true if (and only if) all of the columns match for the ERT, WRT, and CONST values. When this occurs, there is an opportunity for code compaction and the same EET number will be assigned to two or more registers.

5.4.9 Symbolic Reference to RAM Locations

Whenever a dynamic register is defined (RAM = 1 or 2), an on-chip RAM bit or memory location is reserved. Entries in the Symbolic Reference to RAM (VAR) column represent the symbolic name to be used for references to that RAM bit or memory location. The VAR entry serves an important purpose — it reserves a location in on-chip memory and declares a public symbol under the name of that variable. Read, write and edit routines can now reference that variable directly without needing to know it's exact location.

One word of caution with respect to bit variable declarations. If the RAM parameter in the worksheet is a byte location (RAM=1), then the value listed as default will be loaded into the location reserved for that symbolic name at initialization time as expected (VAR=DFT). But if a bit location was reserved (RAM=2), the current software has no provision to load the reserved bit location with its default value. Users are therefore advised to initialize all reserved bit locations explicitly in their INIT_USER routines. The public variable specified by the VAR parameter will be in effect and can be used as a symbolic reference for direct addressing (e.g., setb/clrb var). An example of the use of bit variables can be found in the RD04 and WR03 routines contained in Appendix C.

5.5 LOADING THE MACRO TABLES

The interconnect worksheet was used as an intermediate step to allow the user to identify opportunities for code reduction through the use of generic read, write, and edit routines. Now that the worksheet is complete, you must enter this data into table generating macro files on your IBM® PC. The macro assembler will use these tables to generate the final interconnect firmware code.

5.5.1 The ENTAB Table

The program file \ICFW\SRC\TABGEN.A51 contains a list of calls to the ENTAB configuration macro (figure 5-2). This list determines the register order in the interconnect template. Enter the values for GAR, RAM, DFT, RRT, EET, and VAR from the worksheet into this table.

5.5.2 The EETGEN Table

The program file \ICFW\SRC\EETGEN.A51 contains a list of calls to the EETGEN configuration macro (figure 5-3). This list is used to build the edit entry table. Enter the values for ERT, CONST, and WRT from the worksheet into this table.

```
;*****************************************************
;
;  NAME:          EELBL (EDIT ENTRY TABLE)
;
;  PURPOSE:       PROVIDE EDIT ROUTINE ADDRESSES,
;                 VALUE CHECKING CONSTANTS, AND WRITE
;                 ROUTINE ADDRESSES FOR THE IC MODULE
;
;  DESCRIPTION:   A ONE DIMENSIONAL LOOKUP TABLE THAT
;                 SPECIFIES LOW BYTES OF ADDRESSES
;                 THAT RESULT FROM EXPRESSIONS, i.e.
;                 (ERXX-ERLBL), AND ALSO PROVIDES
;                 CONSTANTS THRU THE USE OF SYMBOLS
;
;  UPDATE HISTORY: 5-14-87 JR UPDATED FOR IDX
;
;*****************************************************
;
;      ERT   CONST   WRT

%EEGEN(00H,%(?),?)            ;READ ONLY REGISTERS     0
%EEGEN(01H,%(00H),00H)        ;PROGRAM TABLE INDEX     1
%EEGEN(01H,%(038H),00H)       ;NMI ENABLE              2
%EEGEN(01H,%(078H),02H)       ;GENERAL CONTROL         3
%EEGEN(05H,%(018H,07H,04H),00H) ;BIST SUPPORT LVL      4
%EEGEN(01H,%(050H),00H)       ;BIST MASTER STATUS      5
%EEGEN(02H,%(01H,0FEH),00H)   ;BIST TEST ID            6
%EEGEN(01H,%(0FEH),03H)       ;PROTECTION LEVEL        7
%EEGEN(03H,%(?),?)            ;PSB ARB. ID             8
%EEGEN(01H,%(0FFH),01H)       ;PSB ERROR               9
%EEGEN(01H,%(063H),04H)       ;PSB CONTROL STATUS      A
%EEGEN(04H,%(?),?)            ;PSB DIAGNOSTIC          B
%EEGEN(01H,%(0FEH),06H)       ;LOCAL PROC CONTROL      C
$EJECT
```

Figure 5-3. EEGEN Macro Source Code Listing

5.5.3 The EXTERNS Macro: Generating External Labels

After completing the entries in the ENTAB and EETGEN tables, the final step before code generation is to supply values to the EXTERNS macro so that the correct number of external labels are generated for your user routines. The EXTERNS macro is used in the file \ICFW\SRC\TABLE.MOD. Here you will find the call to EXTGEN, where you must supply the highest number of your edit, write, and read routines as parameters in that order:

```
%EXTERNS(06,07,08)      ;Six edit routines
                        ;Seven write routines
                        ;Eight read routines
```

5.6 PROGRAMMING TECHINQUES FOR READ, WRITE, AND EDIT ROUTINES

There are a group of user callable macro functions supplied with the interconnect firmware in the file \ICFW\SRC\GLOBAL.MAC. These are called by entering a "%" symbol in front of the name and supplying a list of parameters. This results in code

expansion, which supplies the requested function. The following macros are callable by users.

```
RD_MPC (dest,src)          Read MPC Register
WRT_MPC(dest,src)          Write MPC Register
LOOKUP(table,offset)       Retrieve parameter from
                            table
SUB (opr)                  Subtract operand from
                            accumulator
MOVBIT(dest_bit,src_bit)   Perform bit move
                            operation
GET_EEC                    Get Edit Table Constant
                            (Data pointer must be
                             pointing to EETABLE.
                             Byte is returned in
                             Accumulator.  You must
                             perform one LOOKUP be-
                             fore making this call.)
```

All user routines should be declared PUBLIC so that the table generating macros can locate the routines entry point. Place the user routines you write in the file ICFW\SRC\USER.MOD.

5.6.1 Retrieving the Constant List from the EETABLE

While in a read, write, or edit routine, one or more constants can be retrieved by using the global symbol EET_OFF as the offset into the EETABLE for this register. For example, consider this range checking algorithm in an edit routine:

```
ER05H: INC   EET_OFF              ;Point to RFU Mask
                                  ;in EETABLE
       %LOOKUP(#EETABLE,EET_OFF) ;Get It
       ANL   A,ICDATA             ;Check all RFUs=0
       JNZ   ERR_5                ;JMP if Illegal
       %GET_EEC                   ;Get Don't Care Mask
                                  ;From EETABLE
       ANL   A,ICDATA             ;Mask off Don't Care
                                  ;Bits from ICDATA
       MOV   TEMP,A               ;Save Result
       %GET_EEC                   ;Get Max Allowable
                                  ;Write Value
       %SUB(TEMP)                 ;SUB Value Being
                                  ;Written
       JC    ERR_5                ;If ICDATA > Max
                                  ;Allowable, ERR
       %GET_EEC                   ;Else Get Write
                                  ;Routine Offset
       MOV   DPTR,#WRLBL          ;Point to WR
       JMP   @A+DPTR              ;GOTO WR
ERR_5: AJMP  VALERR               ;Report Value ERR
```

5.6.2 Handling of Value Errors

If a value error is detected in an edit routine, then the write operation will not be performed and instead, a jump to a public routine VALERR should be executed. This routine updates the General Status Register with the appropriate Value Error and returns to the IC handler to complete the interconnect operation.

5.7 HOOKS FOR USER SUPPLIED ROUTINES

Every board design is just a little bit different and recognizing this fact, calls to user supplied routines are provided at certain critical locations in the code. These include initialization, polling, and reset routines and timer based functions. As delivered, these routines are nothing more than program stubs. You can find them in the \ICFW\SRC\USER.MOD file. The following discussion suggests some ideas for what you might do with the user routine calls in your interconnect design.

5.7.1 INIT USER: Custom Initialization Code

The INIT_USER routine is only called on a cold reset. It gives the user an opportunity to initialize their I/O and modify defaults before entering the mainline code. This routine also gives the user a chance to modify the MPC register defaults after the call to MPC_INIT. If you are using any bit segments (RAM=2), they must be explicitly initialized at this time. A special feature has been added to the INIT_USER routine which disables the RSTNC timeout function based on a jumper input. This helps out ICE™ users since these emulators have difficulty dealing with external resets.

5.7.2 RST USER: Special Handling upon Warm Reset

When RST_USER is called, you should reset any I/O devices connected to the microcontroller. If dual-port memory is present, rewrite the upper and lower address boundries to the MPC because the internal MPC registers are all cleared after an iPSB reset. In most cases, the contents of interconnect registers will remain unchanged.

5.7.3 POLL USER: Polled User Functions

The POLL_USER routine is called from the mainline code. It polls user functions on each loop through the mainline code (approximatly 39μs).

Typical uses are to scan for on-board errors and to set the general error status bit in the General Status Register when errors are found.

5.7.4 USER TIMER: Timer Based Functions

The USER_TIMER routine is called every 1-ms as part of timer 0 interrupt routine. USER_TIMER is similar to the POLL_USER, except that the granularity of the timer is predictable since the timer 0 interrupt has the highest priority (other than reset). The USER_TIMER routine is used for self-toggling resets and other time based functions. A software prescale counter can be maintained for timing longer intervals.

5.8 GENERATING THE OBJECT MODULE

The final step in code preparation is to assemble all of your new user routines with ASM-51 and then link to the core module using the RL51 linker. Two batch files are supplied to automate the firmware generation process. To use these files, the current directory must be the \ICFW\SRC subdirectory. Here you will find three files with a .MOD file extension. These are the default module, the tables, and your user code. To assemble these modules enter:

```
ASMMOD  dft
ASMMOD  table
ASMMOD  user
```

The list files will be placed in the \ICFW\SRC\LST subdirectory and the object files are placed in the \ICFW\SRC\OBJ subdirectory. Next, the object modules will be linked with the core library to produce a PROM image. Select a name for the output file (e.g. TEST), and invoke the linker by entering:

```
LNKUSR  file name
```

You will now find a PROMmable file with a .LNK extension in the \ICFW\SRC\OBJ subdirectory. There will also be a file in this subdirectory with a .MAP extension. This file (a link map) will be used in the next step to verify microcontroller resources.

5.8.1 Checking Microcontroller Resource Utilization

The link map file contains information about how much code and register space was requested by the core and user routines and where symbolic variables are located. Examine the link map to check that the limits on RAM and ROM usage have not been exceeded. A gap will be reported whenever additional free space is available. The core firmware will create the following segments:

```
LINK MAP FOR TEST.LNK(MAIN)

TYPE   BASE      LENGTH    RELOCATION   SEGMENT NAME
----   -----     -----     --------     ------------
REG    0000H     0008H                  "REG BANK 0"
DATA   0008H     000DH     UNIT         RW_SEG
DATA   0015H     0008H     UNIT         DATA_SEG
       001DH     0003H                  *** GAP ***
DATA   0020H     0002H     BIT_ADDR     BIT_ADDR_BYTE_SEG
BIT    0022H     0000H.2   UNIT         BIT_SEG
       0022H.2   0000H.6                *** GAP ***
IDATA  0023H     0008H     UNIT         STACK_SEG

CODE   0000H     004EH     ABSOLUTE
CODE   004EH     0433H     UNIT         CODE_SEG
CODE   0481H     00BFH     UNIT         EDIT_SEG
CODE   0540H     0067H     UNIT         WRITE_SEG
CODE   05A7H     005AH     UNIT         READ_SEG
CODE   0601H     000AH     UNIT         USER_CSEG
       060BH     09F3H                  *** GAP ***
CODE   0FFEH     0002H     ABSOLUTE
```

5.8.2 Checking Critical Timing Paths

Figure 5-4 shows a state diagram of the microcontroller core firmware and includes the instruction cycle counts for each of the major functions. In this step, you will evaluate the impact user code has on interconnect subsystem timing. All timing calculations are based on counting microcontroller machine cycles to compensate for the different crystal frequencies that may be used. To calculate the elapsed time, multiply the cycle count by 1/xtal frequency (i.e., 0.909 μs at 11-MHz). Some microprocessors have specific timing requirements for the pulse width of reset and and interrupt signals. These should be checked against the timing values given below:

Reset Timing:

Max. RES Latency from UCINT = 41 Cycles
Min. RES Inactive after UCINT Inactive = 38 Cycles + RST_USER

NMI Timing:

Max. Interrupt Latency = Mainline + IC Handler= 39 + 136 Cycles

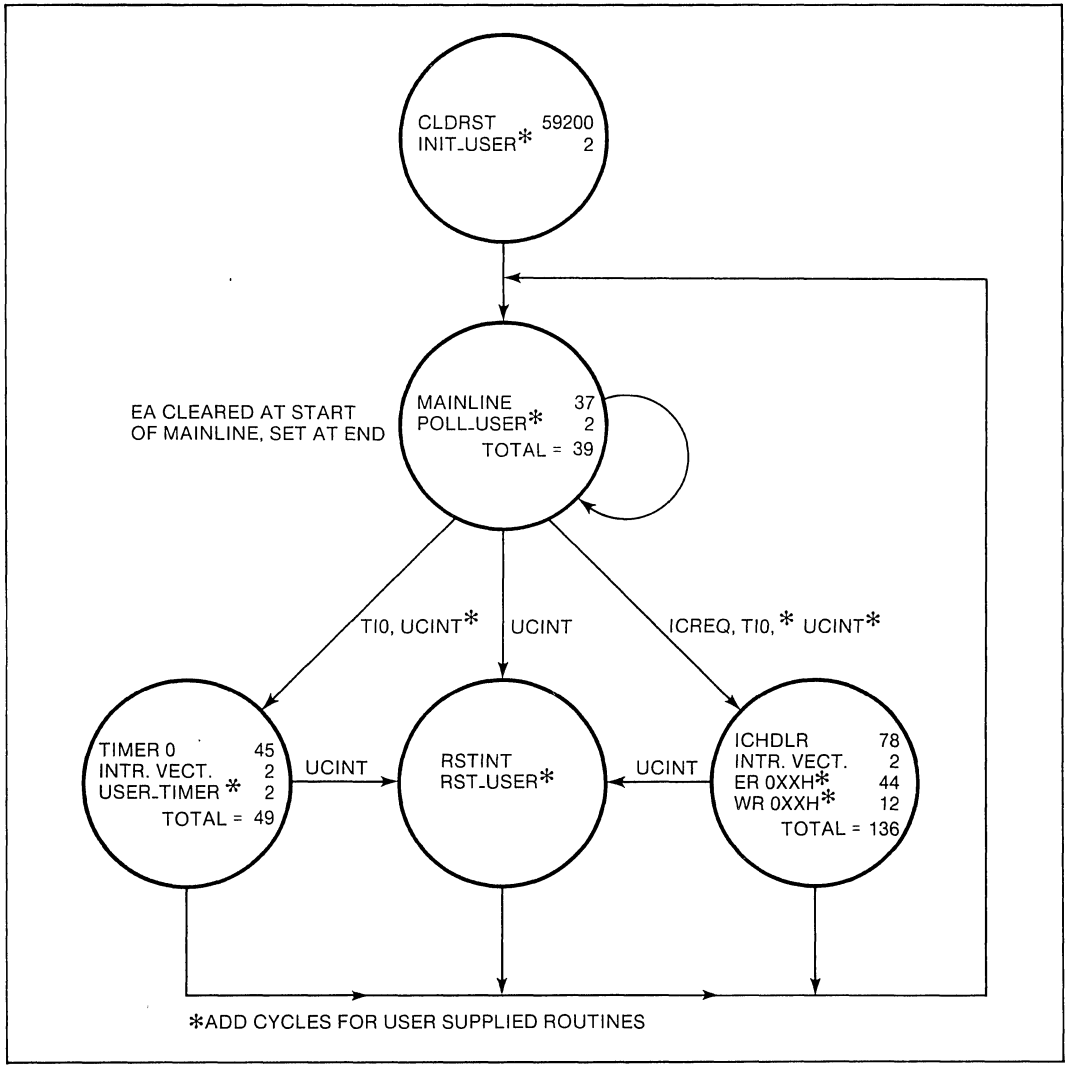


Figure 5-4. Microcontroller Cycle Count of Major Firmware Functions

Min. Interrupt Latency = Mainline = 39 Cycles

Max. Timer 0 Latency = IC Handler + Mainline = 136 + 49 Cycles

The timer 0 interrupt occurs every 1 ms when using the recommended 11-MHz crystal oscillator. However, the timer 0 interrrupt is masked in both the IC handler and the mainline code. Therefore, the worst case timer 0 latency is calculated:

An iPSB timeout will occur if an incoming interconnect request is not completed with in 1 ms after **SC0*** becomes active. Consider the case where a local interconnect request is received just a moment before the iPSB request. The microcon-

troller might be in a timer 0 interrupt routine and would have to pass again through mainline code before servicing the local request. Thus, the worst case iPSB response can be calculated:

Worst Case iPSB Response = Mainline + Timer_0 + mainline + Local IC Access + mainline + PSB Access

For the core firmware without user code, the worst case local firmware response is approximately 485 μs at 11-MHz. This leaves a total of 515 μs for the iPSB interconnect cycle to complete (including MPC and write data delays) before a timeout occurs.

5.8.3 Programming the 8751

One of the three initialization self-checks that the microcontroller performs following a reset is the PROM check sum test. The firmware computes a 16-bit sum of all code bytes from 0 through (top of the PROM memory - 2). The actual check sum is the two's complement of the sum. This value is programmed into the last two bytes of PROM space and given the label CHECKSUM. The check sum label is assigned in the default program file \ICFW\SRC\DFT.MOD and is originally set to address 0FFEH. The address of this label can be modified to accomodate other microcontrollers with larger EPROM size (e.g. an 8752 Microcontroller), as long as the check sum always occupies the last two bytes of code space.

The check sum is initially assembled with a data value of 0000H. After assembling and linking the interconnect firmware, the actual check sum value will be programmed into the PROM in a separate step. The following procedure assumes that an Intel iUP 201 PROM Programmer and iPPS software used. If you are using another brand of PROM programmer, you must guarantee that their check sum algorithm matches the method used here.

STEP 1. Invoke iPPS.

STEP 2. Initalize default base to hex and the file format to 80.
I 80

STEP 3. Set PROM type to 8751.
T 8751

STEP 4. Load data buffer with 0FFH.
L B W 0FFH

STEP 5. Copy object file to buffer. iPPS will display a check sum.
C <file name> T B

STEP 6. Copy check sum displayed by iPPS into address 0FFEH. The low byte of the 16-bit check sum loads to address 0FFEH and the high byte to address 0FFFH.
S 0FFEH
0FFE:(low byte) (high byte)

STEP 7. Save a copy of the complete object file on your hard disk
C B T <file name>

STEP 8. Install a blank 8751 in the programmer and copy the buffer to PROM.
C B T P

At this point you will have a programmed microcontroller, which is ready to install on the prototype board for functional testing. As an alternative, you can load the file created in step 7 above into an ICE 51 In-Cicuit Emulator and test the firmware directly without programming the 8751. This method gives you access to internal variables and tables in addition to hardware control of the microcontroller.

CHAPTER 6
FUNCTIONAL TESTING OF THE INTERCONNECT SUBSYSTEMS

6.1 TESTING THE INTERCONNECT SUBSYSTEM

Assuming you were successful in Chapters 4 and 5, you now have a functional prototype of your interconnect subsystem ready for testing. Proper interconnect operation depends on complex interrelationships between the 8751 Microcontroller, CPU, Message Passing Coprocessor (MPC), and various TTL circuits. Since interconnect forms the basis of higher level diagnostic services, we must have complete confidence that the interconnect subsystem is functioning properly before implementing the design. Evaluation of the interconnect subsystem is facilitated through the use of an interconnect utility program and a systematic approach to design testing.

6.2 POWER-ON AND INITIALIZATION FUNCTIONS

When a prototype board is inserted into the backplane for the first time, there is a distinct possibility that a design error in the reset initialization sequence could cause a bus error or Reset-Not-Complete (RSTNC) condition. Either of these conditions prevents other boards from coming on-line. Fortunately, these events can usually be detected by watching the front panel LEDs cycle through the power-on sequence.

Under normal reset conditions, the majority of MULTIBUS® II boards first illuminate their yellow BIST LED, progress on to illuminate the green LED (run indicator) and leave no other LEDs lit. If the Central Services Module (CSM) has a BUSERR indicator, it will be lit to indicate a bus error due to RSTNC condition. If the bus error persists for more than a few seconds, then a strong likelihood exists that your board has failed to clear the RSTNC* bit. This is confirmed if, after 30 seconds, the CSM BUSERR condition disappears due to the microcontroller clamping reset active and forcing a reset complete indication to the iPSB (unless disabled using the ICE™ compatibility jumper).

6.3 THE INTERCONNECT CONFIGURATION UTILITY

In the \ICFW\ICU subdirectory is an interconnect utility (IC.P28) written in PL/M source code to run on the iRMX™ 86/286 Operating Systems. The executable object file for this utility may be obtained from the iRMX Users Group (iRUG), 5200 N.E. Elam Young Parkway, Hillsboro, OR 97124-9987. If you are not working in one of these programming environments, you will need to adapt this code to your own environment by substituting the appropriate I/O and interconnect system calls.

Before invoking the IC utility on your prototype, you must create the subdirectory /USER/BOARDS and enter into that directory a file for the board. The file name you assign must correspond with the ASCII description contained in the 10 character board ID field of the Interconnect Header Record. Slashes (/) present in the description must be converted to ASCII periods. For example, iSBC386/100 is converted to iSBC386.100. The actual values for each interconnect register are entered into the template plus any user comments you may wish to add for that board. A typical configuration file listing is shown in figure 6-1. The control characters that appear in the file listing are used by the utility to mark the display area and field boundaries.

- The @ symbol is the screen delimiter. As the user pages through the file and encounters this symbol, the utility will prompt the user for input.

- The ; symbol is the register field delimiter. The utility scans for an opening semicolon; then fetches a register value for each character

```
Reg     Function                    Value  RW   Comments
00      Vendor ID Number             ;h h ; ro
02      Board ID Number          ;aaaaaaaaaa; ro
0C      PBA Number, Rev           ;d d d d ; ro
10      Hardware Test Rev #          ;d ; ro
11      Class ID                     ;b ; ro
12      Reserved                 ;b b b b ; ro
16      Program Table Index          ;b ; rw
17      NMI Enable Register          ;b ; rw
18      General Status               ;b ; ro
19      General Control              ;b ; rw
1A      BIST Support Level           ;b ; rw
1B      BIST Data In                 ;b ; rw
1C      BIST Data Out                ;b ; rw
1D      BIST Slave Status            ;b ; rw
1E      BIST Master Status           ;b ; rw
1F      BIST Test ID                 ;b ; rw
@
Reg     Protection Record           Value  RW   Comments
20      Protection Record Type       ;b ; ro
21      Record Length                ;b ; ro
22      Protection Level Reg         ;b ; rw
23      Reserved                     ;b ; ro
@
Reg     Memory Record Type          Value  RW   Comments
24      Memory Record Type           ;b ; ro
25      Record Length                ;b ; ro
26      Memory Size (bits 7..0)      ;b ; ro
27      Memory Size (bits 15..8)     ;b ; ro
28      Memory Control               ;b ; rw
29      Memory Status                ;b ; ro
2A      MM2 ID                       ;b ; ro
@
Reg     iPSB Control Record         Value  RW   Comments
2B      iPSB Control Record Type     ;b ; ro
2C      Record Length                ;b ; ro
2D      iPSB Slot ID Number          ;b ; ro
2E      iPSB Arbitration ID Number   ;b ; rw
2F      iPSB Error Register          ;b ; rw
30      iPSB Control/Status Register ;b ; rw
31      iPSB Diagnostic Register     ;b ; rw
32      Reserved                     ;b ; ro
@
```

Figure 6-1. Configuration File Format for the IC Utility (Sheet 1 of 3)

```
Reg     iPSB Memory Record Type      Value  RW   Comments
33      iPSB Memory Record Type        ;b ; ro
34      Record Length                  ;b ; ro
35      iPSB Start Address (23..16)    ;b ; rw
36      iPSB Start Address (31..24)    ;b ; rw
37      iPSB End Address (23..16)      ;b ; rw
38      iPSB End Address (31..24)      ;b ; rw
39      iPSB Memory Control            ;b ; rw
@
Reg     Local Memory Record Type     Value  RW   Comments
3A      Local Memory Record Type       ;b ; ro
3B      Record Length                  ;b ; ro
3C      Local Start Address (23..16)   ;b ; ro
3D      Local Start Address (31..24)   ;b ; ro
3E      Local End Address (23..16)     ;b ; rw
3F      Local End Address (31..24)     ;b ; ro
40      Local Control Register         ;b ; rw
@
Reg     Memory Parity Record Type    Value  RW   Comments
41      Memory Parity Record Type      ;b ; ro
42      Record Length                  ;b ; ro
43      Parity Control Register        ;b ; rw
44      Parity Status Register         ;b ; ro
45      Bank Status Register           ;b ; ro
46      Error Offset             ;b b b b ; ro
4A      Reserved                       ;b ; ro
@
Reg     Cache Memory Record          Value  RW  Comments
4B      Cache Memory Record Type       ;b ; ro
4C      Record Length                  ;b ; ro
4D      Cache Size (bits 7..0)         ;b ; ro
4E      Cache Size (bits 15..8)        ;b ; ro
4F      Cache Entry Size               ;b ; ro
50      Cache Control                  ;b ; rw
51      Reserved                       ;b ; ro
@
```

Figure 6-1. Configuration File Format for the IC Utility (Sheet 2 of 3)

```
Reg     Firmware Comm Record Type   Value  RW   Comments
52      Firmware Comm Record Type     ;b ; ro
53      Record Length                 ;b ; ro
54      Communications Byte 1         ;b ; rw
55      Communications Byte 2         ;b ; rw
56      Communications Byte 3         ;b ; rw
57      Communications Byte 4         ;b ; rw
58      Communications Byte 5         ;b ; rw
59      Communications Byte 6         ;b ; rw
5A      Communications Byte 7         ;b ; rw
5B      Communications Byte 8         ;b ; rw
5C      Communications Byte 9         ;b ; rw
5D      Communications Byte 10        ;b ; rw
5E      Communications Byte 11        ;b ; rw
5F      Communications Byte 12        ;b ; rw
60      Communications Byte 13        ;b ; rw
61      Communications Byte 14        ;b ; rw
62      Communications Byte 15        ;b ; rw
63      Communications Byte 16        ;b ; rw
@
Reg     Host ID Record Type         Value  RW   Comments
64      Host ID Record Type           ;b ; ro
65      Record Length                 ;b ; ro
66      Host ID (bits 7..0)           ;b ; rw
67      Host ID (bits 15..8)          ;b ; rw
68      Message Address               ;b ; rw
69      Reserved                      ;b ; ro
@
Reg     Serial Comm Record Type     Value  RW   Comments
6A      Serial Comm Record Type       ;b ; ro
6B      Record Length                 ;b ; ro
6C      Serial Data In                ;b ; ro
6D      Serial Data Out               ;b ; rw
6E      Serial Port Status            ;b ; ro
6F      Serial Interrupt Enable       ;b ; rw
70      Serial Port Options           ;b ; rw
71      Reserved                      ;b ; ro
@
Reg     386/100 Specific Record     Value  RW   Comments
72      386/100 Spec Record Type      ;b ; ro
73      Record Length                 ;b ; ro
74      On-board Control Register     ;b ; rw
@
Reg     End Of Template Record      Value  RW   Comments
75      EOT Record Type               ;b ; ro
@
```

Figure 6-1. Configuration File Format for the IC Utility (Sheet 3 of 3)

found until encountering the closing semicolon. Legal character type descriptors are:

b (binary)
d (decimal)
h (hexadecimal)
a (ASCII)

The IC utility enables you to identify, examine, and modify the configuration of a system board from the comfort of your console (no jumpers to move!). The utility locates the board ID character fields in the Interconnect Header Record of the board in the cardslot you select to configure. It then searches the subdirectory /USER/BOARDS for the file having a file name that matches the board ID. Specific interconnect registers may be modified (provided access rights and value checking are valid), however, an attempt to enter illegal values into an interconnect register are ignored and the existing register content is unaffected. Entries are displayed as they are made. Once invoked, the IC utility prompts the user for input and you will find it very easy to use. A sample screen output from this utility is shown in figure 6-2.

6.4 A TEST METHODOLOGY FOR INTERCONNECT FUNCTIONS

Given a functional prototype board and an interconnect utility such as the one described above, we now use these tools to verify correct operation of the interconnect subsystem. One might assume that this process would be quite simple and straight forward — and usually, it is. But we are now looking for more subtle conditions such as timing related problems and secondary affects on interconnect register content. While every board is different, a test methodology that looks for all possible contingencies is an important step in design verification. The following checklist is an example of what to look for during evaluation testing of interconnect using a prototype board:

Look for initialization conditions:

Do the correct default register values appear after cold-start?

Do the front panel LEDs work properly?

Is RSTNC handled correctly? (also test RSTNC recovery by disabling the CPU).

Look for dual-port operation on both local and system buses:

Can the registers be accessed from off-board as well as on-board?

Do the function records all have correct record type values and byte counts?

Are the read/write privileges working for both on-board and off-board references?

Is the protection record function working properly?

Look for proper register content and function:

Do all the bit level functions in control registers work as expected?

Are all the status registers reporting correctly?

Look for secondary effects:

Changes to memory addressing registers are only allowed when memory is disabled.

Look for iPSB timeout conditions:

Saturate the local bus with interconnect I/O commands in a tight loop; then attempt an interconnect operation from off-board.

Once you have accomplished this type of testing methodology for each register on the prototype board, you can have reasonable confidence that all is well with the interconnect subsystem. Now you can proceed forward with the development of device drivers and extended diagnostics. If any changes are made to the microcontroller after its initial release, be sure to update the revision control field in the header record so users can identify which version of the firmware they hold.

```
                INTERCONNECT CONFIGURATION UTILITY

Commencing Board Search Routine:  .........
Board Search Complete.  The following boards were found:

Slot#  Vendor ID          Board ID    PBA #         Class
00     Intel Corporation  CSM/001     147304-0003   Central Services Module
01     Intel Corporation  186/410     000000-0002   16-bit Communications Board
02*    Intel Corporation  386/100     000000-0002   32-bit Processor Board
03
04     Intel Corporation  MEM/310     000000-0001   Memory Board
05     Intel Corporation  286/100A    000000-0001   16-bit Processor Board
06
07     Intel Corporation  186/224A    000000-0001   16-bit Peripheral Controller
08

Options: <Configure,Slot#>, <Memory> or <Quit> Type First Letter):c2

Configuration File Attached - File: /user/boards/386.100

Reg    Function                    Value  RW   Comments
00     Vendor ID Number             0100; ro
02     Board ID Number            386/100; ro
0C     PBA Number, Rev           00000000; ro
10     Hardware Test Rev #            02; ro
11     Class ID                       F1; ro
12     Reserved                  00000080; ro
16     Program Table Index            00; rw
17     NMI Enable Register            04; rw
18     General Status                 80; ro
19     General Control                00; rw
1A     BIST Support Level             00; rw
1B     BIST Data In                   00; rw
1C     BIST Data Out                  00; rw
1D     BIST Slave Status              10; rw
1E     BIST Master Status             30; rw
1F     BIST Test ID                   10; rw

Would you like to make any changes?

Enter <reg_num,new_value_in_hex>,<Quit>,<CR> or <Up>:
```

Figure 6-2. Interconnect Utility Screens (Sheet 1 of 2)

```
Reg    Protection Record          Value  RW       Comments
20     Protection Record Type       0B; ro
21     Record Length                02; ro
22     Protection Level Reg         00; rw
23     Reserved                     00; ro

Reg    Memory Record Type         Value  RW       Comments
24     Memory Record Type           01; ro
25     Record Length                05; ro
26     Memory Size (bits 7..0)      3F; ro
27     Memory Size (bits 15..8)     00; ro
28     Memory Control               01; rw
29     Memory Status                A1; ro
2A     MM2 ID                       22; ro

Would you like to make any changes?

Enter <reg_num,new_value_in_hex>,<Quit>,<CR> or <Up>:

Reg    iPSB Control Record        Value  RW       Comments
2B     iPSB Control Record Type     06; ro
2C     Record Length                06; ro
2D     iPSB Slot ID Number          10; ro
2E     iPSB Arbitration ID Number   E8; rw
2F     iPSB Error Register          88; rw
30     iPSB Control/Status Register 18; rw
31     iPSB Diagnostic Register     00; rw
32     Reserved                     00; ro

Would you like to make any changes?

Enter <reg_num,new_value_in_hex>,<Quit>,<CR> or <Up>:q
```

Figure 6-2. Interconnect Utility Screens (Sheet 2 of 2)

6.5 SUMMARY AND CONCLUSION

MULTIBUS II is a system architecture composed of standardized hardware and software modules. Having successfully implemented the interconnect subsystem on your board, you are ensured of the functional compatibility and interoperability of that board with other industry-standard MULTIBUS II products that use interconnect.

If you found this guide useful, you may also be interested in other design guides available from Intel. A list of documents available at the time of this guide's publication are listed in Appendix A. As the MULTIBUS II system architecture matures and proliferates, we look forward to supplying additional design guides that encourage the production of compatible hardware and software products.

APPENDIX A
BIBLIOGRAPHY OF RELATED READING

Interconnect Architectural Specification - Intel Order Number 149299-002.

MPC User's Guide - Intel Order Number 176526-001.

IEEE 1296 Specification (High Performance Synchronous 32-bit Bus Standard)

iSBC™ 386/116 Hardware Reference Manual - Intel Order Number 451833-001.

BUSCON Paper: An Architecture for Initializing Multibus® II Multiprocessor Systems. Stephen Rogers, October 1987.

Computer Technology Review Quarterly: Interconnect Simplifies System Configuration. Roger Finger, September 1987.

Intel Application Note AP-70, Using the Intel MCS® 51 Boolean Processing Capabilities. John Wharton, 1980.

Embedded Controller Handbook 8751 Data Sheet - Intel Order Number 210918-005.

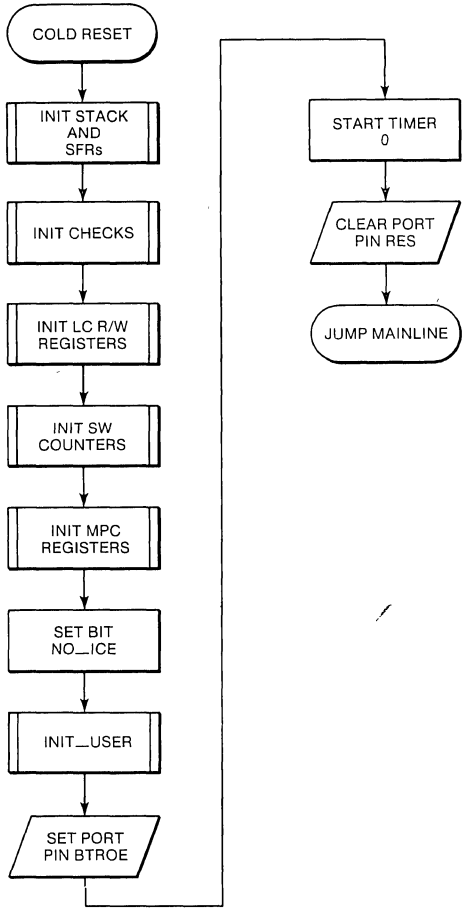
iUP 200/201A Programmer User's Guide - Intel Order Number 166608-001.

MCS™ Macro Assembler User's Guide for DOS Systems - Intel Order Number 122752-001.

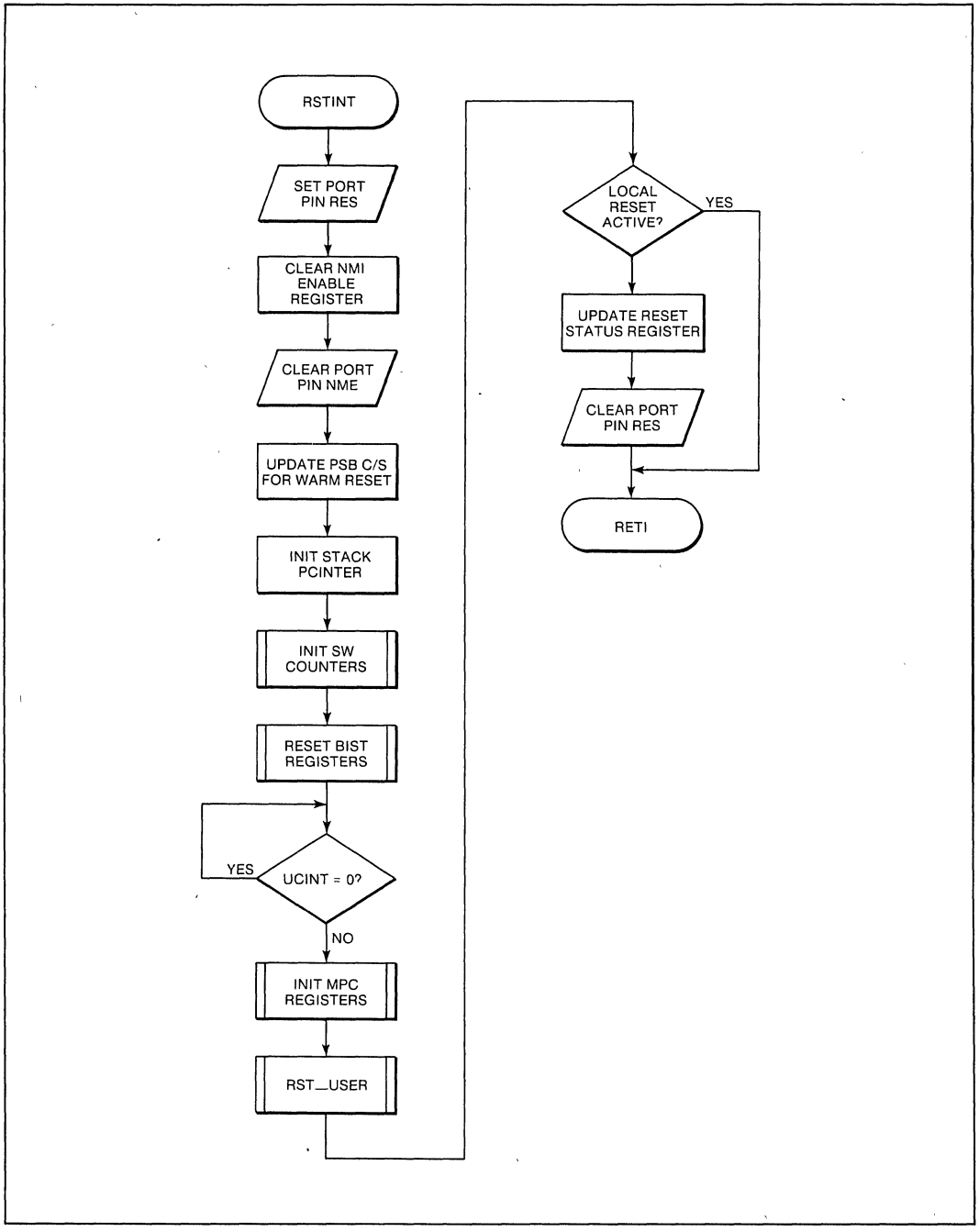
MCS™ 51 Utilities User's Guide for DOS Systems - Intel Order Number 122747-001.

Intel Application Note AP-422, Designing a Central Services Module for MULTIBUS®II. Jory Radke, 1987.

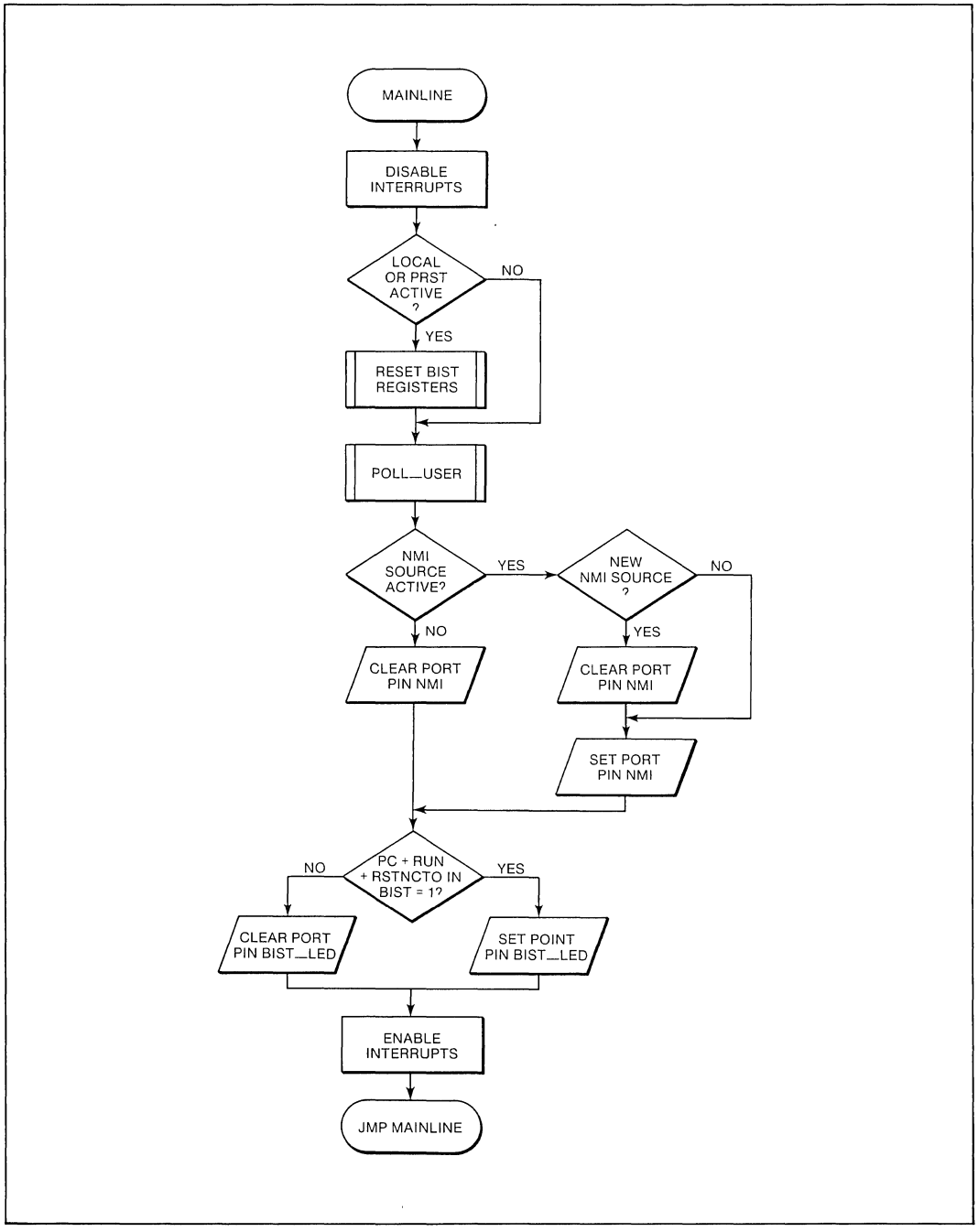
APPENDIX B
FLOWCHARTS FOR IC CORE FIRMWARE



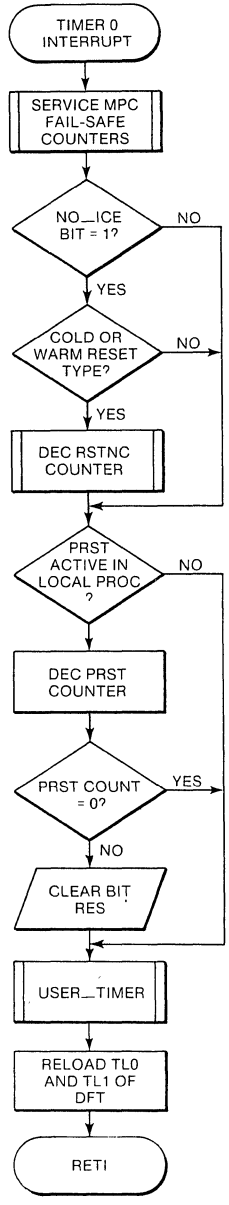
B.1. Cold Reset Routine



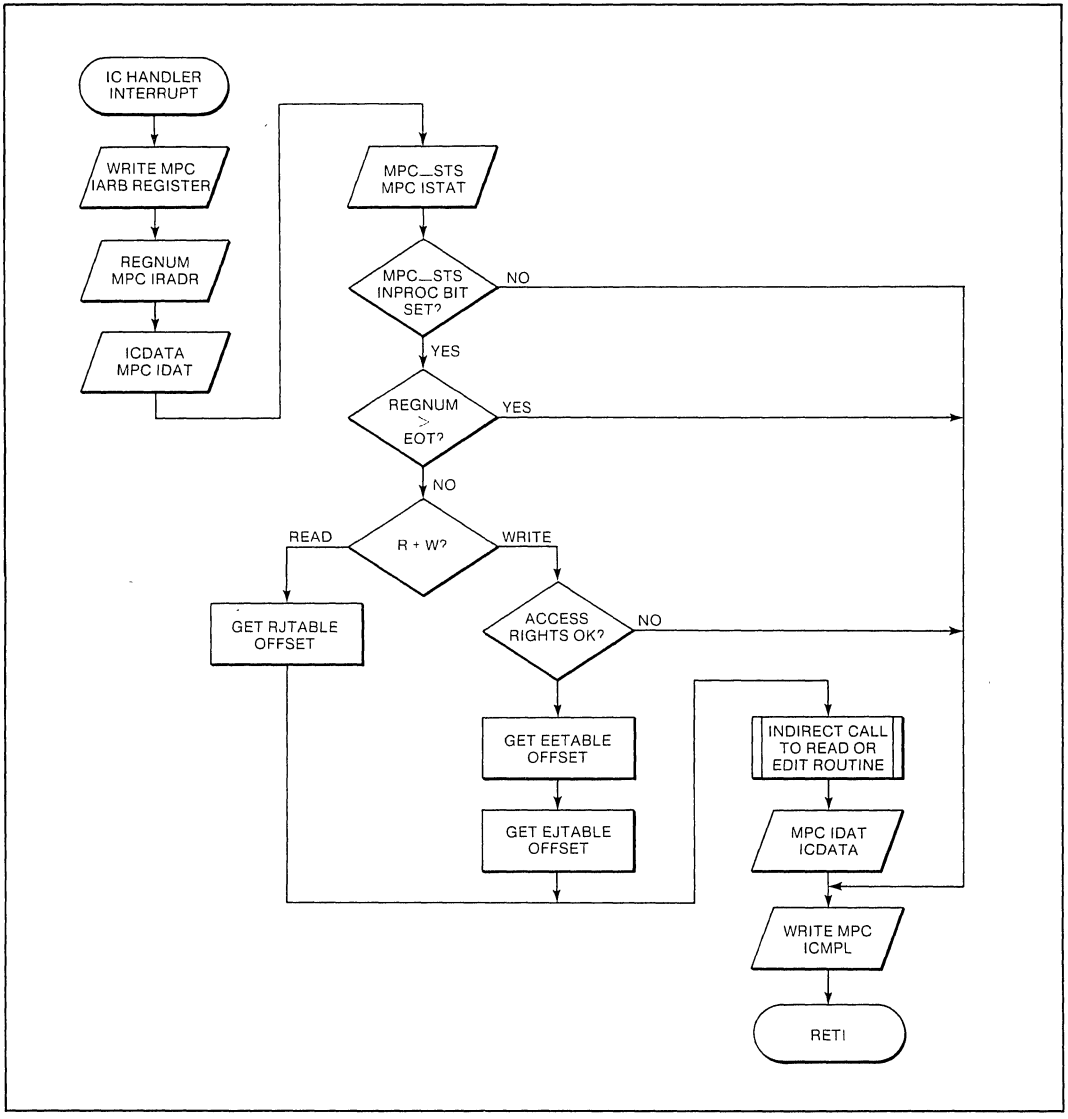
B.2. Reset Interrupt Routine



B.3. Mainline Program Flow



B.4. Timer 0 Interrupt Routine



B.5. IC Handler Interrupt Routine

APPENDIX C
READ, WRITE, AND EDIT ROUTINES

```
;*****************************************************
;
;  NAME:          IC_TO_MPC
;
;  PURPOSE:       CORRELATE IC REGISTER NUMBER TO MPC
;                 REGISTER NUMBER
;
;  DESCRIPTION:   CORRELATES IC PSB C/S REGISTERS WITH
;                 THE PROPER MPC REGISTER NUMBER,
;                 STARTING WITH THE PSB SLOT ID
;                 REGISTER.
;
;  UPDATE HISTORY:
;
;*****************************************************
;
IC_TO_MPC:  DB      SID         ;PSB SLOT ID
            DB      AID         ;PSB ARB ID
            DB      RERR        ;PSB ERROR
            DB      00H         ;DUMMY
            DB      DIAG        ;PSB DIAGNOSTIC
$EJECT
```

C.1. IC_TO_MPC Correlation

```
;*****************************************************
;
;  NAME:          RD00H
;
;  PURPOSE:       READ DEFAULT VALUE FROM DFTABLE
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES REGNUM TO INDEX INTO DFTABLE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        REGNUM
;
;  DESTROYS:      ACC, DPTR
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*****************************************************

RD00H:  %LOOKUP(#DFTABLE,REGNUM) ;GET DEFAULT FM TABLE
        MOV     ICDATA,A        ;RET DEFAULT IN ICDATA
        RET                      ;RETURN TO IC HANDLER
$EJECT
```

C.2. RD00H Read DFTABLE

```
;*****************************************************
;
;  NAME:          RD01H
;
;  PURPOSE:       READ RWTABLE VALUE
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES REGNUM TO INDEX INTO RATABLE.
;                 USES RWTABLE INDEX PORTION OF RA BYTE
;                 TO INDEX INTO RWTABLE TO GET REGISTER
;                 VALUE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        REGNUM
;
;  DESTROYS:      ACC, ADDR, DPTR
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*****************************************************

RD01H:  %LOOKUP(#RATABLE,REGNUM)  ;GET RAM/ACCESS BYTE
        CLR     ACC.GAR ;MASK OFF GLOBAL ACCESS RIGHTS
        ADD     A,#RWTABLE ;ADD TBL ADDR TO RAM OFFSET
        MOV     ADDR,A                  ;SAVE RAM ADDR
        MOV     A,@ADDR                 ;READ RAM VALUE
        MOV     ICDATA,A        ;RETURN VALUE IN ICDATA
        RET                       ;RETURN TO IC HANDLER

$EJECT
```

C.3. RD01H Read RWTABLE

```
;*******************************************************
;
;  NAME:          RD02H
;
;  PURPOSE:       READ MPC REGISTER FOR PSB
;                 CONTROL/STATUS REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES LOOKUP TABLE IC_TO_MPC TO XREF
;                 IC ADDRESS TO MPC REGISTER NUMBER,
;                 READS MPC REG AND RETURNS DATA IN
;                 ICDATA.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      ACC, DPTR, MPC_RNUM
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*******************************************************

RD02H:  MOV     A,REGNUM    ;GET REG NO. BEING ACCESSED
        SUBB    A,#PSB_CTL_OFF+5     ;IC_TO_MPC STARTS
                                     ;WITH SLOT ID REG
        %LOOKUP(#IC_TO_MPC,A)  ;GET CORRES MPC REG NO.
        %RD_MPC(ICDATA,A)              ;READ MPC REG
        RET                      ;RETURN TO IC HANDLER

$EJECT
```

C.4. RD02H Read MPC Register

```
;*******************************************************
;
;  NAME:          RD03H
;
;  PURPOSE:       READ ROUTINE FOR GENERAL STATUS
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES LOOKUP TABLE IC_TO_MPC TO XREF
;                 IC ADDRESS TO MPC REGISTER NUMBER,
;                 READS MPC REG AND RETURNS DATA IN
;                 ICDATA.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      ACC, DPTR, MPC_RNUM
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*******************************************************

RD03H:  MOV     ICDATA,GEN_STS              ;READ RAM
        %MOVBIT(ICDATA.PFI,DCLO)  ;GET PWR FAIL STATUS
        RET                       ;RETURN TO IC HANDLER

$EJECT
```

C.5. RD03H Read General Status Register

```
;*******************************************************

;  NAME:          RD04H
;
;  PURPOSE:       READ ROUTINE FOR PROTECTION LEVEL
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   READS THE BIT UACRS AND PUTS IT IN
;                 ICDATA
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*******************************************************

RD04H:  MOV     ICDATA,#CLEAR              ;ICDATA = 0
        %MOVBIT(ICDATA.ACRS,UACRS)           ;GET BIT
        RET                        ;RETURN TO IC_HANDLER

$EJECT
```

C.6. RD04H Read Protection Level Register

```
;*******************************************************
;
;  NAME:          RD05H
;
;  PURPOSE:       READ ROUTINE FOR PSB CONTROL/STATUS
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   COMBINES VALUES OF MPC GENERAL
;                 PARAMETER REGISTER WITH VALUE
;                 FROM RWTABLE FOR PSB_CS TO GET
;                 COMPLETE REGISTER VALUE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      ACC, MPC_RNUM, CARRY FLAG
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*******************************************************

RD05H:  %RD_MPC(A,#GEN_PAR)   ;READ MPC GENL PARAM REG
        MOV     ICDATA,PSB_CS                ;READ RAM
        %MOVBIT(ICDATA.HPRO,ACC.PRY)  ;MOV MPC BITS TO
                                              ;ICDATA
        %MOVBIT(ICDATA.RSTNC,ACC.RSTNCIN)
        RET                       ;RETURN TO IC_HANDLER

$EJECT
```

C.7. RD05H Read Control/Status Register

```
;*******************************************************
;
;  NAME:          RD06H
;
;  PURPOSE:       READ ROUTINE FOR LOCAL PROC CONTROL
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   IF PROC RESET AND RESET STATUS =
;                 PRST, THEN SETS PRST = TRUE, ELSE
;                 PRST = FALSE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:  5-14-87 JR UPDATED FOR IDX
;
;*******************************************************

RD06H:  MOV    ICDATA,#CLEAR                ;ICDATA = 0
        JNB    RES,END_R6  ;IF NOT PROC RESET THEN DONE
        MOV    A,RST_STS       ;ELSE GET LAST RESET TYPE
        CJNE   A,#PRST_TYPE,END_R6    ;IF NOT PRST THEN
                                                ;DONE
        SETB   PRST               ;ELSE SET PRST = TRUE
END_R6: RET                       ;RETURN TO IC HANDLER

$EJECT
```

C.8. RD06H Read Local Processor Control Register

```
;*******************************************************
;
;  NAME:          RD07H
;
;  PURPOSE:       READ ROUTINE FOR LOCAL STATUS
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   IF RESET STATUS = PRST TYPE, THEN
;                 PRST STATUS = TRUE, ELSE PRST
;                 STATUS = FALSE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:  5-14-87 JR ADDED FOR IDX
;
;*******************************************************

RD07H:  MOV     ICDATA,#CLEAR              ;ICDATA = 0.
        MOV     A,RST_STS          ;GET LAST RESET TYPE
        CJNE    A,#PRST_TYPE,END_R7  ;IF NOT PRST THEN
                                                ;DONE
        SETB    ICDATA.PRST_STS  ;ELSE PRST_STS = TRUE
END_R7: RET                      ;RETURN TO IC HANDLER

$EJECT
```

C.9. RD07H Read Local Status Register

```
;*******************************************************
;
;  NAME:          WR00H
;
;  PURPOSE:       WRITE TO A RAM BASED IC REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINES
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES REGNUM TO INDEX INTO RATABLE.
;                 USES RWTABLE INDEX PORTION OF RA BYTE
;                 TO INDEX INTO RWTABLE, WHERE ICDATA
;                 IS THEN WRITTEN TO.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        REGNUM
;
;  DESTROYS:      ACC, DPTR, ADDR
;
;  RETURNS:       NOTHING, RETURNS TO ICHANDLER
;
;  UPDATE HISTORY:
;
;*******************************************************

WR00H:  %LOOKUP(#RATABLE,REGNUM)       ;GET RAM/ACCESS
                                         ;BYTE FOR REG
        CLR     ACC.GAR          ;EXTRACT RWTABLE INDEX
        ADD     A,#RWTABLE  ;ADD TABLE ADDR TO GET RAM
                                                 ;ADDR
        MOV     ADDR,A  ;PUT ADDR IN INDIRECT ADDR REG
        MOV     @ADDR,ICDATA            ;WR RWTABLE REG
        RET                        ;RETURN TO IC HANDLER

$EJECT
```

C.10. WR00H Write RAM Based IC Register

```
;*******************************************************
;
;  NAME:          WR01H
;
;  PURPOSE:       WRITE ROUTINE FOR MPC BASED IC
;                 REGISTERS
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINES
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES LOOKUP TABLE IC_TO_MPC TO XREF
;                 IC ADDRESS TO MPC REGISTER NUMBER,
;                 WRITES MPC REG WITH DATA IN ICDATA.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        ICDATA
;
;  DESTROYS:      DPTR, MPC_RNUM
;
;  RETURNS:       NOTHING, RETURNS TO ICHDLR
;
;  UPDATE HISTORY:
;
;*******************************************************

WR01H:  MOV     A,REGNUM        ;GET REQUESTED REG NO.
        SUBB    A,#PSB_CTL_OFF+5     ;IC_TO_MPC STARTS
                                ;WITH PSB SLOT ID REG
        %LOOKUP(#IC_TO_MPC,A)  ;GET CORRES MPC REG NO.
        %WRT_MPC(A,ICDATA)              ;WR MPC REG
        RET                     ;RETURN TO IC HANDLER

$EJECT
```

C.11. WR01H Write MPC Based IC Register

```
;*****************************************************
;
;  NAME:          WR02H
;
;  PURPOSE:       WRITE ROUTINE FOR GENERAL CONTROL
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   PROC RESET = LOCAL RESET.  IF LOCAL
;                 RESET, THEN UPDATE RESET STATUS REG.
;                 UPDATES GEN_CTRL IN RWTABLE BY
;                 JUMPING TO WR00H.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        ICDATA
;
;  DESTROYS:      NOTHING
;
;  RETURNS:       NOTHING, JUMPS TO WR00H
;
;  UPDATE HISTORY:  5-14-87 JR UPDATED FOR IDX
;
;*****************************************************

WR02H:  %MOVBIT(RES,ICDATA.LRST)  ;UPDATE RST PORT PIN
        JNC     END_W2    ;IF NO LOCAL RESET THEN DONE
        MOV     RST_STS,#LRST_TYPE  ;ELSE UPDATE RESET
                                                ;STATUS
END_W2: SJMP    WR00H

$EJECT
```

C.12. WR02H Write General Control Register

```
;*******************************************************
;
;  NAME:          WR03H
;
;  PURPOSE:       WRITE ROUTINE FOR PROTECTION LEVEL
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   SETS DYNAMIC ACCESS RIGHTS AS
;                 REQUESTED
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        ICDATA
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       NOTHING, RETURNS TO ICHDLR
;
;  UPDATE HISTORY:
;
;*******************************************************

WR03H:  %MOVBIT(UACRS,ICDATA.ACRS);UACRS = ICDATA.ACRS
        RET                        ;RETURN TO IC HANDLER

$EJECT
```

C.13. WR03H Write Protection Level Register

```
;*****************************************************
;
;  NAME:          WR04H
;
;  PURPOSE:       WRITE ROUTINE FOR PSB CONTROL/STATUS
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   SINCE RST TYPE BITS MUST = 0 WHEN
;                 WRITING THIS REGISTER, RSTNCOUT IS
;                 CLEARED IN THE MPC GENERAL PARAM REG.
;                 ALSO,  THE PRY BIT IN THE GEN PARAM
;                 REG IS MODIFIED ACCORDING TO ICDATA
;                 BIT HPRO.  SINCE THE GEN PARAM REG
;                 SUPPORTS OTHER FUNCTIONS, A
;                 READ/MODIFY/WRITE OPERATION IS USED.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        ICDATA
;
;  DESTROYS:      ACC, CARRY FLAG, MPC_RNUM
;
;  RETURNS:       NOTHING
;
;  UPDATE HISTORY:
;
;*****************************************************

WR04H:  MOV   A,PSB_CS ;GET VALUE OF PSB CONT/STAT REG
        %MOVBIT(ICDATA.RSTH0,ACC.RSTH0)    ;SAVE RESET
                                        ;TYPE HISTORY BITS
        %MOVBIT(ICDATA.RSTH1,ACC.RSTH1)
        MOV   PSB_CS,ICDATA              ;UPDATE RWTABLE
        %RD_MPC(A,#GEN_PAR)             ;READ MPC GEN_PAR
        CLR   ACC.RSTNCOUT      ;QUIT DRVG RSTNC TO BUS
        %MOVBIT(ACC.PRY,ICDATA.HPRO)   ;UPDATE PRY BIT
        %WRT_MPC(#GEN_PAR,A)           ;WR MODIFIED DATA
                                        ;BACK TO GEN_PAR
        RET                          ;RETURN TO IC HANDLER

$EJECT
```

C.14. WR04H Write PSB Control/Status Register

```
;*******************************************************
;
;  NAME:          WR05H
;
;  PURPOSE:       WRITE ROUTINE FOR PSB DIAGNOSTIC
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   UPDATES PSB_DIAG REG IN RWTABLE.  IF
;                 LOOPBACK MODE SELECTED, THEN TURNS
;                 OFF BUS TRANSCEIVERS AND WRITES MPC
;                 DIAG REG, ELSE WRITES MPC DIAG REG
;                 AND ENABLES TRANSCEIVERS.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  DESTROYS:      ACC, MPC_RNUM
;
;  INPUTS:        ICDATA
;
;  RETURNS:       NOTHING, RETURNS TO ICHDLR
;
;  UPDATE HISTORY:
;
;*******************************************************

WR05H:  MOV    PSB_DIAG,ICDATA          ;UPDATE RWTABLE
        JB     ICDATA.LBACK,BUSOFF        ;IF LOOPBACK
                                  ;SELECTED, THEN JUMP
        %WRT_MPC(#DIAG,ICDATA)       ;ELSE WR MPC REG
        SETB   BTROE                ;ENABLE BUS XCVRS
        JMP    END_W5                     ;DO NO MORE
BUSOFF: CLR    BTROE                   ;DISABLE XCVRS
        %WRT_MPC(#DIAG,ICDATA)           ;WR MPC REG
END_W5: RET                       ;RETURN TO IC HANDLER

$EJECT
```

C.15. WR05H Write PSB Diagnostic Register

```
;*****************************************************
;
;  NAME:          WR06H
;
;  PURPOSE:       WRITE ROUTINE FOR LOCAL PROC CONTROL
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   IF ICDATA BIT PRST SET, ASSERT RESET
;                 TO PROC AND UPDATE RESET STATUS REG.
;
;  INPUTS:        ICDATA
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       NOTHING
;
;  UPDATE HISTORY:  5-14-87 JR ADDED FOR IDX
;
;*****************************************************

WR06H:  JNB    ICDATA.PRST,END_W6;IF NO RESET THEN DONE
        SETB    RES           ;ELSE ASSERT RESET TO PROC
        MOV    RST_STS,#PRST_TYPE          ;UPDATE RESET
                                             ;STATUS REG
        MOV    PRST_CNTR,#PRST_CNT    ;LOAD PRST COUNTER
END_W6: RET                         ;RETURN TO IC HANDLER

$EJECT
```

C.16. WR06H Write Local Processor Control Register

```
;*****************************************************
;
;  MACRO NAME:    GET_EEC (GET EDIT ENTRY CONSTANT)
;
;  SYNTAX:        GET_EEC
;
;  PURPOSE:       INDEX INTO EETABLE TO GET NEXT EDIT
;                 ENTRY CONSTANT
;
;  DESCRIPTION:   EET_OFF IS INCREMENTED AND THE CODE
;                 BYTE AT EET_OFF FROM DPTR (ASSUMED TO
;                 BE POINTING TO EETABLE) IS MOVED TO
;                 THE ACCUMULATOR.
;
;  INPUTS:        NONE, HOWEVER DPTR ASSUMED = EETABLE
;                 (INTR ROUTINES USING DPTR SHOULD
;                 PUSH/POP).
;
;  DESTROYS:      ACC
;
;  RETURNS:       NEXT EET CONSTANT FROM LIST IN ACC
;
;  UPDATE HISTORY:
;
;*****************************************************

%*DEFINE(GET_EEC)(
$SAVE NOGEN
        INC     EET_OFF
        MOV     A,EET_OFF
        MOVC    A,@A+DPTR
$RESTORE
)
$EJECT
```

C.17. Macro GET_EEC

```
;*****************************************************
;
;  NAME:          VALERR
;
;  PURPOSE:       ROUTINE FOR REPORTING IC WRITE VALUE
;                 ERRORS
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINES UPON
;                 VALUE ERROR DETECTION
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   CHECKS FOR LOCAL OR GLOBAL ACCESS AND
;                 UPDATES GENERAL STATUS REGISTER
;                 ACCORDINGLY WITH VALUE ERROR STATUS.
;                 RETURNS TO ICHDLR.
;
;  REG BANK:      ASSUMES 0, SELECTS NONE
;
;  INPUTS:        MPC_STS
;
;  DESTROYS:      NOTHING
;
;  RETURNS:       GENERAL STATUS REG, UPDATED FOR VALUE
;                 ERROR, TO ICHDLR
;
;  UPDATE HISTORY:
;
;*****************************************************

VALERR: JB     MPC_STS.PORT,LCVLER     ;IF LOCAL ACCESS
                                       ;THEN JUMP, ELSE
        ORL    GEN_STS,#GBL_VL_ERR  ;SET GENERAL STATUS
                                    ;= GLOBAL VALUE ERROR
        SJMP   ENDVAL                 ;RESTORE AND RETURN
LCVLER: ORL    GEN_STS,#LCL_VL_ERR  ;SET GENERAL STATUS
                                    ;= LOCAL VALUE ERROR
ENDVAL: RET                         ;RETURN TO IC HANDLER

$EJECT
```

C.18. Report IC Write Value Error

```
;*******************************************************
;
;  NAME:          ER00H
;
;  PURPOSE:       GENERATE LOCAL STATIC ACCESS RIGHTS
;                 ERROR
;
;  CALLED BY:     JUMPED TO BY SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   EDIT VECTOR 0 GETS EDIT ENTRY 0,
;                 WHICH IS USED FOR LOCAL READ ONLY
;                 REGISTERS.  ONCE HERE, THE GENERAL
;                 STATUS REG IS UPDATED FOR LOCAL
;                 STATIC ERROR.  THE ACCESS IS THEN
;                 TERMINATED.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NOTHING
;
;  DESTROYS:      NOTHING
;
;  RETURNS:       NOTHING, RETURNS TO ICHDLR
;
;  UPDATE HISTORY:
;
;*******************************************************

ER00H:  ORL    GEN_STS,#LCL_ST_ERR          ;UPDATE GSR
        RET    ;BACK TO IC HANDLER TO TERMINATE ACCESS

$EJECT
```

C.19. ER00H Generate Local Static Access Rights Error

```
;*******************************************************
;
;  NAME:          ER01H
;
;  PURPOSE:       CHECK ALL RFU BITS = 0
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   INDEXES INTO THE EDIT ENTRY TABLE TWO
;                 TIMES, ONCE FOR THE EDIT CONSTANT AND
;                 THEN A SECOND TIME FOR THE OFFSET TO
;                 THE WRITE ROUTINE.  ANDs CONSTANT
;                 WITH ICDATA.  IF RESULT > 0, THEN NOT
;                 ALL RFU = 0,  JUMPS TO VALUE ERROR
;                 ROUTINE, ELSE JUMPS INDIRECTLY TO
;                 WRITE ROUTINE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        EET_OFF,ICDATA
;
;  DESTROYS:      ACC, DPTR, EET_OFF
;
;  RETURNS:       NOTHING, JUMPS TO VALERR, ELSE WRITE
;                 ROUTINE
;
;  UPDATE HISTORY:
;
;*******************************************************

ER01H:  INC    EET_OFF          ;INC INDEX INTO EETABLE
        %LOOKUP(#EETABLE,EET_OFF)       ;GET CONSTANT
        ANL    A,ICDATA         ;CHECK VALUE OF ICDATA
        JNZ    ERR_1             ;JMP IF ILLEGAL VALUE
        %GET_EEC         ;GET WR ROUTINE OFFSET FM EET
        MOV    DPTR,#WRLBL   ;GET WR ROUTINE LABEL ADDR
        JMP    @A+DPTR              ;JMP TO WR ROUTINE
ERR_1:  AJMP   VALERR                ;REPORT VALUE ERR
                                ;(TOO FAR FOR REL JMP)

$EJECT
```

C.20. ER01H Check for RFU Bits = 0

```
;*******************************************************
;
;  NAME:          ER02H
;
;  PURPOSE:       CONST_1 <= ICDATA <= CONST_2
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   ICDATA BANDPASS FILTER.  IF VALUE OF
;                 ICDATA IS WITHIN THE RANGE SPECIFIED
;                 BY THE TWO CONSTANTS FROM THE EET,
;                 THEN THE WRITE ROUTINE OFFSET FROM
;                 EET IS JUMPED TO, ELSE A VALUE ERROR
;                 IS REPORTED IN THE GENERAL STATUS
;                 REG.  THUS EET MUST INCLUDE THREE
;                 BYTES:  CONST_1:  SMALLEST LEGAL
;                 VALUE FOR ICDATA, CONST_2:  LARGEST
;                 LEGAL VALUE FOR ICDATA, AND AN WRITE
;                 ROUTINE OFFSET
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        EET_OFF, ICDATA
;
;  DESTROYS:      ACC, DPTR, EET_OFF, TEMP
;
;  RETURNS:       NOTHING, JUMPS TO WRITE ROUTINE, ELSE
;                 VALERR
;
;  UPDATE HISTORY:
;
;*******************************************************
```

C.21. ER02H ICDATA Between Two Values

```
ER02H:  INC    EET_OFF              ;EET_OFF = EET_OFF+1
        %LOOKUP(#EETABLE,EET_OFF)          ;GET CONST_1
        MOV    TEMP,A                     ;STORE CONST_1
        MOV    A,ICDATA         ;GET VALUE BEING WRITTEN
        %SUB(TEMP)               ;SUB CONST_1 FM ICDATA
        JC     ERR_2           ;IF ICDATA < CONST_1 THEN
                                              ;VALUE ERR
        %GET_EEC                           ;GET CONST_2
        %SUB(ICDATA)             ;SUB ICDATA FM CONST_2
        JC     ERR_2           ;IF ICDATA > CONST_2 THEN
                                              ;VALUE ERR
        %GET_EEC                 ;GET WR ROUTINE OFFSET
        MOV    DPTR,#WRLBL    ;POINT TO WR ROUTINE LABEL
        JMP    @A+DPTR       ;JMP INDIRECT TO WR ROUTINE
ERR_2:  AJMP   VALERR          ;JMP TO VALUE ERR ROUTINE

$EJECT
```

C.21. ER02H ICDATA Between Two Values (Continued)

```
;*****************************************************
;
;  NAME:          ER03H
;
;  PURPOSE:        EDIT ROUTINE FOR PSB ARB ID REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   CHECKS ALL RFUs = 0, THEN CHECKS FOR
;                 LEGAL ARB ID BY COUNTING THE BIT
;                 TRANSITIONS IN ICDATA.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      ACC, TEMP
;
;  RETURNS:       JUMPS DIRECTLY TO WR01H, ELSE VALERR
;
;  UPDATE HISTORY:
;
;*****************************************************

ER03H:  MOV   A,#PSB_ARB_MSK              ;GET RFU MASK
        ANL   A,ICDATA                ;MASK ALL BUT RFUs
        JNZ   ERR_3    ;IF NOT ALL RFUs = 0 THEN ERROR
        MOV   TEMP,#TRANS_CNT          ;LOAD TRANS CNTR
        MOV   A,ICDATA         ;GET VALUE BEING WRITTEN
        CLR   C
        RLC   A                          ;SEE IF MSB SET
        JNC   ID_TR               ;IF NO CARRY, THEN OK
        SETB  ACC.3                      ;ELSE SET BIT 3
                                 ;(ARB ID IS ONLY 5 BITS)
ID_TR:  XRL   A,ICDATA               ;ID BIT TRANSITIONS
ROT_A:  CLR   C                   ;ROTATE ZEROS INTO ACC
        RLC   A                    ;LOOK FOR TRANSITIONS
        JNC   CHK_ACC          ;IF CARRY BIT IS NOT SET
                                                ;THEN JMP
        DJNZ  TEMP,ROT_A ;ELSE TALLY UP ONE TRANSITION
ERR_3:  AJMP  VALERR        ;IF TRANS_CNTR REACHES ZERO
                                              ;THEN ERROR
CHK_ACC:JNZ   ROT_A                    ;IF ACC NOT ZERO
                                ;THEN CONTINUE LOOKING
        AJMP  WR01H            ;ELSE JUMP TO WR ROUTINE

$EJECT
```

C.22. ER03H Edit PSB ARB ID Register

```
;*******************************************************
;
;  NAME:          ER04H
;
;  PURPOSE:       EDIT ROUTINE FOR PSB DIAG REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   CHECKS ALL RFUs = 0, THEN IF
;                 LOOPBACK MODE BEING SELECTED,
;                 CHECKS FOR RSTNC* ASSERTED.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      ACC, DPTR
;
;  RETURNS:       JUMPS DIRECTLY TO WR06H, ELSE VALERR
;
;  UPDATE HISTORY:
;
;*******************************************************

ER04H:  MOV   A,#PSB_DIAG_MSK   ;GET PSB DIAG RFU MASK
        ANL   A,ICDATA              ;MASK ALL BUT RFUs
        JNZ   ERR_4               ;IF ALL RFUs NOT = 0
                                    ;THEN VALUE ERROR
        JNB   ICDATA.LBACK,END_E4     ;IF LOOPBACK NOT
                                ;SELECTED, SET THE WR REG
        MOV   A,#RST_TYPE_MSK      ;GET MASK FOR RESET
                                              ;TYPE BITS
        ANL   A,PSB_CS             ;MASK ALL BUT RESET
                                     ;TYPE FROM PSB C/S
        JZ    ERR_4          ;IF RSTNC HAS BEEN CLEARED
                                        ;THEN VALUE ERR
END_E4: AJMP  WR05H                           ;WR REG
ERR_4:  AJMP  VALERR              ;REPORT VALUE ERROR

$EJECT
```

C.23. ER04H Edit PSB Diagnostic Register

```
;*******************************************************
;
;  NAME:          ER05H
;
;  PURPOSE:       EDIT ROUTINE FOR REGISTERS
;                 REQUIRING > THAN CHECK
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   CHECKS FOR ALL RFUs = 0.  MASKS OFF
;                 ANY DON'T CARE BITS, CHECKS FOR WRITE
;                 DATA (ICDATA) <= MAX ALLOWABLE VALUE.
;                 IF BOTH CHECKS PASS, THEN JUMPS
;                 INDIRECTLY TO WRITE ROUTINE, ELSE
;                 VALUE ERROR.  EET(1) = RFU MASK.
;                 EET(2) = DON'T CARE MASK.  EET(3) =
;                 MAX ALLOWABLE WRITE VALUE.  EET(4) =
;                 WRITE ROUTINE OFFSET.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        ICDATA
;
;  DESTROYS:      ACC, TEMP, DPTR
;
;  RETURNS:       JUMPS INDIRECTLY TO WRITE ROUTINE,
;                 ELSE VALERR
;
;  UPDATE HISTORY:
;
;*******************************************************
```

C.24. ER05H Edit Register for Greater Than Check

```
ER05H:  INC    EET_OFF     ;POINT TO RFU MASK IN EETABLE
        %LOOKUP(#EETABLE,EET_OFF)              ;GET IT
        ANL    A,ICDATA          ;CHECK FOR ALL RFUs = 0
        JNZ    ERR_5               ;JMP IF ILLEGAL VALUE
        %GET_EEC     ;GET DON'T CARE MASK FROM EETABLE
        ANL    A,ICDATA        ;MASK OFF DON'T CARE BITS
                                             ;FROM ICDATA
        MOV    TEMP,A                        ;SAVE RESULT
        %GET_EEC           ;GET MAX ALLOWABLE WR VALUE
        %SUB(TEMP)            ;SUB VALUE BEING WRITTEN
        JC     ERR_5  ;IF ICDATA > MAX ALLOWABLE VALUE,
                                               ;THEN ERR
        %GET_EEC           ;ELSE GET WR ROUTINE OFFSET
        MOV    DPTR,#WRLBL          ;POINT TO WR ROUTINE
        JMP    @A+DPTR                 ;GO TO WR ROUTINE
ERR_5:  AJMP   VALERR               ;REPORT VALUE ERROR

$EJECT
```

C.24. ER05H Edit Register for Greater Than Check (Continued)

APPENDIX D
USER CALLABLE MACRO ROUTINES

```
;*****************************************************
;
;  MACRO NAME:    RD_MPC  (READ MPC)
;
;  SYNTAX:        RD_MPC(DEST,SRC)
;
;  PURPOSE:       READ REGISTER FROM MPC 1C REGISTER
;                 SET
;
;  DESCRIPTION:   THE VALUE CONTAINED IN OR BY SRC (MAY
;                 BE CONSTANT OR DIRECT ADDRESS) MUST
;                 BE A LEGAL MPC REGISTER ADDRESS.  THE
;                 REGISTER DATA IS RETURNED IN THE
;                 LOCATION SPECIFIED IN DEST.
;
;  INPUTS:        DEST, SRC
;
;  DESTROYS:      MPC_RNUM, ACC
;
;  RETURNS:       MPC REGISTER DATA IN DEST
;
;  UPDATE HISTORY:
;
;*****************************************************

%*DEFINE(RD_MPC(DEST,SRC))(
$SAVE nogen
        MOV     MPC_RNUM,%SRC
        MOVX    A,@MPC_RNUM
        %IF(%NES(A,%DEST)) THEN (
        MOV     %DEST,A
        )FI
$RESTORE
)

$EJECT
```

D.1. Macro RD_MPC

```
;*****************************************************
;
;  MACRO NAME:    WRT_MPC  (WRITE MPC)
;
;  SYNTAX:        WRT_MPC(DEST,SRC)
;
;  PURPOSE:       WRITE DATA TO MPC IC REGISTER
;
;  DESCRIPTION:   THE VALUE CONTAINED IN OR BY SRC (MAY
;                 BE CONSTANT OR DIRECT ADDRESS) IS
;                 WRITTEN TO THE MPC REGISTER SPECIFIED
;                 BY DEST, WHICH MAY ALSO BE A CONSTANT
;                 OR DIRECT ADDRESS.
;
;  INPUTS:        DEST, SRC
;
;  DESTROYS:      MPC_RNUM, ACC
;
;  RETURNS:       NOTHING
;
;  UPDATE HISTORY:
;
;*****************************************************

%*DEFINE(WRT_MPC(DEST,SRC))(
$SAVE nogen
        MOV     MPC_RNUM,%DEST
        %IF(%NES(A,%SRC)) THEN (
        MOV     A,%SRC
        )FI
        MOVX    @MPC_RNUM,A
$RESTORE
)

$EJECT
```

D.2. Macro WRT_MPC

```
;*****************************************************
;
;  MACRO NAME:    LOOKUP
;
;  SYNTAX:        LOOKUP(TABLE,OFFSET)
;
;  PURPOSE:       RETRIEVE BYTE FROM TABLE.
;
;  DESCRIPTION:   THE SPECIFIED TABLE IS INDEXED BY THE
;                 VALUE OF OFFSET AND RESULTING ADDRESS
;                 IS READ INTO THE ACCUMULATOR.
;
;  INPUTS:        TABLE, OFFSET
;
;  DESTROYS:      DPTR, ACC
;
;  RETURNS:       DESIRED BYTE IN ACC
;
;  UPDATE HISTORY:
;
;*****************************************************

%*DEFINE(LOOKUP(TABLE,OFFSET))(
$SAVE nogen
        %IF(%NES(A,%OFFSET)) THEN (
        MOV     A,%OFFSET
        )FI
        MOV     DPTR,%TABLE
        MOVC    A,@A+DPTR
$RESTORE
)

$EJECT
```

D.3. Macro LOOKUP

```
;*************************************************
;
;  MACRO NAME:   SUB     (SUBTRACT)
;
;  SYNTAX:       SUB(OPR)
;
;  PURPOSE:      SUBTRACT OPERAND FROM ACC
;
;  DESCRIPTION:  THE CARRY FLAG IS CLEARD AND THE
;                VALUE SPECIFIED IN OR BY OPR IS
;                SUBTRACTED FROM THE ACCUMULATOR.
;
;  INPUTS:       OPR
;
;  DESTROYS:     ACC, CARRY FLAG
;
;  RETURNS:      RESULT IN C AND ACC
;
;  UPDATE HISTORY:
;
;*************************************************

%*DEFINE(SUB(OPR))(
$SAVE nogen
        CLR     C
        SUBB    A,%OPR
$RESTORE
)

$EJECT
```

D.4. Macro SUB (Subtract)

```
;*****************************************************
;
;  MACRO NAME:    MOVBIT  (MOVE BIT)
;
;  SYNTAX:        MOVBIT(DEST_BIT,SRC_BIT)
;
;  PURPOSE:       PERFORM A BIT MOVE OPERATION
;
;  DESCRIPTION:   THE SOURCE BIT IS MOVED TO THE CARRY
;                 FLAG.  THE CARRY FLAG IS MOVED TO THE
;                 DESTINATION BIT.
;
;  INPUTS:        DEST_BIT, SRC_BIT
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       DEST_BIT = SRC_BIT
;
;  UPDATE HISTORY:
;
;*****************************************************

%*DEFINE(MOVBIT(DEST_BIT,SRC_BIT))(
$SAVE nogen
        MOV     C,%SRC_BIT
        MOV     %DEST_BIT,C
$RESTORE
)

$EJECT
```

D.5. Macro MOVBIT

```
;*******************************************************
;
;  MACRO NAME:   GET_EEC (GET EDIT ENTRY CONSTANT)
;
;  SYNTAX:       GET_EEC
;
;  PURPOSE:      INDEX INTO EETABLE TO GET NEXT EDIT
;                ENTRY CONSTANT
;
;  DESCRIPTION:  EET_OFF IS INCREMENTED AND THE CODE
;                BYTE AT EET_OFF FROM DPTR (ASSUMED TO
;                BE POINTING TO EETABLE) IS MOVED TO
;                THE ACCUMULATOR.
;
;  INPUTS:       NONE, HOWEVER DPTR ASSUMED = EETABLE
;                (INTR ROUTINES USING DPTR SHOULD
;                PUSH/POP).
;
;  DESTROYS:     ACC
;
;  RETURNS:      NEXT EET CONSTANT FROM LIST IN ACC
;
;  UPDATE HISTORY:
;
;*******************************************************

%*DEFINE(GET_EEC)(
$SAVE NOGEN
        INC     EET_OFF
        MOV     A,EET_OFF
        MOVC    A,@A+DPTR
$RESTORE
)
$EJECT
```

D.6. Macro GET_EEC

intel

APPLICATION NOTE

AP Note-431

March 1989

Multibus II Simplifies Partitioning of a Complex Design

FRANCOIS HUGUENIN
SENIOR FIELD APPLICATION ENGINEER
INTEL SWITZERLAND

Order Number 280691-001

MULTIBUS II SIMPLIFIES PARTITIONING OF A COMPLEX DESIGN

Francois Huguenin
Senior Field Application Engineer
Intel Switzerland

PREFACE

Several years ago, prior to joining Intel, I worked at the Swiss Federal Institute of Technology in the Group for Automatic Control. As part of my job responsibilities there, we made some real-time simulation studies in the field of aircraft performance computers. The engine model was implemented on an 8086-based single board computer. It also had special custom-made hardware for displaying the instruments similar to those in the original aircraft cockpit. During the course of this work, we were able to demonstrate the feasibility of using microcomputers in this application field. A paper describing this work was then presented at the AIAA Flight Simulation Technologies Conference, Long Beach, California in June 1980 [1]. While the use of microcomputers in aircraft simulation has become quite common in recent years, this work anticipated the impact of major cost reductions and performance improvements which were to follow in the flight simulation industry.

In more recent years, I have served as an Application Engineer for Intel Corporation in the Swiss District. In this role I am often confronted with the problem of how to demonstrate the capabilities of a product in a way which engineers can easily relate to and understand. This was the case with MULTIBUS II - an industry standard bus, specially designed for the more rigorous demands of multiprocessing. Since few applications today are partitioned to take advantage of parallel computing, my goal was to demonstrate how functional partitioning could be applied to a computationally intensive application with relative ease. The goal of breaking the "Von Neuman bottleneck" would be to increase the aggregate computing power without a substantial increase in overhead.

For assistance on this project, I called upon the Furrer & Gloor Company, a MULTIBUS manufacturer with broad experience in industrial automation. With their assistance I was able to modify the work done seven years ago on aircraft simulation and convert it to MULTIBUS II to demonstrate multiprocessing. This application provides a good example of the typical MULTIBUS II design cycle. We begin with the basic architectural decisions, define the message contents between processors, show the use of software development and debug tools, and then finally test for performance tuning on the final system once it is up and running. As you read through this document you will learn about the process of application development in addition to the special demands of the aircraft simulation experiment.

I would like to thank Markus Schoenbucher and Christoph Graf of the Furrer & Gloor Company for their great support during the weeks of implementation in the labs. George Walker of WeDV deserves special thanks for designing and debugging the iRMX® II part of the application. I wish also to thank P. Marti, H. -R. Aeschilmann, K. Krizaj and B. Leiser of the Simulator Maintenance Group at Swissair in Zurich who gave me access in the early phase to all the important engine data and most valuable inputs on the flight simulation technology in general back in the late 70's. Finally I especially want to thank Roger Finger from MULTIBUS II Application Engineering at the Intel factory in Hillsboro, Oregon for his guidance in helping me get this Ap Note done and for his corrections to my "Swiss English".

— Francois Huguenin

1.0 INTRODUCTION

Aircraft simulation trains pilots at a substantial savings. Flight crews can train around the clock, without regard to weather conditions and airport congestion. This important simulation model is a complex technical system which includes several high-performance digital computers. One key part of the simulator is the engine model: it is vital to the functioning of the entire machine:

- For the flight model equations (thrust and torque)
- For the "man in the loop" process with the important flight deck instruments (the pilot reacts on the settings and feedback from the engine instruments)
- For the flight engineer with all subsystems which are fed by the engines (hydraulics, electrical power supply, air conditioning, anti-icing, etc.)

The basic problem for good simulation is computing speed, because the quality of the pilot's training is a direct function of the flight simulator's update speed. Previously most simulators were using more than one processor based on a shared memory architecture. The trend today is dictated by the ever-increasing complexity of modern aircraft which are using more and more electronics on board. The new generation of flight simulators has a collection of black boxes (navigation computer, display units, engine control) used directly as in the original aircraft. In this complex environment, the engine is a closed subsystem which can be modeled without having to build a whole flight simulator. This part of the machine has well-known interaction points to the rest of the simulator. The work which was done at the Swiss Federal Institute of Technology some years ago was to be part of an on-board performance computer to calculate fuel optimal flight trajectories [2]. The following implementation of the engine model with MULTIBUS II is based on these results.

2.0 DESIGN PROBLEM OF THE SIMULATION OF THE JET ENGINE

The General Electric CF6-50C Fan Jet Engine which is installed on all Swissair's DC-10 aircraft provides our study model. This engine of the now "older" generation delivers up to 75% of the thrust power through the fan. The model designed during the years 1978 to 1980 is based on data made available by the engine manufacturer General Electric, McDonnell Douglas for the installed data (measurements made during the certification of the DC-10) and finally from CAE, the flight simulator manufacturer, with the actual programs written in assembler for the SIGMA computer.

2.1 The engine model

In order to be as realistic as possible, the engine model is very complex and has the following main characteristics:

- It is multivariable, with secondary interdependencies.
- It has basically two states: the transient state when coming up after ignition, and steady state when being held at a stable working point.
- It is a function of many parameters which directly influence the engine.

A cross section of the CF6-50C engine is shown in Figure 2.1.

Undoubtedly, the main problem of the model is the fuel control unit, which controls the whole engine. This unit also defines the transient behavior of the engine startup and excursions of the working point in the steady state mode. Of course, some simplifications are necessary for simulation. It was necessary to concentrate on the steady state model to reduce the modeling problem to a manageable task [1]. No startup

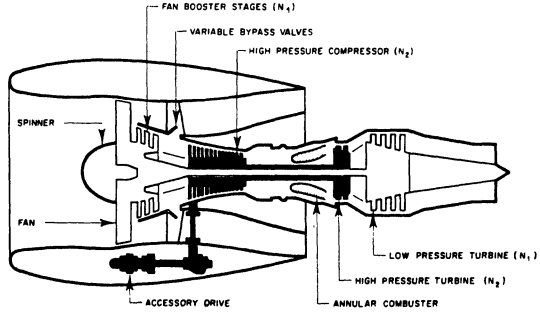


FIGURE 2.1: THE CF6-50C ENGINE LAYOUT

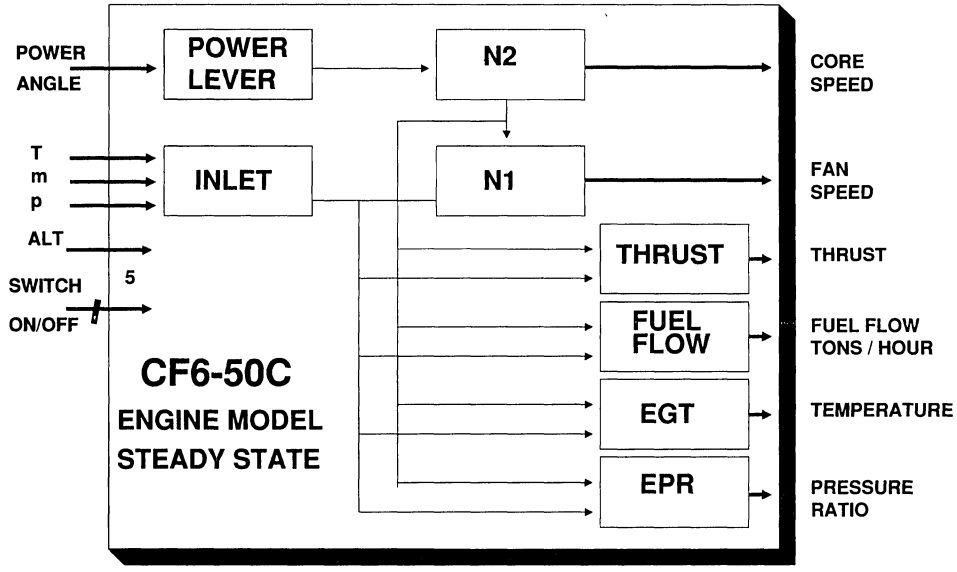


FIGURE 2.2: ENGINE MODEL STRUCTURE FOR STEADY STATE OPERATION

sequence of the engine was to be modelled, but this approach has the great advantage of allowing the direct use of the previously mentioned data tables from the engine manufacturer. The model was exact in the range for which the data tables were originally measured. This reduced and more manageable model is show in Figure 2.2.

A collection of modules are building the model with some interactions between them. These subsystems are:

- INLET: Reduction of mach, temperature and pressure to standard values sea level, standard day and temperature
- POWER LEVEL: Model of the power lever in the cockpit. From these, the pilot sets the throttles
- N2 TRANSIENT: Simplified model of the fuel control unit (basically a fourth order digital filter)
- N1: Model of the FAN as function of mach and N2
- THRUST: Model of the thrust as function of N1 and N2
- FUEL FLOW: Model of the fuel flow inside the engine in tons per hour as function of N2, altitude and mach
- EGT: Model of the engine temperature as function of N2 and mach
- EPR: Model of the engine pressure ration as function of mach and N2

The power plant of an aircraft is, as mentioned before, a closed system and provides additional tasks besides thrust for motion. The engine must provide fresh air for the cabin, and electric power for the cockpit, kitchen, cockpit electronics, etc. It must also activate and sustain the hydraulic subsystem for the control surfaces of the wings, and feed the anti-icing system for the wing tips, etc. The model, to be realistic, must take into account that these additional tasks will also have an appreciable influence on the state of the engine.

This can be demonstrated best with the fuel model as shown in Figure 2.3. The fuel model is basically a function of mach and N2 (referred core speed). This gives the main data curve and the basic fuel flow in tons per hour and will be interpolated through a high-speed algorithm. When the aircraft is climbing, the fuel flow will change according to a second data curve for correction with the value DELTA EWFA1. The correction, due to the on or off switching of air conditioning, etc., will also be taken into account with other data tables. The total fuel flow, as well as the fuel used by the engine, finally can be calculated with all the deltas. This model was validated with actual data taken from the Aircraft Data System of the DC-10 [1].

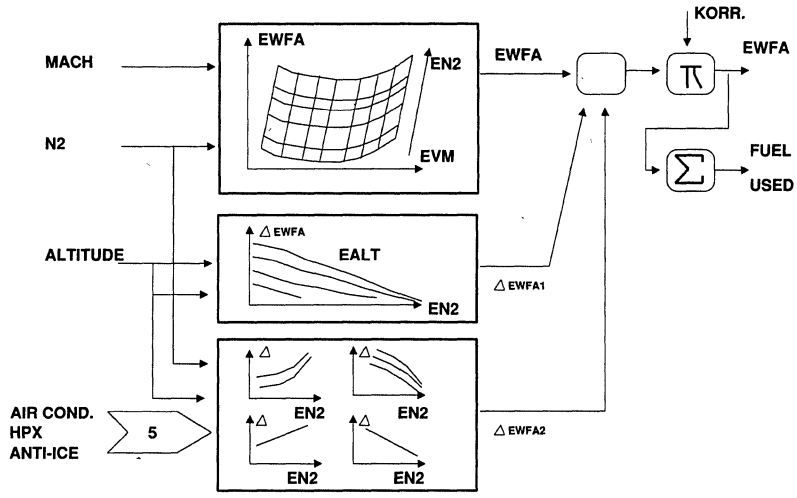


FIGURE 2.3: EXAMPLE OF THE FUEL SUBSYSTEM MODEL

2.2 The design problem

The task to be done now can be summarized as follows:

> **Every 20msec calculate a new state of all three engines of the DC-10 based on the power lever position of the cockpit and the subsystem switches for air conditioning, anti-ice, etc.**

What seems trivial in one sentence is in reality not! The original engine model was designed around a single processor modeling a single jet engine. The new design will extend the model to three engines, and will add an I/O subsystem to provide an improved human interface. These new requirements introduce a need for synchronization and communications across the backplane. The implementation of the single processor system will be discussed first, followed by the multiprocessing extensions.

2.3 The Uni-processor approach

The original engine model was designed around an iSBC 86/12 Single Board Computer (without numeric coprocessor) [1]. A small display unit was built as a copy of the central instrument panel of the cockpit to give feedback on the model's status [See Figure 2.4]. With a single processor, the software architecture is

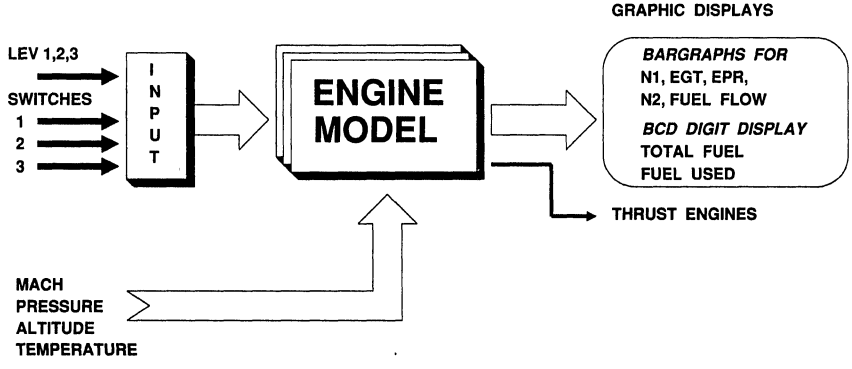


FIGURE 2.4: THE ENGINE MODEL ARCHITECTURE FOR THE DC-10

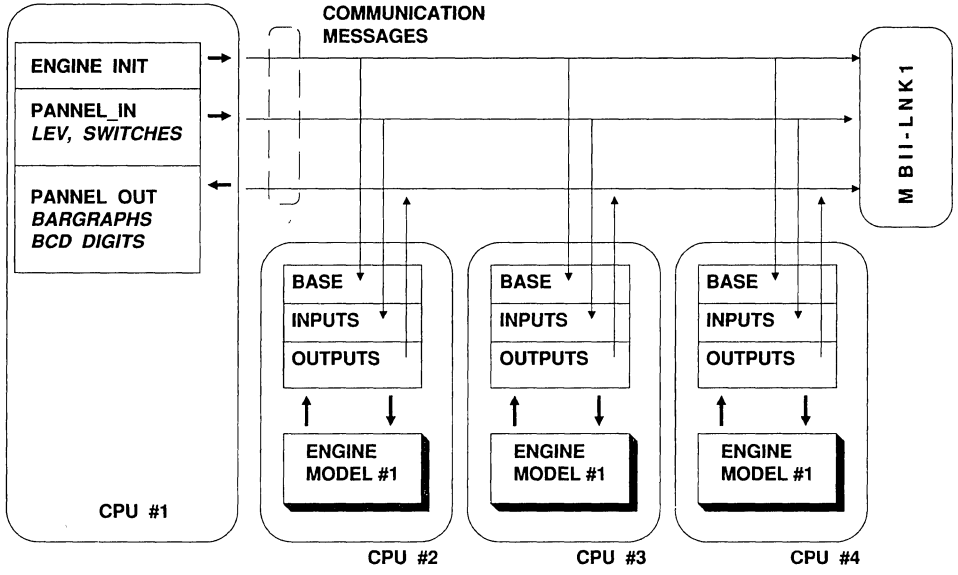


FIGURE 2.5: FUNCTIONAL PARTITIONING USING FOUR PROCESSORS

relatively simple: on every clock interrupt, the I/O unit was activated to read input data (power level position, on/off switches, etc.) and the engine model called three times in sequence with the appropriate parameters. Afterwards, some scaling took place to display the bar graphs and fuel used digits [see Figure 2.4]. Using this approach, the compute time depended on programming quality in the loop and therefore was optimized using assembler coded routines. The complete computing cycle consists of: read the inputs, calculate the engine model three times with the new state due to input parameters, scale the results and output them to the I/O unit.

2.4 The multiprocessor approach

To relieve the computational bottleneck and further reduce the update time, additional processors are needed. The basic question when using multiple processors is how to partition the application (in this case the modeling of all three DC-10 engines). For this application, one processor for each engine is appropriate because each engine model can be considered as a closed system with well-known interaction points. This results in a four-computer system: one processor must be a kind of "master" to coordinate the I/O and simulate the flight simulator; the remaining three are each modeling one engine of the DC-10.

The data flow changes slightly compared to the one processor approach. Because each processor is independent, each must be synchronized in some way. Using the MULTIBUS II technology [4] overcomes the hurdles of designing a multiprocessing hardware and is available off the shelf. The synchronization uses messages, exchanged between the host and its partners allowing a data exchange. This results in the architecture shown in Figure 2.5 where CPU #2, #3, #4 all have a local implementation of the engine model. The host or CPU #1 will handle all the coordination work and some key functions for the transient model and display scaling. The coupler board MBII/LNK-1 will handle the input-output processing.

3.0 MULTIPROCESSOR SOFTWARE DESIGN WITH iRMX II

The architecture chosen for the application is based on a fully- configured iRMX II system as a host, complemented with three "EPROM"-based application processors, one for each engine of the DC-10. This decision was made to minimize the amount of recoding which would be necessary for the original engine model. Software which was written a couple of years ago can be reused without change. In the process, the older iSBC 86/12 hardware will be upgraded to iSBC 286/100A boards for faster execution speed and Multibus II compatibility. Figure 3.1 shows the system architecture of the new multiprocessing system.

Each of the four processors in this system has some primary task to fulfill:

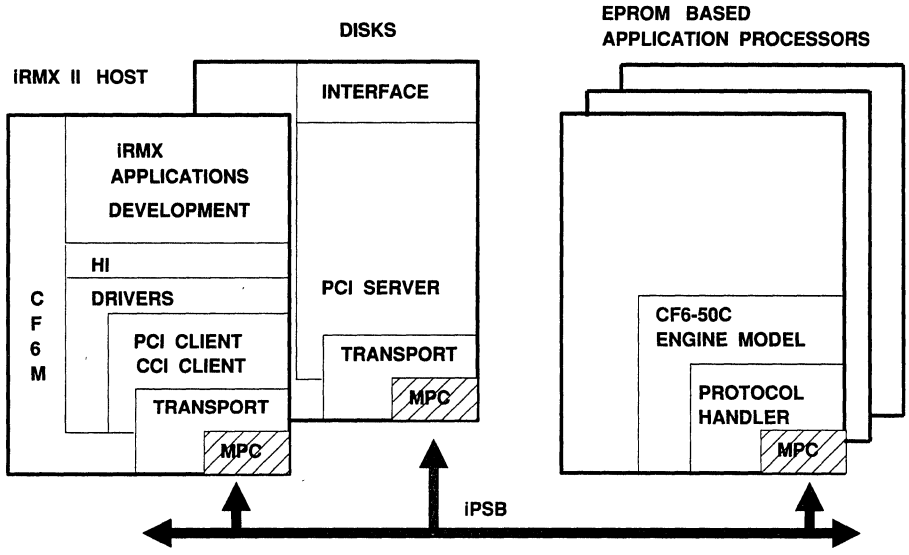


FIGURE 3.1: BASIC ARCHITECTURE OF THE SYSTEM

Host:

This is the central control processor which runs a complete real-time operating system. The iRMX II operating system was configured with a human interface, disk I/O subsystem (PCI - Peripheral Communications Interface), and a message passing communications layer. The iRMX II console will also be used as an on-target host for software development.

Applications Processors:

Each of the applications processors runs an EPROM based program which contains the engine model software. The only change which was required to

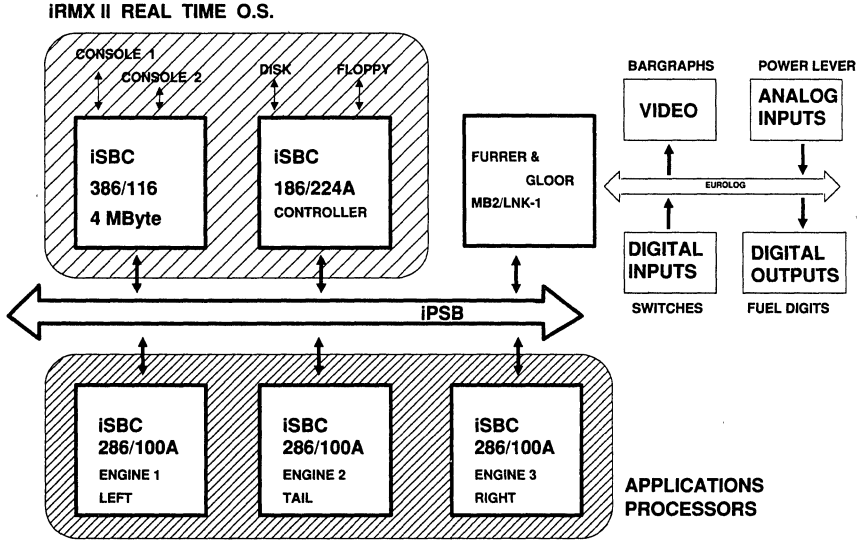


FIGURE 3.2: THE APPLICATION SYSTEM ARCHITECTURE

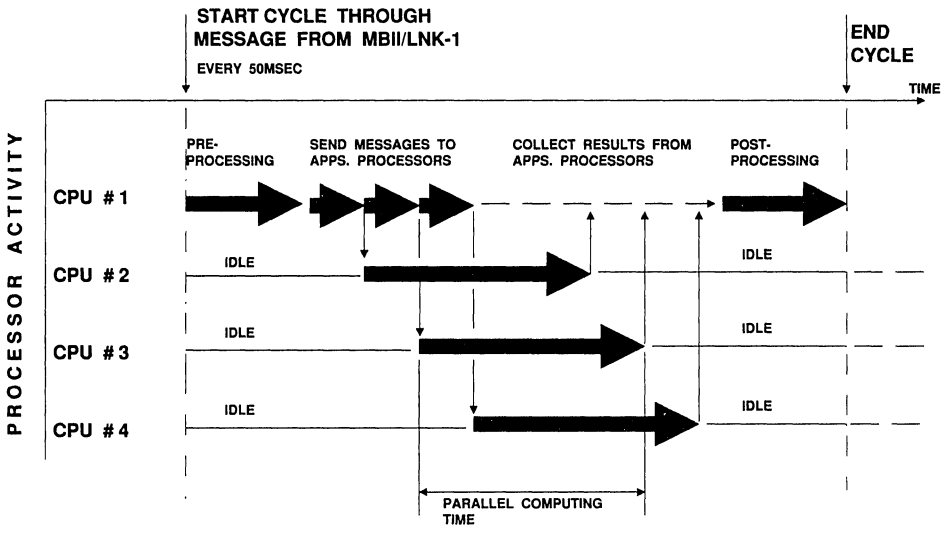


FIGURE 3.3: THE PROCESSOR ACTIVITY TIMING DIAGRAM

the original software was the addition of a message passing communications interface. Since there is no operating system on these boards, a low-level MPC protocol handler is responsible for this function.

LNK-1:

This board serves as a bridge to the low-cost I/O subsystem and the master clock during simulation time.

3.1 The I/O subsystem

As previously mentioned, the engine model also contains a display unit for control of the model. In order to add this important function, a coupler board manufactured by an independent MMG vendor (MULTIBUS Manufacturers Group) was used as a bridge to the I/O subsystem, as shown in Figure 3.2. The I/O system interfaces to:

- A panel with the "on/off" switches for each engine (anti-ice subsystem, etc.) as mentioned in section 2.1.
- An A/D converter to read and convert the position of the throttles to a binary value.
- A video controller to display the state of the engine as on the cockpit instruments.

In the following sections we will discuss in detail the various steps necessary to integrate the above design into a real MULTIBUS II system:

- Choice of the messages and their structure
- HOST startup sequence, iRMX II related topics
- Use of the iRMX II communications layer
- "Application Processor" design

3.2 Choosing a communications protocol

The MULTIBUS II architecture defines several types of processor access to the parallel system bus [4.1]: solicited message transfer, unsolicited message transfer, dual port memory access and interconnect address access. Interconnect address space will be used for configuration and initialization of the MULTIBUS II system and will not be accessed following system start-up. Dual-port memory as implemented on the MBII/LNK-1 board is used in the design [3] and allows use of unsolicited messages and memory transfer. The transfer of information between the host and the application processors is expected to require high speed transfers of numerous small blocks of data. For this, message passing is ideal.

The decision as to what type of message transfer will be used (solicited or unsolicited) will be based on how much data has to be transferred among the processors. As shown in Figure 2.5, there are basically three types of messages to be transferred:

- BASE: This message contains the base information, such as mach, pressure and altitude for the INLET

portion of the engine model [Figure 2.2]. This message is outbound from the host only.

- INPUTS: This message contains the N2 variable (core speed) of the engine, as calculated from the preprocessing in the Host. It also contains the decoded switch information from the panel of the model "cockpit" for the "on/off" switches of the subsystems, i.e., anti-ice, etc.
- OUTPUTS: This message has much more data to transfer back to the host for the final processing: all main computed data from the model must be sent back to the host.

Once the structure of "what has to be transferred" is defined, detailed analysis begins on the amount of data to be transferred. In this application the number of bytes to be transferred is relatively small, as follows: BASE - 3 words; INPUTS - 7 words; OUTPUTS - 9 words. During the implementation phase it was quickly realized that it would be wise to define a flag word to ease the debugging task. This flag word is an overhead, but it allows identification of messages being received for debugging.

The next choice to make is what kind of message: unsolicited or solicited? Using solicited messages means that a certain amount of data has to be sent between the processors in order to be useful. In this design, the use of a solicited message will mean only overhead and complexity in the software. Only a few words need to be transferred. Therefore the choice is to simplify and use unsolicited messages only. In a case where a data transfer with more than 20 bytes is needed, an unsolicited message may be sent twice. This is faster than setting up an entire solicited transaction [4.3]. The big advantage is simplification of application software and of the debugging task.

3.3 The final configuration and state cycle

Looking at the final configuration of the system as shown in Figure 3.2, there will be a number of transfers through the PSB for each simulation cycle. Each of them will be started by the local processor sitting on the EUROLOG local bus. This processor will read the analog data, convert it to a binary value, read the switch positions and pack it all in a MULTIBUS II message. This will be sent to the host. The host will decode it and do some pre-processing and send it to the appropriate application processor (or engine). After a simulation step has been done locally, each of the application processors will send the result back to the host for encoding the display and some post-processing. Afterwards, the results will be sent back to the MBII/LNK-1 board using two unsolicited messages for the display process because there are more than 20 bytes needed for this task. Therefore, a complete simulation cycle will involve nine unsolicited transfers among the various processors.

3.4 The cycle time analysis

The next task is to look at the activities of all processors with respect to time. For this purpose, a "processor activity timing" diagram, as shown in Figure 3.3, is set up. The horizontal axis is time; the vertical has one entry for each processor. During the work cycle, in this case one simulation step based on the incoming unsolicited message from the MBII/LNK-1 coupler board, the main tasks of each of the processors is estimated and shown with an arrow. Each outgoing or incoming message is a transition vertical arrow to the next processor and means the start or the end of an activity.

Using this diagram, it is now possible to make a quantitative analysis about the workload of each of the processors. In this case we see that the CPU #2, #3 and #4 are usually sitting idle waiting for messages. This simply means that there is a lot of spare processing time available. The diagram also shows that during some time in the cycle there is true parallel processing happening, which confirms the design goal of shortening the simulation calculation cycle. The compute time of each of the application processors is usually bound to the engine model calculation. This will be studied in more detail in section 5.

3.5 The iRMX II host

As earlier mentioned, there will be a human interface (HI) job running under iRMX II which will monitor all the simulation activities. This monitor program will use the comm layer of iRMX II and be responsible for the startup of the system.

One part of the monitor program will also be the pre- and post-processing for the engine model as mentioned before. The startup sequence will include a board scan in order to be able to locate the application processors and define the necessary communications ports. The monitor program will function as follows:

- Create the connections to the terminal
- Scan the backplane and check the BIST (Built-In Self Test) status of each board
- If okay, then define the ports and sockets for the iRMX communications layer

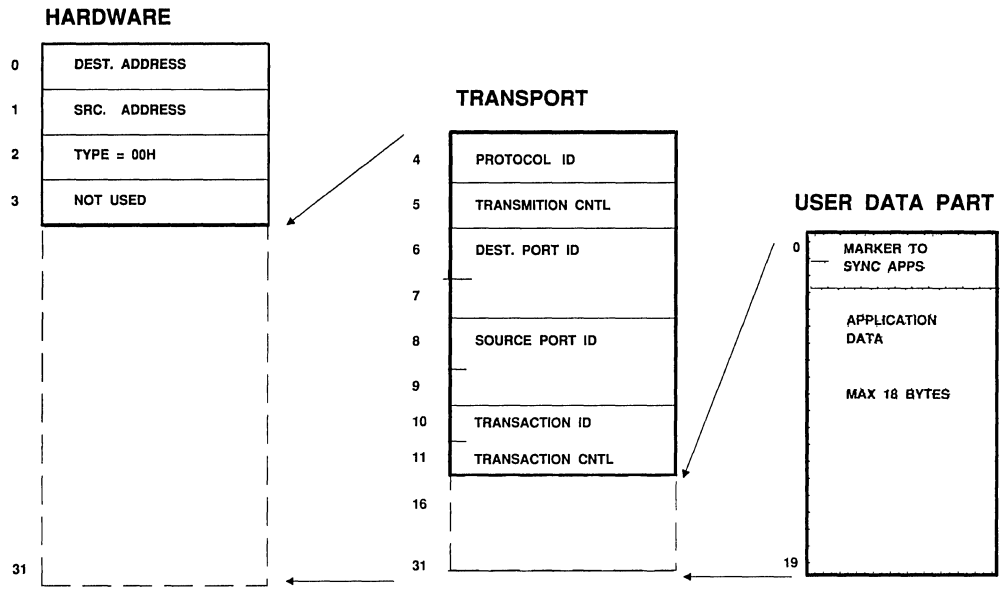


FIGURE 3.4: THE iRMX II COMM-LAYER UNSOLICITED MESSAGE FORMAT

- Create a buffer pool to preallocate free memory segments
- Create the iRMX memory segments and release them to the buffer pool
- Attach pool to port
- Ask for an amount of fuel in tons at the console
- Send the startup synchronization message to CPU #2, #3 #4 and to the MBII/LNK-1 board
- Send the start command to the MBII/LNK-1 board and give it control
- Do every 50msec until no fuel available.
  - Receive the message from the MBII/LNK-1 board with the decoded power lever angle and switches
  - Make the preprocessing
  - Send to each application processor the incoming data for a simulation step
  - Collect the resulting messages asynchronously (there might be a slower board ...!)
  - After having received all the results, make the post-processing
  - Send the two result messages back to the MBII/LNK-1 board. This finishes the "new state calculation" cycle.
- When no more fuel, stop everything and ask again at the console for more fuel

The coding of this monitor program is done in PL/M 286 using the iRMX II system calls. All the features of the PL/M 286 are used, especially the STRUCTURE DEFINITIONS which are very useful in this kind of application programming. Portions of the code are shown in Appendix B. Note that the transport system calls are very easy to use.

4.0 THE APPLICATION PROCESSOR DESIGN

The structure of the local software is, in principle, simple, due to the task which must be performed and can be split in two portions:

- A main program which will initialize the board and the whole local software, and
- An interrupt handler to handle the incoming messages accordingly. In addition, the interrupt handler calculates the new state of the engine and contains the engine model which needs no modification.

This leads to the design of several separate modules with well-known tasks:

- MAIN: Init of the board upon cold reset, (int. controller, MPC)
- INTERRUPT HANDLER: Activated on an MPC hardware interrupt. Will initialize the engine model or calculate the next state of the engine (so-called simulation step).

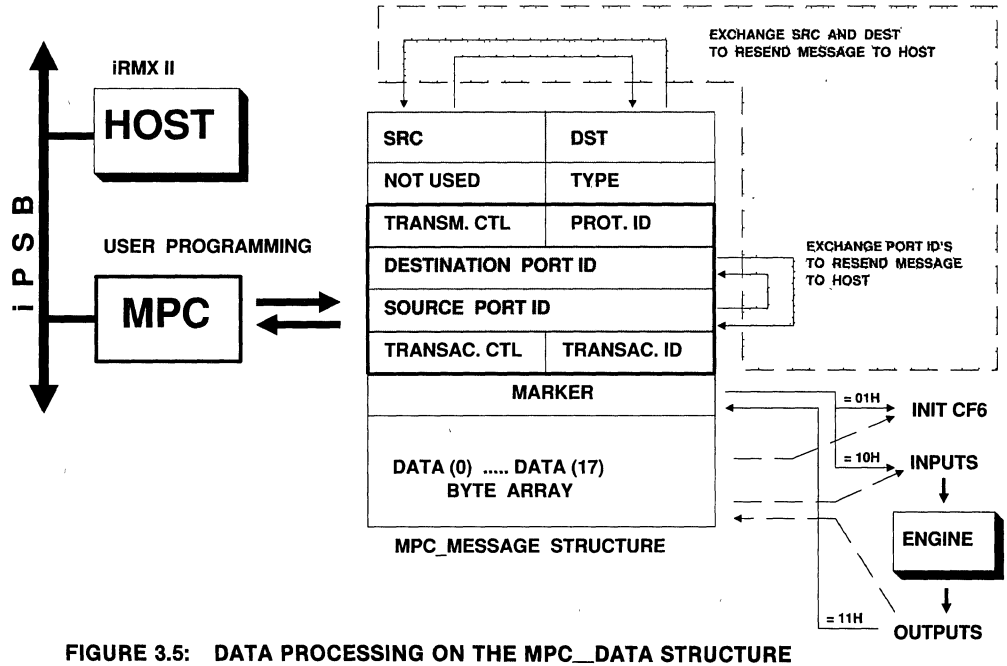


FIGURE 3.5: DATA PROCESSING ON THE MPC_DATA STRUCTURE

– MPC-LOW LEVEL DRIVER: Receive a message, transmit a message, init the MPC, error case.

The MPC low level driver routines can be directly taken from the MPC User's Manual [4.2]. The programming was done using 32 byte messages since the structure of the application fits exactly in the unsolicited message format. However there are some implications when receiving an iRMX II message sent under the comm layer. The general format of the message is shown in Figure 3.4. This explains that not all available bytes can be used in the unsolicited message transfer since eight of them are used for the transport protocol overhead, making the logical task binding over the bus possible [4.3]. Therefore, the user has to be careful when using the message formatted under iRMX II and received locally without any operating system software. Since only one single task runs on the board (our interrupt handler), no port and sockets are needed locally, but the host has a port and a socket defined for accessing the application processor. To transmit the data back as described in the design section above, the comm layer must "understand" what is coming back. The following operations ensure this:

1. The source and destination fields of the message structure will have to be exchanged before returning the message with the engine simulation results. That is for the low-level hardware and is absolutely logical.

2. Exactly the same has to be done for the transport part of the message structure. The DESTINATION_PORT_ID and the SOURCE_PORT_ID have to be exchanged to allow the comm layer software to receive the results correctly.

The second step seems trivial, but it took several hours in front of the emulator to understand why the engine variable N1, a result of the simulation step, was always the same value. We had not taken into account the unsolicited message set up under iRMX II which has the whole TRANSPORT PROTOCOL integrated. Once this was discovered, we had no problems at all. Therefore, all applications using no operating system, kernel or executive locally, must have an "own" protocol handler implemented. Main portions of the documented code listings for total implementation of the application processor can be found in Appendix C.

The "data processing" done flag in the MPC_MESSAGE data structure is shown in Figure 3.5. The previously mentioned marker differentiates the kind of message and is, in some sense, part of a "user application protocol". In this case it allows initialization of the engine software or to make a simulation step.

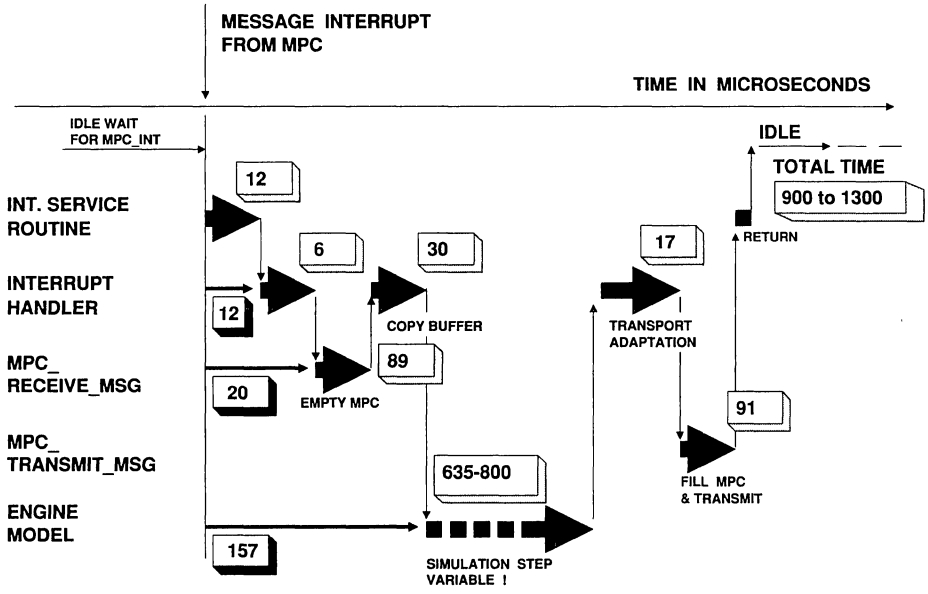


FIGURE 5.1: THE APPLICATION PROCESSOR TIMING ANALYSIS

5.0 DEBUGGING AND PERFORMANCE ANALYSIS

The debugging of the application for the local processor was done using the I2ICE 286 emulator. This instrument allows a very efficient debugging, especially when using the macro facility to display data. Therefore a small library of macros was written to display the incoming and outgoing messages. With this it was possible to quickly locate the above mentioned problem of the missing TRANSPORT adaptation. The whole application was debugged in the high speed RAM of the emulator and afterwards relocated to be EPROM resident. However, debugging a multiprocessor system is, by definition, not simple because of all the coordination involved. The debug session around the application processor was based on the following methodology:

- Send a known message from the host
- Debug the MPC_RECEIVE_MESSAGE routine step by step and get the known message to local RAM
- Loopback with same message to check the MPC_TRANSMIT_MESSAGE routine
- When the loopback is okay, then add the full engine calculation and ... it runs!

A small log of a debug session with I2ICE 286 and the use of the predefined macros can be found in Appendix D1. It shows the big advantage of using symbolic debugging.

Afterwards, an analysis with the iPAT Performance Analyzer was done to get more confidence about the timing situation. Most impressive was the terrific speed of the whole thing.

The iPAT performance analyzer offers many features such as profile, coverage, usage, linkage mode, etc.[5]. In this kind of environment, the objective was to measure how long the various elements of the cycle are, so only the duration mode was of real use. Of first interest was to check how much time is spent in a routine and the latency from the MPC Interrupt Hardware signal to the start of a particular routine. Initially the cycle time of 50msec was chosen to have enough spare time due to the use of the relatively slow 5MHz iSBC 86/12A. A time of about 15.5msec could be achieved with the MULTIBUS I system.

Porting to the much faster iSBC 286/100A with 8MHz speeds up the whole design. Depending on the switch position for the engine subsystems [see Figures 2.3 and 2.4] the total cycle time is in the range of 1380 to 1490 microseconds. Figure 5.1 shows in detail the timing situation of one cycle and how much time was spent in each of the routines. Note the overhead due to the MULTIBUS II transport, and that the message passing handling is not very large. The MPC handling (receive and transmit the message) takes 180 microsec, and the swap for the transport adaptation takes 17 microsec, which means a total of around 200 microsec for the "overhead". Therefore, the bottom line message is

very clear: There is enough spare time to add more functions to the engine model. During the timing analysis, 15 tons of "software fuel" was burned!

6.0 SUMMARY AND CONCLUSIONS

The design and implementation of the engine simulation model has given the opportunity to discuss and study a couple of interesting problems around a MULTIBUS II application:

- Given a task, how to partition an application to use more than one processor and assess the problems around it.
- To demonstrate that the port of an existing software written a couple of years ago can still be integrated and ported to the latest hardware technology.
-
- To adapt an existing MULTIBUS I system to use the benefits of the MULTIBUS II technology.
- To demonstrate the use of the iRMX II operating system together with custom-made software and study its implications when the comm layer is used.

In addition, this application allowed discussion of actual implementation starting from the given problem statement and ending with the final integration in a real system using single board computers. The results show the feasibility, including the important performance analysis.

Appendix A

A. Bibliography and related readings

1. Francois Huguenin. Microcomputer Based Engine Model Used in Flight Simulation Applications. AIAA Paper 81-0973, AIAA Flight Simulation Technologies Conference, June 16-18, 1981, Long Beach, CA

2. F. Huguenin, P. Grepper. Four-Dimensional Helical Approach of Aircraft in an Air Traffic Control Environment. AIAA Paper 79-1776R, Journal of Guidance and Control, June 1981

3. Furrer+Gloor AG MULTIBUS II MMG
   Furrer+Gloor Silbernstrasse 10
   CH 8953 Dietikon ZH Switzerland
   MBII/LNK-1 Coupler Board User's Guide

4. Intel MULTIBUS II Documentation:
   4.1 MULTIBUS II Specifications, order #146077C
   4.2 MPC User's Manual, order #176526-001
   4.3 MULTIBUS II Transport Specs, order #453508-001
   4.4 iSBC 286/100A User's Manual, order #149093-001
   4.5 iRMX II Nucleus User's Guide, order #461845-001
   Nucleus Communication Services, section 12
   5. iPAT Analyst User's Guide, order #450583-002

Appendix B

B. Glossary of Terms

Application Terms:

T	Temperature
p	Pressure
m	Mach
HPX	Horse Power Extraction
EN2	Engine Variable N2 calibrated
EALT	Engine Variable Altitude calibrated
EWFA	Engine Variable Fuel Flow calibrated
EVM	Engine Variable Mach calibrated

MULTIBUS II Terms:

MPC	Message Passing Co-Processor (VLSI MULTIBUS II Interface)
Solicited Message Transfer	A data transfer through MULTIBUS II message space that requires buffer negotiation. May be up to 16Mbytes long.
Unsolicited Message Transfer	Unsolicited messages arrive at a host unpredictably and can have the effect of an interrupt. Message can have up to 20 bytes of user data.
Dual Port Memory access	A means of accessing shared memory between two processors. Hardware arbitration is required.
Interconnect Address Access	Access to the MULTIBUS II interconnect address space for test and/or configuration purposes.
Buffer Pool	A collection of memory buffers which are managed by the iRMX II Operating System.
Transport Protocol	This is a generic term describing the function of the software layer that implements the MULTIBUS II Transport Protocol as defined in the IEEE 1296 Specification.

APPENDIX C

C. Documented portions of the iRMX II host monitor program

Note that all comments written in bold are added to the original listing portions.

The following code listing contains the most important parts of the iRMX II Host Monitor Program. Since this is a quite large lisiting, it was choosen only to take out which is really important.

B1. Declarations

These word declarations make the interface to the exisiting assembly coded routines.

```
 840   1  =   declare std_conditions word external;
 841   1  =   declare state_1 word external;
 842   1  =   declare state_2 word external;
 843   1  =   declare state_3 word external;
 844   1  =   declare result_1 word external;
 845   1  =   declare result_2 word external;
 846   1  =   declare result_3 word external;
```

The following is an abstract of the definition for the messages which are used between the processors in the system.

```
           =
           =   /*
           =   ; 1. standard conditions message from main to all
                 slaves
           =   */
```

This is the general unsolicited message structure to be used.

```
 847   1  =   declare mpc_message structure
           =   ( dest byte,
           =   src byte,
           =   type byte, not_used byte, message_data (28) byte)
               public;
```

This message will be used for synchronizing all processors together.

```
 848   1  =   declare std_conditions_msg  structure
           =            ( header_std_conditions (4) byte,
           =                transport(4) word,
           =                std_marker word,
           =                evm word,
           =                ealt word,
           =                ep0 word,
           =                et0 word,
           =                dummy_std_conditions (10) word) at
                            (@std_conditions);
```

These messages are to be used for the information transfer between the application processors.

Note that the structures are overlayed to the word defined a lines 840 to 846 !

```
          =    ; 2.1 Engine #1
          =    */

 849    1  =    declare state_1_msg  structure
          =             ( header_state_1 (4) byte,
          =               transport (4) word,
          =               message_data (10) word) at (@state_1);

          =    ; 3.1 Engine #1
          =    */

 852    1  =    declare result_1_msg  structure
          =             ( header_result_1 (4) byte,
          =               transport (4) word,
          =               message_data (10) word) at (@result_1);
```

The following three procedures are used for creating a port, and to send and receive the messages under iRMX II control. See also iRMX II System documentation.

```
1074   1       NEW$PORT: PROCEDURE (port$token$ptr,id,type);
               /*****************************************/
               /*
               this procedure creates a port for access by iRMX II
               */


1075   2          DECLARE port$token$ptr POINTER;
1076   2          DECLARE id word;
1077   2          DECLARE type byte;
1078   2          DECLARE port$token based port$token$ptr token;

                  /*
                  **  Create a new port: port$token$ptr is pointer
                  **  to port$token to be returned
                  **     id is:  for data transport: port ID
                  **             for signal service: message ID
                  **   type is:  port type
                  */

1079   2          if type = data_port then do;
1081   3             port$info.port$id = id;
1082   3             port$info.type = type;
1083   3             port$info.flags = 0;
1084   3             port$token = rq$create$port
                                 (queue$size,@port$info,@status);
1085   3          end;

1086   2          if type = signal_port then do;
1088   3             msg$info.msg$id = id;
1089   3             msg$info.type = type;
1090   3             msg$info.flags = 0;
1091   3             port$token = rq$create$port
                                 (queue$size,@msg$info,@status);
1092   3          end;

1093   2          return;
1094   2       end NEW$PORT;
```

```
send a message to the MPC

1095   1      FORWARD$MSG: PROCEDURE (message$pointer,port$id);
1096   2         DECLARE message$pointer POINTER;
1097   2         DECLARE port$id BYTE;

1098   2      DECLARE
                  transaction$id          WORD,
                  status                  WORD;



1099   2         call no$exc$mode;       /*  Set Exception Mode
                                             to 0 (No RMX Action) */

1100   2         transaction$id = rq$send(
                              own$port$tok,sockets(port$id),
                              message$pointer, NIL, 0, 0,
                                    @status);

1101   2         call full$exc$mode;        /*  Reset Exception
                                              Handling */
1102   2         count$out = count$out+1;   /*  Update Counter for
                                              messages sent */

1103   2         RETURN;
1104   2      END FORWARD$MSG;
```

receive a message from the MPC

```
1105   1      WAIT$FOR$MSG: PROCEDURE (buffer$ptr) WORD;

1106   2         DECLARE buffer$ptr pointer;

1107   2         DECLARE buffer$ptr_origin POINTER;

1108   2         DECLARE
                 info_buf                STRUCTURE(
                                         flags           WORD,
                                         status          WORD,
                                         transaction$id  WORD,
                                         length          DWORD,
                                         forward$port    TOKEN,
                                         socket          DWORD,
                                         message(20)     BYTE,
                                         reserve(4)      BYTE),

                 status                  WORD;


1109   2         call no$exc$mode;    /*  Set Exception Mode to 0
                                          (No RMX Action) */

1110   2         buffer$ptr_origin = rq$receive(
                                       own$port$tok, notimeout,
                                       @info_buf, @status);

1111   2         call full$exc$mode;  /* Reset Exception
                                          Handling*/
                                      /*  Update Counter for
                                          Messages sent */
1112   2         count$in = count$in + 1;
1113   2         socket = info_buf.socket;
                 /*
                 ** We expect just a Control Message ==>
                 ** Copy contents of control message to buffer
                 ** provided by paramter buffer$ptr and return
                 ** sending host ID.
                 */

1114   2         call movb(@info_buf.message,buffer$ptr,20);
1115   2         return socket$def.host$id;

1116   2      END WAIT$FOR$MSG;
```

The following code section is the DO UNTIL NO_FUEL loop with the iRMX II System calls used to transfer the data to and receive them back from the Appliaction processors.

```
/*****************************************************************/
/*                                                               */
/*  Wait for state_in Message from LINK                          */
/*  Copy MSG to Buffer =>                                        */
/*                        SWITCH_DECODE,LEVER_TO_2,N2_TRANSIENT  */
/*  Send State_i Message to Engine-Simulators                    */
/*  Wait for Simulators having calculated  ==>                   */
/*          EGT_FILTER, SKAL_BARGRAPHS,SKAL_DIGITS               */
/*  Send display  Message to Link                                */
/*                                                               */

/*****************************************************************/


1366   2          DO WHILE nofuel=0;
```

The state_message contains the data from the I/O subsystem. When received, one simulation step can be processed.

```
                     /* Wait for STATE_IN Message, then Copy */

1367   3             id = WAIT$FOR$MSG(@state_in_msg.state_in);
```

Make the preprocessing.

```
1368   3             CALL SWITCH_DECODE;
1369   3             CALL LEVER_TO_N2;
1370   3             CALL N2_TRANSIENT;
```

Set the marker word to 10H prior to send the data.

```
1371   3             call movb
                          (@(10H),@state_1_msg.message_data,1);
1372   3             call movb
                          (@(10H),@state_2_msg.message_data, 1);
1373   3             call movb
                          (@(10H),@state_3_msg.message_data, 1);

1373   3             call movb (@(10H), @state_3_msg.message_data,
1);
```

```
1374   3            call forward$msg(
                              @state_1_msg.message_data,engine_1);
1375   3            call forward$msg(
                              @state_2_msg.message_data,engine_2);
1376   3            call forward$msg(
                              @state_3_msg.message_data,engine_3);


                    /*
                    ** Wait for all three having terminated:
                    ** Use logical variables all$done,done$1
                    **   ,done$2,done$3.
                    **
                    ** When a Message arrives, copy it to local
                    ** buffers.
                    */

1377   3            all$done,done$1,done$2,done$3 = FALSE;

1378   3            DO WHILE NOT all$done;

1379   4               id = WAIT$FOR$MSG(@intermediate);

                       /*
                       ** Message arrived: set corresponnding done
                       ** flag and copy buffer
                       */

1380   4               if id = host$ids(engine_1) then do;
1382   5                  done$1 = true;
1383   5                  call movb(@intermediate,
                                 @result_1_msg.message_data,28);
1384   5               end;
1385   4               if id = host$ids(engine_2) then do;
1387   5                  done$2 = true;
1388   5                  call movb(@intermediate,
                                 @result_2_msg.message_data,28);
1389   5               end;


1390   4               if id = host$ids(engine_3) then do;
1392   5                  done$3 = true;
1393   5                  call movb(@intermediate,
                                 @result_3_msg.message_data,28);
1394   5               end;

1395   4               all$done = done$1 AND done$2 AND done$3;
1396   4            END;
```

Make the postprocessing.

```
1397   3            CALL EGT_FILTER;
1398   3            CALL SKAL_BARGRAPHS;
1399   3            CALL SKAL_DIGITS;
```

```
                /*    send results to MBII/LNK-1 Board with two
                      consecutive unsolicited messages
                */

Set the marker word to 44H and 45H for identification
inside the MBII/LNK-1 Baord.

1400   3              call movb (@(44H),
                         @display_on_line_msg_1.display_on_line,1);
1401   3              call movb (@(45H),
                         @display_on_line_msg_2.display_on_line,1);

Send the two conscutive messages to the MBII/LNK-1 board.

1402   3              CALL FORWARD$MSG(
                            @display_on_line_msg_1.display_on_line,
                                                          link_1);
1403   3              CALL FORWARD$MSG(
                            @display_on_line_msg_2.display_on_line,
                                                          link_1);

                      /*
                      **  Update Message Counter for display on the
                      **  console
                      */

1404   3              status = put$char(@pos$in,fb);
1405   3              status = put$ddec(count$in,fb);
1406   3              status = put$char(@(alloff,0),fb);
1407   3              call put$line(nolf);

1408   3              status = put$char(@pos$out,fb);
1409   3              status = put$ddec(count$out,fb);
1410   3              status = put$char(@(alloff,0),fb);
1411   3              call put$line(nolf);

1412   3              ttime.systime=0;
1413   3              call dq$decode$time(@ttime,@status);
1414   3              if last$sec <> ttime.time(7) then do;
1416   4                 call movb(@pos$time,@outbuff,12);
1417   4                 call movb(@ttime.time,@outbuff(12),8);
1418   4                 call movb(@(alloff),@outbuff(20),4);
1419   4                 out$buff(24)=0;
1420   4                 call disp(@out$buff);
1421   4                 last$sec = ttime.time(7);
1422   4              end;
1423   3           END;
```

APPENDIX D

D. Documented code listing of the application software based in EPROM

This appendix contains most of the listings generated for the application processor EPROM resident software:

- MAIN05 : main module for cold start
- MPCDR3 : low-level MPC driver
- C286I6 : interrupt handling routine
- CF650L : engine model module

Note also the last page of the LOCATE MAP which gives an idea on how much memory was needed for this application.

```
iRMX 86 8086/87/88/186 MACRO ASSEMBLER V2.0 ASSEMBLY OF MODULE MAIN_MODULE_286
OBJECT MODULE PLACED IN MAIN05.OBJ
ASSEMBLER INVOKED BY:  :LANG:asm86 MAIN05.A86 DEBUG SYMBOLS TYPE

LOC  OBJ                  LINE     SOURCE

                             1     name    main_module_286
                             2
                             3     ;             PROJECT      CF6-50C Multicomputer
                             4
                             5     ;             08.10.87   v1.0 fh add reset not complete on entry point
                             6     ;             03.10.87   v1.0 fh add in it of ewfaold, efuelused
                             7     ;             21.08.87   v2.1 CB/Scm OCW2 has address pic_cnt10
                             8     ;             04.08.87   v2.0fh  change interrupt structure
                             9     ;             03.08.87   v1.1fh  change external segments
                            10
                            11     ;             03.08.87   v1.0fh  initial start up module
                            12
                            13
                            14     ; This module contains the cold reset start up of the sbc286/100A
                            15     ; and the interrupt processing routine.
                            16
                            17
                            18
                            19     ; note that all written in CAPITAL is original software written
                            20     ; end of 1979, early 1980 '
                            21     ; all other is modification done to port the original application
                            22     ; to Multibus II
                            23
                            24
                            25
                            26
                            27
                            28 +1  $nogen
                            29
                            30     ; interrupt controller PIC
                            31
  00C0                      32     pic_cnt10       equu     0c0h    ; command port 1
  00C2                      33     pic_cnt11       equu     0c2h    ; command port 2
                            34
                            35
----                        36     int_pointers    segment at 0
                            37
0000                        38                     org      0h
0000 (32                    39     type_x          dd       32      dup(?)  ; discard all system interrupts
     ????????
     )
                            40                                             ; which are not used here
                            41
                            42                     ; master pic of iSBC 286/100A
                            43
0080 00000000               44     type_32         dd       0       ; ir 0
0084 00000000               45     type_33         dd       0       ; ir 1
0088 00000000               46     type_34         dd       0       ; ir 2 message interrupt
008C 00000000               47     type_35         dd       0       ; ir 3 iPSB message error
0090 00000000               48     type_36         dd       0       ;
0094 00000000               49     type_37         dd       0       ;
0098 00000000               50     type_38         dd       0       ;
009C 00000000               51     type_39         dd       0       ;
                            52
                            53
----                        54     int_pointers    ends
                            55
                            56     ; external variables needed for initialization
                            57     ; first time after reset
                            58
```

```
LOC  OBJ                  LINE     SOURCE

----                        59     EFGIO   SEGMENT COMMON
                            60     extrn   ewfaold:word
                            61     extrn   efuelused:word
                            62     extrn   deltat:word
                            63
----                        64     EFGIO   ENDS
                            65
                            66     $eject
                            67
                            68
----                        69     STACKSEG                SEGMENT STACK
0000 (100                   70                     DW       100 DUP(?)       ;ALLOCATE 100 WORDS FOR STACK
     ????
     )
00C8                        71     STACKTOP        LABEL   WORD             ;OFFSET ADDRESS OF THE TOS
----                        72     STACKSEG                ENDS
                            73
----                        74     iodatseg                SEGMENT public
                            75
----                        76     iodatseg                ENDS
                            77
                            78
                            79     ; two external routines
                            80
                            81     extrn   mpc_init:far            ; initialization of the mpc
                            82     extrn   mpc_interrupt:far       ; mpc interrupt routine
                            83                                     ; for handling I/O and engine
                            84
                            85
----                        86     EFGCOD          SEGMENT PUBLIC
                            87
                            88             ;GIVE ASSEMBLER INITIAL REGISTER TO SEGMENT CORRESPONDENCE
                            89
                            90     ASSUME  CS: EFGCOD,     DS: IODATSEG,   ES: EFGIO,      SS: STACKSEG
                            91
                            92
                            93
0000                        94     interrupt_routine       proc
                            95
                            96
                            97     ; This Interrupt routine will be activated by the MINT line
                            98     ; of the MPC when the MPC has received a message
                            99     ; This routine call the interrupt handler written in PLM86
                           100     ; which will calculate a new state of the engine based on the
                           101     ; incoming message. Afterwards the result message will be formed
                           102     ; and send back to the host.
                           103
0000 50                    104             push    ax
0001 9A0000----      E     105             call    mpc_interrupt
                           106
                           107     ; EOI handling for the 8259A Controller
                           108
0006 B020                  109             mov     al,20h          ; non specific eoi
0008 E6C0                  110             out     pic_cntl0,al    ; write to ocw2
000A 58                    111             pop     ax
000B FB                    112             sti                     ; enable interrupts
000C CF                    113             iret                    ; return from interrupt
                           114
                           115
                           116     interrupt_routine       endp
                           117
                           118
                           119
000D                       120     start:
                           121
                           122             %reset_interconnect     ; RSTNC# Bit reset inside 8751
                           132                                     ; Microcontroller
                           133
                           134
                           135             %set_int_vector(08h,08h,interrupt_routine)
                           144
                           145     ; set up segment registers of cpu
                           146
0032 B8----          R     147             mov     AX,IODATSEG     ;LOAD SEGMENT REG
0035 8ED8                  148             mov     DS,AX           ;CS IS LOADED AUTOMATICALY BY RESET
0037 B8----          R     149             MOV     AX,EFGIO
003A 8EC0                  150             MOV     ES,AX
003C B8----          R     151             MOV     AX,STACKSEG
003F 8ED0                  152             MOV     SS,AX
0041 BCCB00          R     153             MOV     SP,OFFSET STACKTOP  ;SET SP  TO TOS
                           154
                           155             ; cpu now ready for work for the engine model
                           156
                           157             ; but first initialize the mpc
                           158
                           159
0044 9A0000----      E     160             call    mpc_init
                           161
                           162             ; then set to zero fuel variables
                           163
0049 26C706000000000 E     164             mov      es:ewfaold,0
0050 26C706000000000 E     165             mov      es:efuelused,0
0057 26C706000005046 E     166             mov      es:deltat,18000
                           167
                           168
                           169             ; then initialize master PIC of iSBC 286/100A
                           170
005E BAC000                171             mov     dx,pic_cntl0    ; icw1
0061 B017                  172             mov     al,00010111b    ; edge triggered, single mode, call
```

```
LOC  OBJ                  LINE     SOURCE

                          173                                    ; address
0063 EE                   174              out     dx,al
0064 BAC200               175              mov     dx,pic_cntl1  ; icw2
0067 B020                 176              mov     al,00100000b  ; base=type 32 (80h)
0069 EE                   177              out     dx,al
006A B01D                 178              mov     al,00011101b  ; special fully nested mode, buffered
                          179                                    ; master, normal eoi, 86 mode
006C EE                   180              out     dx,al
006D B0FB                 181              mov     al,0fbh       ; enable only ir2 message interrupts
006F EE                   182              out     dx,al
                          183
                          184              ; and finally enable cpu interrupts
                          185
0070 FB                   186              sti
                          187
                          188
0071 90                   189     idle:    nop
0072 90                   190              nop
0073 90                   191              nop                   ; do nothing but wait for message
```

```
iRMX 86 PL/M-86 V2.7 COMPILATION OF MODULE C286_INT_HANDLER
OBJECT MODULE PLACED IN C286I6.OBJ
COMPILER INVOKED BY:  :LANG:plm86 C286I6.P86 DEBUG SYMBOLS LARGE TYPE


  1          c286_int_handler:   do;


             /*
             main control interrupt handler for the local engine
             control. Will check for type of message, start up the model
             and carry out one simulation step at a time based on
             incomming messages.
             */

             /*
             history

             c286i6  v1.1    08.10.87    remoce error for std conditions
             c286i6  v1.0    03.10.87    clear mpc_message structure after transmit
                                         and after receiving for next time
             c286i5  v1.0    03.10.87    remove error in adressing mpc_messsage struc
             c286i4  v1.3    03.10.87    normal mode of ops for i2ice test
             c286i4  v1.22   21.08.87    engine and movb removed for test
             c286i4  v1.2    11.08.87    exchange source/dest for retransmit
             c286i4  v1.1    06.08.87    change data strutures
             c286i4  v1.0    06.08.87    change for std_conditions message
             c286i3  v1.1    04.08.87    add source id for the main processor
             c286i3  v1.0    04.08.87    add destination id
             c286i2  v1.0    04.08.87    remove types
             c286i1  v1.0    03.08.87    initial version

             */


  2   1      declare std_conditions word external;
  3   1      declare inputs word external;
  4   1      declare outputs word external;
  5   1      declare byte_dummy   byte;
  6   1      declare word_dummy   word;


  7   1      declare mpc_message structure
             ( dest           byte,              /* 4 Byte Hardware-Header */
               src            byte,
               type           byte,
               not_used       byte,
               prot_id        byte,              /* 8 Byte RMX-Header */
               transm_ctr     byte,
               dest_port_id   word,
               source_port_id word,
               transact_id    byte,
               transact_ctr   byte,
               message_data (20) byte) public; /* 20 Byte User-Data */


  8   1      declare index byte;

             /* Declaration of external procedures ********************/

  9   1      engine: procedure external;
 10   2      end engine;

 11   1      mpc_receive_message: procedure external;
 12   2      end mpc_receive_message;

 13   1      mpc_transmit_message: procedure external;
 14   2      end mpc_transmit_message;

             /* Start MPC-Interrupt-Procedure (MINT-Signal) **********/

 15   1      mpc_interrupt: procedure  public;

             /*

             first empty the MPC FIFO into mpc_message structure

             */

 16   2      call mpc_receive_message;


 17   2      if mpc_message.message_data(0)= 01h

                 then                /* we had a engine init message */
 18   2      do;
 19   3                call movb(@mpc_message.message_data(0),@std_conditions,20);

                       /* clear the mpc_message data structure after receive  */
 20   3                do index = 0 to 19;
 21   4                mpc_message.message_data(index)= 00h;
 22   4                end;
 23   3           end;
```

```
24  2          else              /* normal message for one simulation step */

                if mpc_message.message_data(0) = 10h then
25  2            do;
26  3              call movb (.mpc_message.message_data(0),.inputs,20);

27  3              call engine;      /* one simulation step */

28  3              call movb (.outputs, .mpc_message.message_data(0),20);


29  3             mpc_message.message_data(0) = 11h;    /*Set marker for result */

                  /* Transport Protocol adjustment to be
                     conforming to the Comm Layer of iRMX II*/

30  3             byte_dummy        = mpc_message.dest;
31  3             mpc_message.dest = mpc_message.src;
32  3             mpc_message.src  = byte_dummy;

33  3             word_dummy                = mpc_message.dest_port_id;
34  3             mpc_message.dest_port_id  = mpc_message.source_port_id;
35  3             mpc_message.source_port_id = word_dummy;

                 /* transmit results back to host */

36  3             call mpc_transmit_message;

                  /* clear the mpc_message data structure after transmit */
37  3               do index = 0 to 17;
38  4               mpc_message.message_data(index)= 00h;
39  4               end;
40  3             end;



41  2                                                    else
                  do;
                  /* Error-processing: Error in message_data(0)
                     error case not implemented in this version */
42  3             end;


43  2      end mpc_interrupt;


44  1      end;
```

```
iRMX 86 8086/87/88/186 MACRO ASSEMBLER V2.0 ASSEMBLY OF MODULE MPC_HANDLER_MODULE
OBJECT MODULE PLACED IN MPCDR3.OBJ
ASSEMBLER INVOKED BY:  :LANG:asm86 MPCDR3.A86 DEBUG SYMBOLS TYPE


LOC  OBJ                  LINE     SOURCE

                             1     name    mpc_handler_module
                             2
                             3     ;       mpcdr3  v1.5 CG/Scm 21.8.87     reference to mpc_message via 'seg'
                             4     ;       mpcdr3  v1.4 fh 07.08.87        change ds register for far call
                             5     ;       mpcdr3  v1.3 fh 06.08.87        change cl to cx for loop
                             6     ;       mpcdr3  v1.1 fh 06.08.87        rotate message id
                             7     ;       mpcdr3  v1.0 fh 05.08.87        add mctl register
                             8     ;       mpcdr2  v1.1 fh 04.08.87        make procedures far type
                             9     ;       mpcdr2  v1.0 fh 03.08.87        add mpc_init
                            10     ;       mpcdr1  v1.0 fh 22.06.87        add mpc control
                            11     ;       mpcdr0  v1.0 fh 19.06.86        original version
                            12
                            13
                            14     ; This module contains the low level MPC handling routines needed
                            15     ; for this application. Several simplifications are assumed:
                            16     ; the receive and transmit message routines are working on 32 byte
                            17     ; messages only. No length programming assumed. Further, the error
                            18     ; handling is not implemented in this first release of the low level
                            19     ; driver.
                            20     ; all three routines are a direct implementation of the work
                            21     ; flow described in the mpc user's guide and are coded in the large
                            22     ; model for compatibility with the engine model and the plm86
                            23     ; interrupt processing.
                            24     ;
                            25     ;
                            26
                            27
                            28 +1  $include(mpca86.def)
                   =1       29     ; i/o port addresses of MPC
                   =1       30
                   =1       31
  0010             =1       32     mdata   equ     10h     ; message data port
  001C             =1       33     mcmd    equ     1ch     ; message command port
  0000             =1       34     mstat   equ     00h     ; message status register
  0000             =1       35     mrst    equ     00h     ; message reset register
  000C             =1       36     mctl    equ     0ch     ; message control register
  0014             =1       37     merr    equ     14h     ; message error port
  0008             =1       38     mcon    equ     08h     ; message configuration register
                   =1       39
  0020             =1       40     msocmp  equ     20h     ; sol message completion port
  0024             =1       41     msicmp  equ     24h     ; sol message input completion port
  0020             =1       42     msocan  equ     20h     ; sol message output cancel port
  0024             =1       43     msican  equ     24h     ; sol message input cancel port
                   =1       44
  0004             =1       45     mid     equ     04h     ; message id register
                   =1       46
  0030             =1       47     icadrl  equ     30h     ; interconnect address low
  0034             =1       48     icadrhi equ     34h     ; interconnect address high1
  0031             =1       49     icadrh2 equ     31h     ; interconnect address high2
  003C             =1       50     icdata  equ     3ch     ; interconnect data register
                   =1       51
                   =1       52     ; bit definitions
                   =1       53
  0001             =1       54     xmtnf   equ     1               ; transmit FIFO not full
  0004             =1       55     xmterr  equ     4               ; transmit error on PSB
  0083             =1       56     rcvne   equ     10000011b       ; receive FIFO not empty
                   =1       57
  0080             =1       58     init_done       equ     80h     ; filter for bit 7
  0000             =1       59     zero            equ     00h
  0085             =1       60     width_config    equ     10000101b       ; set 16 bit fifo and dma
                   =1       61                                             ; full message passing
                   =1       62
  0002             =1       63     mpc_control     equ     00000010b       ; receive interrupt enable
                            64
                            65     extrn   mpc_message:word
                            66
                            67
----                        68     code    segment
                            69     assume  cs:code
                            70     assume  ds:seg mpc_message
                            71
                            72     public  mpc_receive_message,mpc_transmit_message,mpc_init
                            73
                            74
0000                        75     mpc_transmit_message    proc far
                            76
                            77     ; v1.1 mpc_write_message                add transmit FIFO control
                            78     ; v1.0 mpc_write_message                basic priciple written
                            79
                            80     ; mpc_transmit message will copy the data from the mpc_message
                            81     ; structure to the fifo of the mpc and send it out to the iPSB.
                            82     ; 32 byte length assumed.
                            83
0000 1E                     84             push    ds
0001 B8----          E      85             mov     ax,seg mpc_message
0004 8ED8                   86             mov     ds,ax
                            87
0006 B91000                 88             mov     cx,16           ; 32 byte message
                            89                                     ; 16 words
0009 BA1000                 90             mov     dx,mdata        ; data port of mpc
000C BE0000          E      91             mov     si, offset mpc_message
                            92
```

```
LOC  OBJ                  LINE     SOURCE

000F BB04                   93     write_1: mov     ax, [si]        ; load data to ax
0011 EF                     94              out     dx,ax           ; store to mpc
0012 46                     95              inc     si
0013 46                     96              inc     si              ; double to next word
0014 E2F9                   97              loop    write_1         ; continue till all 32 byte
                            98                                      ; writte to MPC FIFO
                            99
0016 32C0                  100              xor     al,al           ;
0018 E61C                  101              out     mcmd,al         ; write 00 to MPC CMD Register
                           102
001A                       103     continue:
                           104
001A E400                  105              in      al,mstat        ; read MPC status_register
001C 2401                  106              and     al,xmtnf        ; mask transmit fifo bit
001E 3C01                  107              cmp     al,1
0020 7409                  108              je      transmit_complete       ; message transmitted '
0022 2404                  109              and     al,xmterr       ; mask transmit error bit
0024 3C00                  110              cmp     al,0
0026 74F2                  111              je      continue        ; message transmition still in progress
                           112
0028 E85600                113              call    transmit_error  ; error handling
                           114
002B                       115     transmit_complete:
                           116
002B 1F                    117              pop     ds
002C CB                    118              ret
                           119
                           120     mpc_transmit_message    endp
                           121
                           122 +1  $eject
                           123
                           124
                           125
002D                       126     mpc_receive_message     proc far
                           127
                           128     ; v1.1 mpc_read_message         add receive FIFO control
                           129     ; v1.0 mpc_read_message         basic priciple written
                           130
                           131     ; mpc_receive_message will copy the data to the mpc_message
                           132     ; structure from the fifo of the mpc.
                           133     ; 32 byte length assumed.
                           134
002D 1E                    135              push    ds
002E B8----          E     136              mov     ax,seg mpc_message
0031 8ED8                  137              mov     ds,ax
                           138
0033                       139     not_ready_1:
                           140
0033 E400                  141              in      al,mstat        ; 8 bit read of MPC_status_register
0035 3C83                  142              cmp     al,rcvne
0037 75FA                  143              jne     not_ready_1     ; wait until flag set
                           144
0039 E410                  145              in      al,mdata        ; read one byte for length
                           146
                           147     ; read mpc_fifo  assuming 32 byte valid data and 16 bit transfers
                           148     ; initialized inside the MPC
                           149
003B B91000                150              mov     cx,16           ; 32 byte message
                           151                                      ; 16 words
003E BA1000                152              mov     dx,mdata        ; data port of mpc
0041 BF0000          E     153              mov     di, offset mpc_message
0044 ED                    154     read_2:  in      ax,dx           ; read from mpc
0045 8905                  155              mov     [di],ax         ; store data to message buffer
0047 47                    156              inc     di
0048 47                    157              inc     di              ; double to next word
0049 E2F9                  158              loop    read_2          ; continue till all 32 byte
                           159                                      ; read
                           160
004B E41C                  161              in      al,mcmd         ; read MPC_command port for resync
                           162
                           163                                      ; service of message read now complete
                           164
                           165
004D 1F                    166              pop     ds
                           167                                      ; continue with processing of data
004E CB                    168              ret
                           169
                           170     mpc_receive_message     endp
                           171
                           172
                           173 +1  $eject
                           174
                           175
004F                       176     mpc_init        proc far
                           177
                           178
                           179     ; this routine will initialize the MPC for use in this application.
                           180
004F 1E                    181              push    ds
0050 B8----          E     182              mov     ax,seg mpc_message
0053 8ED8                  183              mov     ds,ax
                           184
```

```
LOC  OBJ                  LINE     SOURCE

                           185
0055 B000                  186              mov     al,zero
0057 E600                  187              out     mrst,al          ; reset the mpc
                           188
0059                       189     chck_reset:
0059 E400                  190              in      al,mstat
005B 2480                  191              and     al,init_done
005D 3C00                  192              cmp     al,zero
005F 75F8                  193              jne     chck_reset       ; do whil not reset
                           194
0061 B085                  195              mov     al,width_config
0063 E608                  196              out     mcon,al          ; set to 16 bit
0065 B002                  197              mov     al,mpc_control   ; interrupt control
0067 E60C                  198              out     mctl,al
0069 E408                  199              in      al,mcon          ; no check for error case this time
006B B0F8                  200              mov     al,0f8h
006D E634                  201              out     icadrh1,al       ; interconnect address high
006F E631                  202              out     icadrh2,al       ; interconnect address high
0071 B098                  203              mov     al,098h
0073 E630                  204              out     icadrl ,al       ; interconnect address low
0075 E43C                  205              in      al,3ch           ; read interconnect register slot_id
0077 D0E8                  206              shr     al,1
0079 D0E8                  207              shr     al,1
007B D0E8                  208              shr     al,1
007D E604                  209              out     mid,al           ; set the message id to slot_id
                           210
007F 1F                    211              pop     ds
0080 CB                    212              ret
                           213
                           214
                           215
                           216     mpc_init         endp
                           217
                           218
0081                       219     transmit_error   proc
                           220
                           221     ; this routine is  n o t  implemented in this release of
                           222     ; this handler
                           223     transmit_error   endp
                           224
                           225
----                       226     code     ends
                           227
                           228
```

```
iRMX 86 0086/87/88/186 MACRO ASSEMBLER V2.0 ASSEMBLY OF MODULE CF6_ENGINE_LOCAL
OBJECT MODULE PLACED IN CF650L.OBJ
ASSEMBLER INVOKED BY:  iLANG:asm86 CF650L.A86 DEBUG SYMBOLS TYPE

LOC  OBJ                  LINE     SOURCE

                             1     NAME CF6_engine_local
                             2
                             3
                             4     ;                    ****************************************
                             5     ;                    ****************************************
                             6     ;                    **                                    **
                             7     ;                    **      ENGINE   MAIN-PROGRAM         **
                             8     ;                    **                                    **
                             9     ;                    ****************************************
                            10     ;                    ****************************************
                            11
                            12     ;             PROJECT      CF6-50C Multicomputer
                            13
                            14     ;             03.10.87   v1.0    put n2lim as a constant inside routine
                            15     ;             11.08.87   v2.2cg  correct flags
                            16     ;             04.08.87   v2.1fh  make engine procedure far
                            17     ;             03.08.87   v2.0fh  remove local start
                            18     ;             21.07.87   v1.2fh  add for main module test version
                            19     ;             19.06.87   v1.1fh  adapt to local processor architecture
                            20     ;             19.06.87   v1.0fh  add message structure
                            21
                            22
                            23     ;       PROGRAM FOR THE SIMULATION OF THE CF6-50C ENGINE. FOLLOWING
                            24     ;       SUBSYTEM MODULES ARE USED:
                            25     ;                          EPR-SYSTEM
                            26     ;                          EGT-SYSTEM
                            27     ;                          FUEL-FLOW-SYSTEM
                            28     ;                          THRUST-SYSTEM
                            29     ;                          N1-SYSTEM
                            30     ;                          INLET SYSTEM
                            31
                            32     ;       SEE ALSO ADDITIONAL DOCUMENTATION IN THE WRITTEN DOCUMENTS
                            33     ;       SIMULATION OF A JET ENGINE USING A MICROCOMPUTER
                            34     ;       DIPLOMA WORK        WS 1979/80
                            35
                            36     ;*******************************************************************************
                            37     ;
                            38     ;     THIS FROGAM USES FOLLOWING EXTERNAL MODULE: CF6LIB.LIB(SYSTEM-MODULE),
                            39     ;       1.CF6LIB.LIB (SYSTEM-MODULE)
                            40     ;       2.TABLIB.LIB (DATA FUNCTIONS-TABLES )
                            41     ;*******************************************************************************
                            42
                            43
                            44
                            45     ; note that all written in CAPITAL is original software written
                            46     ; end of 1979, early 1980 '
                            47     ; all other is modification done to port the original application
                            48     ; to Multibus II
                            49
                            50     public std_conditions
                            51
                            52     PUBLIC EVM,EVML,EVMH,EALT,EALTL,EALTH,EP0,ET0,ENGN,INPUTS
                            53     PUBLIC ET2,ED2,ETHE,EDAM,ERTHE,DEPR,DEGT,DEWFA,DN1,DFNA
                            54
                            55     PUBLIC EN2,EN2L,EN2H,OUTPUTS,EN1,EN1L,EN1H,EN1A
                            56     PUBLIC EPR,EGT,EWFA,EFUELUSED,EFNA
                            57     PUBLIC EACPSW,EHPXSW,ETAISW,ECAISW,EBLESW,EREVSW
                            58     PUBLIC EWFAOLD,DELTAT
                            59
                            60     ;            ****************************
                            61     ;            *                          *
                            62     ;            *  DATA VARIABLES          *
                            63     ;            *                          *
                            64     ;            ****************************
                            65
                            66     EFGID   SEGMENT COMMON
                            67
                            68     ;          ***************************************
                            69     ;          *  WORK DATA OF ENGINE MODEL          *
                            70     ;          ***************************************
                            71
                            72     ; global engine input variables
                            73
                            74     ; NOTE THE USE OF ACCESS BOTH AS BYTE AND WORD '
                            75
  0000                      76     std_conditions  equu     this word
0000 (1                     77                     dw      1       dup(0)                  ; Flag-Word only used to sync
                                   hronize
     0000
     )
                            78                                                                     ; sim step from initi
                                   alization
                            79
  0002                      80     EVM     EQU     THIS WORD          ;MACH               (INPUT)
0002 00                     81     EVML    DB      0                  ;                     "
0003 00                     82     EVMH    DB      0                  ;                     "
  0004                      83     EALT    EQU     THIS WORD          ;ALTITUDE             "
0004 00                     84     EALTL   DB      0                  ;                     "
0005 00                     85     EALTH   DB      0                  ;                     "
0006 0000                   86     EP0     DW      0                  ;AMBIENT PRESSURE     "
0008 0000                   87     ET0     DW      0                  ;AMBIENT  TEMPERATURE "
000A (10                    88     reserved        DW  10 dup(0)          ;10 byte reserved (multibus II
     0000
     )
```

```
LOC  OBJ                  LINE     SOURCE

                           89                                                    ; data strucutre o
                                    f the message '
                           90
                           91
                           92      ; local engine variables
                           93
001E 0000                  94      ET2             DW      0
0020 0000                  95      ED2             DW      0
0022 0000                  96      ETHE    DW      0
0024 0000                  97      EDAM    DW      0
0026 0000                  98      ERTHE   DW      0
                           99              ;
                          100      ;TEMP VARIABLES
0028 0000                 101      DFPR    DW      0
002A 0000                 102      DEGT    DW      0
002C 0000                 103      DEWFA   DW      0
002E 0000                 104      DN1             DW      0
0030 0000                 105      DFNA    DW      0
                          106              ;
                          107              ;N2 LOW LIMIT CONSTANT = 80
                          108
0032 FF0F                 109      EN2LIM  DW      4095    ;SCALED VALUE FOR LOWER LIMIT N2
                          110      ;
                          111      ;
                          112
                          113
                          114 +1   $eject
                          115
                          116      ; message data buffers for MPC communication
                          117
  0034                    118      INPUTS EQU      THIS WORD       ;                       "
0034 (1                   119                      dw              1 dup(0)        ; Flag-Word only
     0000
     )
                          120
  0036                    121      EN2     EQU     THIS WORD       ;REFERRED CORE SPEED    "
0036 00                   122      EN2L    DB      0               ;                       "
0037 00                   123      EN2H    DB      0               ;                       "
0038 0000                 124      EACPSW  DW      0               ;AIR COND. SWITCH       "
003A 0000                 125      EHPXSW  DW      0               ;HORSE POWER EX. SWITCH "
003C 0000                 126      EFAISW  DW      0               ;FULL ANTI ICE SWITCH   "
003E 0000                 127      ECAISW  DW      0               ;COWL ANTI ICE SWITCH   "
0040 0000                 128      ERLESW  DW      0               ;BLEED SWITCH           "
0042 0000                 129      EREVSW  DW      0               ;REVERSER SWITCH        "
0044 (2                   130      in_reserved     dw   2 dup(0)           ; reserved for future use
     0000
     )
                          131
                          132
                          133
                          134
0048 0000                 135      ENGN    DW      0               ;TRIEBWERKNUMMER        "
                          136
                          137
                          138
                          139
  004A                    140      OUTPUTS EQU     THIS WORD
004A (1                   141                      dw          1 dup(0)            ; Flag-word only
     0000
     )
                          142
004C 0000                 143      EWFAOLD DW      0               ;"OLD VALUE FUEL FLOW           OUTPUT
004E 0000                 144      DELTAT  DW      0               ; SAMPLING TIME           "
0050 0000                 145      EFUELUSED       DW      0       ; FUEL USED               "
  0052                    146      EN1     EQU     THIS WORD       ; REFERD FAN SPEEDD       "
0052 00                   147      EN1L    DB      0
0053 00                   148      EN1H    DB      0
0054 0000                 149      EN1A    DW      0               ; N1 DIGITAL DSIPLAY      "
0056 0000                 150      EPR     DW      0               ; ENG.PRESSURE RATIO       "
0058 0000                 151      EGT     DW      0               ; EXHAUST GAS TEMP.        "
005A 0000                 152      EWFA    DW      0               ; FUEL-FLOW                "
005C 0000                 153      EFNA    DW      0               ; THRUST                   "
                          154
                          155
005E 0000                 156      EFUELUSED1      dw      0
0060 0000                 157      EFUELUSED2      dw      0
0062 0000                 158      EFUELUSED3      dw      0
                          159
----                      160      EFGIO   ENDS
                          161
                          162 +1   $eject
                          163
                          164
----                      165      STACKSEG                SEGMENT STACK
0000 (100                 166                      DW      100 DUP(?)      ;ALLOCATE 100 WORDS FOR STACK
     ????
     )
00C8                      167      STACKTOP        LABEL   WORD            ;OFFSET ADDRESS OF THE TOS
----                      168      STACKSEG                ENDS
                          169
----                      170      iodatseg                SEGMENT public
                          171
0000 0000                 172      TIME50  dw      0
0002 00                   173      NOFUEL  db      0
                          174
----                      175      iodatseg                ENDS
                          176
                          177
```

```
LOC  OBJ                  LINE     SOURCE

----                       178     EFGCOD          SEGMENT PUBLIC
                           179
                           180             ;GIVE ASSEMBLER INITIAL REGISTER TO SEGMENT CORRESPONDENCE
                           181
                           182     ASSUME  CS: EFGCOD,     DS: IODATSEG,   ES: EFGID,      SS: STACKSEG
                           183
                           184     public  engine
                           185
                           186
                           187
                           188
                           189
                           190     EXTRN INLETSYS:NEAR,EPRSYS:NEAR,EGTSYS:NEAR,N1SYS:NEAR,EWFASYS:NEAR
                           191     EXTRN FNASYS:NEAR
                           192
                           193             ASSUME ES:EFGID,CS:EFGCOD
                           194
                           195     ;                       ****************************
                           196     ;                       *                          *
                           197     ;                       * ENGINE MODEL ROUTINE    *
                           198     ;                       *                          *
                           199     ;                       ****************************
                           200
                           201
0000                       202     ENGINE          PROC    far
                           203
0000                       204     START_engine:
0000 06                    205                             PUSH    ES
0001 1E                    206                     PUSH    DS          ;ES,DS SAVE ONTO STACK
0002 B8----          R     207                     MOV     AX,EFGID  ; REORGANIZE SEGEMENTS FOR THE
                           208                                                     ; OPTIMAL USE OF THE CPU ;
                               SPEED AND
                           209                                                     ; AGAIN SPEED....
0005 8EC0                  210                     MOV     ES,AX
0007 E80000          E     211                     CALL    INLETSYS  ;CALCULATE THE INLET VALUES
                           212
                           213
                           214
                           215
                           216     ;               ******************
                           217     ;               *    ENGINE    *
                           218     ;               ******************
                           219
                           220
000A 26A13600              221                     MOV     AX,EN2
000E 3DFF0F                222                     CMP     AX,4095         ; >>> put absolute value here
                           223                                                     ; due to EPROM version
0011 7706                  224                     JA      NORMAL2
0013 EB1500                225                     CALL    IDLEBYPASS      ; IN IDLE BYPASS MODEL CALCULATION
                           226                                                     ; BASED ON A SIMPLIFICATION A
                               SSUPTION
0016 EB1090                227                     JMP     idle_state
                           228
                           229                     ; NORMAL NON IDLE MODE : CALCULATE A NEW STATE OF THE ENGINE
                           230                     ; SUBSYSTEMS
                           231
0019 E80000          E     232     NORMAL2:        CALL    EPRSYS          ; EPR
001C E80000          E     233                     CALL    N1SYS           ; N1
001F E80000          E     234                     CALL    EWFASYS         ; FUEL FLOW
0022 E80000          E     235                     CALL    EGTSYS          ; EGT
0025 E80000          E     236                     CALL    FNASYS          ; THRUST
                           237
0028                       238     idle_state :
                           239
0028 1F                    240                     POP     DS
0029 07                    241                     POP     ES
002A CB                    242                     RET
                           243     ENGINE  ENDP
                           244
                           245     ;                       *****************
                           246     ;                       *  END ENGINE   *
                           247     ;                                       *****************
                           248
002B                       249     IDLEBYPASS      PROC
                           250
002B 50                    251             PUSH    AX              ;SAVE EN2
002C 26C706360000 10       252             MOV     EN2,4096
                           253
0033 E80000          E     254             CALL    EPRSYS          ;N1SYS SIMPLIFICATION ASSUMPTION
0036 E80000          E     255             CALL    EWFASYS
0039 E80000          E     256             CALL    EGTSYS
003C E80000          E     257             CALL    FNASYS
                           258
003F 268F063600            259             POP     EN2             ;RECALL EN2
0044 C3                    260             RET
                           261
                           262     IDLEBYPASS      ENDP
                           263
                           264
                           265
----                       266     EFGCOD  ENDS
                           267             END
```

```
MODULE = FUELFLOW_SYSTEM

BASE    OFFSET TYPE SYMBOL              BASE    OFFSET TYPE SYMBOL              BASE    OFFSET TYPE SYMBOL

0041H   0A10H  SYM  CONT1               0041H   0A33H  SYM  CONT2               0041H   0C0BH  SYM  ENDE
0041H   0B25H  SYM  EWFAC              0041H   09C0H  SYM  EWFASYS             0041H   0A06H  SYM  NEG1
0041H   0A29H  SYM  NEG2              0041H   09C0H  SYM  START              0041H   0AABH  SYM  TACP
0041H   0ACEH  SYM  TCAI              0041H   0A6CH  SYM  TFAI               0041H   0A8DH  SYM  THPX


MEMORY MAP OF MODULE MAIN_MODULE_286

MODULE START ADDRESS  PARAGRAPH = 0041H  OFFSET = 000DH
SEGMENT MAP

START     STOP      LENGTH ALIGN NAME             CLASS           OVERLAY

00000H    0009FH    00A0H   A   INT_POINTERS
00200H    00200H    0000H   G   ??SEG
00200H    00263H    0064H   G   EFGIO
00270H    003FFH    0190H   G   STACK_SEG
00400H    00402H    0003H   G   IODATSEG
00410H    010C1H    0C12H   G   EFGCOD
01030H    010B0H    0081H   G   CODE
010C0H    0243FH    1380H   G   EFGDAT
02440H    0251EH    00DFH   W   C286_INT_HANDL_  CODE
                                -ER_CODE
02520H    02543H    0024H   W   C286_INT_HANDL_  DATA
                                -ER_DATA
02544H    0254BH    0008H   W   STACK            STACK
0254CH    0254CH    0000H   W   MEMORY           MEMORY
```

APPENDIX E

E1. I2ICE debug session sample

This shows a sample debug session using symbolic debugging with I2ICE 286.

```
First include all user pre-defined macros for this application
*
*
*include cf6.mac

std_def is a macro for setting the standard values for local
engine test

*define proc std_def = do
.*evm=0t
.*ep0=20783t
.*et0=303t
.*deltat=18000t
.*end
*
mpc_msg is a macro for displaying the content of the MPC_MESSAGE
STRUCTURE

*define  proc mpc_msg = do
.*write ' '
.*write ' dest     = ', mpc_message.dest
.*write ' src      = ', mpc_message.src
.*write ' type     = ', mpc_message.type
.*write ' not_used = ', mpc_message.not_used
.*write ' transport message passing part '
.*write ' protocol_id       = ', byte .mpc_message.prot_id
.*write ' transmition _cntl = ', byte .mpc_message.transm_ctr
.*write ' dest port_id      = ', word .mpc_message.dest_port_id
.*write ' src_port_id       = ', word .mpc_message.sourc_port_id
.*write ' transaction_id    = ', byte .mpc_message.transact_id
.*write ' transaction_cntl  = ', byte .mpc_message.transact_ctr
.*write ' data field = '
.*word .mpc_message+12t length 10t
.*end

macro s makes a single step

*define proc s = do
.*istep
.*ASM $
.*end
*

macro input_cf6 displays the inputs data structure
before and engine simulation step
```

```
*define proc input_cf6 = do
.*write 'flag_word =',:cf6_engine_local.inputs
.*write 'EN2 =',:cf6_engine_local.EN2
.*write 'EN2L =',:cf6_engine_local.EN2L
.*write 'EN2H =',:cf6_engine_local.EN2H
.*write 'EACPSW =',:cf6_engine_local.EACPSW
.*write 'EHPXSW =',:cf6_engine_local.EHPXSW
.*write 'EFAISW =',:cf6_engine_local.EFAISW
.*write 'ECAISW =',:cf6_engine_local.ECAISW
.*write 'EBLESW =',:cf6_engine_local.EBLESW
.*write 'EREVSW =',:cf6_engine_local.EREVSW
.*end

*
```

macro output_cf6 displays the outputs data structure as a result of a simulation step

```
*
*define proc output_cf6 = do
.*write 'flag_word =',:cf6_engine_local.outputs
.*write 'EWFAOLD =',:cf6_engine_local.EWFAOLD
.*write 'DELTAT =',:cf6_engine_local.DELTAT
.*write 'EFUELUSED =',:cf6_engine_local.EFUELUSED
.*write 'EN1  =',:cf6_engine_local.EN1
.*write 'EN1L  =',:cf6_engine_local.EN1L
.*write 'EN1H  =',:cf6_engine_local.EN1H
.*write 'EN1A  =',:cf6_engine_local.EN1A
.*write 'EPR =',:cf6_engine_local.EPR
.*write 'EGT =',:cf6_engine_local.EGT
.*write 'EWFA =',:cf6_engine_local.EWFA
.*write 'EFNA =',:cf6_engine_local.EFNA
.*end
*
```

enginestate decode the full state of variables of the engine model at any time

```
*DEFINE proc ENGINESTATE = do
.*BASE=decimal
.*WRITE 'N1 =  ',EN1, '  non scaled :',(en1a/1696t)*20t
.*WRITE 'EGT = ',EGT, '  non scaled :',egt
.*WRITE 'EPR = ',EPR, '  non scaled :',epr/3200t
.*WRITE 'N2 =  ',EN2, '  non scaled :',(en2/256t)*5t
.*WRITE 'FF =  ',EWFA,'  non scaled :',ewfa
.*WRITE 'FNA = ',EFNA,'  non scaled :',(efna/378t)*1000t
.*WRITE 'INLET ::::   EP0=',EP0,' ET0= ',ET0,' MACH = ',EVM
.*write ' '
.*write ' switch status '
.*write 'acp =',EACPSW,' hpx =',EHPXSW,' fai =',EFAISW
.*write 'cai =',ECAISW,' ble =',EBLESW,' reverse = ',EREVSW
.*write ' fuel used for this step = ',efuelused
.*BASE=Hex
.*end
```

```
macro setn2 sets the en2 variable for a local test
*
*DEFINE proc SETN2 = do
.*word .mpc_message+14t = %0
.*end
*

macro rst allows the reset not complete sequence to take place
see also main text

*define proc rst = do
.*reset regs
.*unithold
.*port(34h)=0f8h
.*port(31h)=0f8h
.*port(30h)=0a4h
.*port(3ch)=000h
.*go from 0ffff:0 forever
.*end
*


make a break at the end of the simulation step

*go til :c286_int_handler#43
*Probe 0 stopped at :C286_INT_HANDLER#41 + 1H because of execute
break
  Clips= F3  Trace Buffer Overflow

see what data are around after a full engine simulation step
using i2ice macros

incoming data packed in mpc_message structure as before
engine simulation step done

message strucuture as received by the MPC

*mpc_msg

 dest     =  01
 src      =  05
 type     =  00
 not_used =  00
 transport message passing part
 protocol_id       =  02
 transmition _cntl =  00
 dest port_id      =  0010
 src_port_id       =  0800
 transaction_id    =  00
 transaction_cntl  =  00
 data field =
0256:000EH  0010 14E0 0000 0001 0001 0000 0000 0000 0000 0000
```

input data

```
*input_cf6
flag_word = 0010
EN2 = 14E0
EN2L = E0
EN2H = 14
EACPSW = 0000
EHPXSW = 0001
EFAISW = 0001
ECAISW = 0000
EBLESW = 0000
EREVSW = 0000
```

output data

```
*
*output_cf6
flag_word = 0000
EWFAOLD = 1411
DELTAT = 29C7
EFUELUSED = 001B
EN1  = 2CDB
EN1L  = DB
EN1H  = 2C
EN1A  = 2605
EPR = 5346
EGT = 035B
EWFA = 5047
EFNA = 4B0A
```

full engine state

```
*
*enginestate
N1 =   11483   non scaled : 100
EGT =  859   non scaled : 859
EPR =  21318   non scaled : 6
N2 =   5344   non scaled : 100
FF =   20551   non scaled : 20551
FNA =  19210   non scaled : 50000
INLET ::::   EP0= 20783  ET0=  303  MACH =  0

 switch status
acp = 0  hpx = 1  fai = 1
cai = 0  ble = 0  reverse =  0
 fuel used for this step =  27
*
```

```
continue the simulation
*
*go forever
?
?
?
/* end of this debug session */
```

APPENDIX E

E2. iPAT analysis protocol

This shows the sample iPAT analysis session done to assess the performance of the application processor design.

```
*
*/* -------ipat analysis    -------*/
*
set the time base to 200 nsec in general
note that we are online with the simulation running

*go
?ptimebase = 200ns

define a macro for analysing

?
?define proc measure = do
.?pat init duration %0
.?histo=false
.?pat display
.?end
?

measure the time spendt in the mpc_receive_message routine

?measure (:mpc_handler_module.mpc_receive_message)

Mode: DURATION
Event: :MPC_HA.MPC_RECEIVE_MESSAGE
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:         0:     0 ns]
   65 us-  447 us+:        58: 5165 us]
------------------+----------+--------]
Time Min:    89.0 us
Time Max:    89.2 us
?
```

measure the time spendt in the mpc_transmit_message routine

```
?measure (:mpc_handler_module.mpc_transmit_message)

Mode: DURATION
Event: :MPC_HA.MPC_TRANSMIT_MESSAGE
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:         0:     0 ns]
   65 us-  447 us+:        58: 5316 us]
------------------+----------+--------]
Time Min:   91.6 us
Time Max:   91.8 us
?
```

measure the time spendt in the whole interrupt routine

```
?
?measure (:c286_int_handler)

Mode: DURATION
Event: :C286_INT_HANDLER
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:         0:     0 ns]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min: 1409.2 us
Time Max: 1412.8 us
?
```

measure the time spendt in the engine model routine
note that the engine model calculation time is variable depending on the switch position of the subsystems

```
?measure (:cf6_engine_local.engine)
```

```
Mode: DURATION
Event: :CF6_EN.ENGINE
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:         0:     0 ns]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min:  633.0 us
Time Max:  636.6 us
?
```

measure the time spendt for copying the data from the mpc_message_structure to the inputs buffer

```
?
?pat init duration :c286_int_handler#26 to :c286_int_handler#27
?pat display

Mode: DURATION
Event: :C286_INT_HAN#26-#27
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:        62: 1895 us]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min:   30.4 us
Time Max:   30.6 us

?
```

measure the time spendt for the transport protocol adaptation

```
?pat init duration :c286_int_handler#30 to :c286_int_handler#35
?pat display

Mode: DURATION
Event: :C286_INT_HAN#30-#35
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK
```

```
    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:        50:   873 us]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min:    17.4 us
Time Max:    17.6 us
?
```

measure the total time used for the mpc interrupt processing from hardware interrupt on.

```
Mode: DURATION
Event: :MAIN_M.INTERRUPT_ROUTINE
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:         0:     0 ns]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min: 1450.0 us
Time Max: 1453.6 us
```

profile the relative time spend by the three main routines

```
?
?pat init profile :mpc_handler_module.mpc_receive_message,&
??:mpc_handler_module.mpc_transmit_message,&
??:cf6_engine_local.engine
?pat display

Mode: PROFILE
PTIMEBASE =  10 us
Include calls
Status: OK
```

```
Event              :   Count  0%        10         20
---------------------------+---------+---------+-------
ENGINE             :         61:xxxxxxxxxxxxxxxx
MPC_TRANSMIT_ME    :         61:xxxxxxxxxxxxxxxx
MPC_RECEIVE_MES    :         61:xxxxxxxxxxxxxxxx
*Background*       :        182:xxxxxxxxxxxxxxxxxxxxxxxxxxxx
---------------------------+---------+---------+-------
      Total        :        365 0%        10         20
?histo=false
?pat display

Mode: PROFILE
PTIMEBASE =  10 us
Include calls
Status: OK

Event                    : Count :  Time:Time Min:Time Avg:Time Max
--------------------------+-------+-------+--------+--------+--
:CF6_EN.ENGINE           : 444 :  318 ms:  710 us:  710 us:  720 us
:MPC_HA.MPC_TRANSMIT     : 444 :   41 ms:   90 us:   90 us:  100 us
MPC_RECEIVE_MESSAGE      : 444 :   39 ms:   80 us:   80 us:   90 us
*Background*             : 1331:   22sec:   40 us:   16 ms:   50 ms
--------------------------------+-------+--------+--------+--
      Total                     :      2663:   22sec
?
?
```

make the interrupt to routine activation measurents

mpc interrupt to start of the interrupt routine

```
?pat init duration interrupt to :c286_int_handler.mpc_interrupt
?histo = false
?pat display

Mode: DURATION
Event: *INT*-:C286_INT_HANDLER#15
Time Range: 10us TO 1sec

PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+---------+--------]
        <   10 us :         0:     0 ns]
   10 us-   11 us+:         0:     0 ns]
   11 us-   19 us+:        70:   919 us]
   19 us-   75 us+:         0:     0 ns]
   75 us-  457 us+:         0:     0 ns]
------------------+---------+--------]
Time Min:   12.8 us
Time Max:   13.6 us
```

mpc interrupt to start of mpc_receive_message

```
?pat init duration interrupt to
:mpc_handler_module.mpc_receive_message
?pat display

Mode: DURATION
Event: *INT*-00109DH
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
       <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:        71:  1523 us]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min:   21.0 us
Time Max:   22.0 us
?
?
?
?/* end of test session */
?
?halt
*Probe 0 stopped at :MAIN_MODULE_286 + 74H because of halt
  Clips= F6  Trace Buffer Overflow
*exit
 I2ICE terminated
```

intel

APPLICATION NOTE

AP-433

May 1989

Simple I/O Design Example using the MULTIBUS® II Modular Interface eXtension (MIX) Architecture

ERIK A. STEEB
OMSO TECHNICAL MARKETING ENGINEER

Order Number 281004-001

PURPOSE

This application note introduces the MULTIBUS® II Modular Interface eXtension (MIX) architecture through the use of a simple I/O design example. The example is intended as a "design primer" to assist an engineer in implementing a custom I/O design on the MIX interface. It is assumed the reader is familiar with MIX architecture concepts, MULTIBUS II and the MULTIBUS II System Architecture (MSA).

RELATED DOCUMENTS

MIX Module Design Specification	order number 500729-001
MIX 386/MOxPP Baseboard User's Guide	order number 500730-001.

1. INTRODUCTION

Since its introduction, MULTIBUS® II has proven to be a very good solution for high performance computing. The bus is well suited high performance applications due to its ability to support multiple CPUs in a loosely coupled environment. This capability is not only borne in hardware but in the firmware and software as well.

In this loosely coupled multiprocessing environment, many CPUs are able to interoperate on a common backplane. With this ability comes the need for a system design which is separated along functional boundaries. This means particular I/O needs of the system are handled in a client/server fashion instead of the traditional master/slave method. The difference in the two approaches lies in the ability of the I/O server to support more than one host processor while the slave cannot.

The use of a functionally partitioned system allows for better structured programming and ease of upgrading system resources. However, the designers of special I/O are faced with the challenge of more complex board designs. For MULTIBUS II this means not only designing a board which supports message passing and the multiple address spaces defined for the bus, but also one with an intelligent CPU core and a high speed memory subsystem. These added requirements not only increase the design complexity but can also greatly affect the time to market of a product.

Intel has developed an I/O strategy which eliminates the burden of developing the compute engine and PSB interface of the I/O server, therefore easing the problems associated with designing special I/O for MULTIBUS II. This is done by decoupling the specific I/O technology from the CPU technology in a baseboard/module fashion as shown in Figure 1. With this design, an Intel baseboard is coupled with an I/O module supplied by Intel, the customer, or a third party. Although the baseboard and module are separate boards, the combination forms a single slot I/O server for MULTIBUS II.

The modular design allows an engineer to focus his efforts on the I/O needs of a product by integrating his special I/O design with an intelligent MULTIBUS II I/O platform supplied by Intel. This platform provides the necessary computational power required of an I/O server but leaves the special I/O module design to the customer.

The strategy outlined above is known as the Modular Interface eXtension (MIX) architecture. This document gives an application example and illustrates a simple I/O design which utilizes the MULTIBUS II MIX architecture.

2. MIX APPLICATION EXAMPLE

Because the MIX architecture allows stacking of up to three MIX modules on a high performance baseboard, price/performance scalability is offered along with ease of design for MULTIBUS II, ease of upgrade potential, and support for Intel operating systems. Therefore, a MIX design fits well in virtually any MULTIBUS II application.

The use of MIX however is best suited for those applications which require high performance and/or a large number of I/O

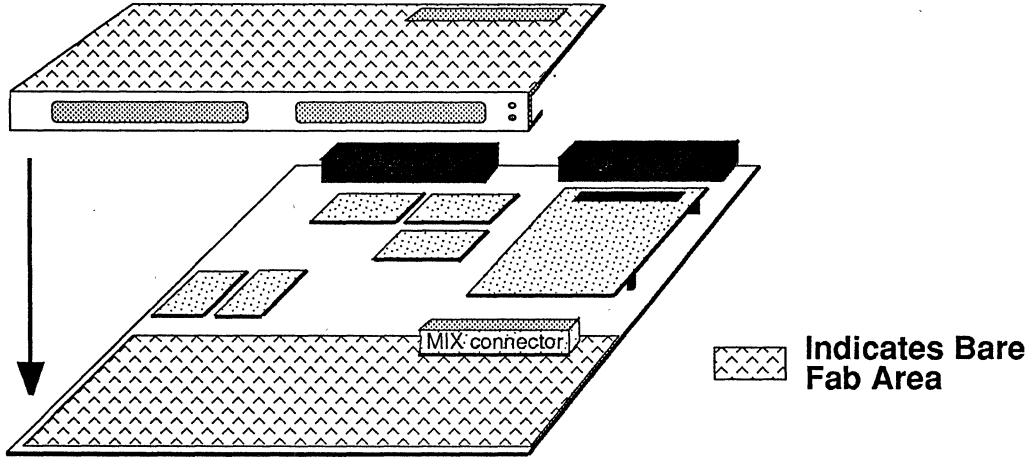


Figure 1. MIX Baseboard/Module Combination

connections which, when combined, demand CPU bandwidth. The following example is given to illustrate the demands which the MIX concept is designed to address.

2.1 Flight Simulation System

In flight simulation system designs, many computationally intensive simulation tasks exist which demand very high performance and multiprocessing capabilities to achieve the necessary real-time I/O processing. These requirements stem from the system's need to accurately and quickly convert pilot and aircraft action into control and instrumentation reaction.

This application requires plenty of computational muscle, a fast real-time kernel to handle the simulation tasks, and lots of I/O bandwidth. In addition, a custom interface for cockpit controls and instrumentation is required to handle all aspects of the simulation process.

Consider simulating airflow over a wing. A processor must not only compute the forces on the wing which result from the fluid motion of air over its surface, but also determine and apply (in real-time) the proper amount of feedback pressure to the pilot's control stick.

In addition to the simulation requirements of the system, a flight data collection and storage mechanism, and possibly an instructor interface, will be needed to handle on line customization of the flight variables and post-flight critiques.

These tasks demand multiprocessing capabilities in an environment which allows constant interaction among the seperate processors completing the tasks. MULTIBUS II is an excellent architecture for such system requirements due to its 32 megabyte per second bus transfer rate, high performance multiprocessor support, and message passing bus communication design. In addition, the MULTIBUS II Systems Architecture firmware architecture and operating systems supported ease the system level integration requirements for such a system.

A rough system configuration is shown in Figure 2. A single iSBC 386/120 CPU running UNIX works in conjunction with an iSBC 386/258 SCSI controller to provide data storage and retrieval as well as an instructor interface. Another iSBC® 386/120 controls the engine simulation and weather variation functions, while a third handles radar and communications simulation.

The MIX baseboard in this example controls the airflow simulation as well as the instrumentation output and control stick feedback I/O. The I/O interface to the actual simulator instruments and control stick is accomplished via a slave MIX module which utilizes a high speed digital I/O link. This module is the only hardware which must be designed by the systems integrator.

In addition to the high performance hardware functions this solution provides, it also offers support for current Intel system firmware and operating systems.

From a firmware point of view, all boards discussed ship with Intel MSA compatible firmware. This allows ease of integration into a system environment in terms of board and system level diagnostics, boot mechanisms, and board-to-board communication support.

Ease of integrating the custom MIX module into this firmware architecture is also provided. Functions in the MIX baseboard firmware allowthe I/O module firmware to interact with the

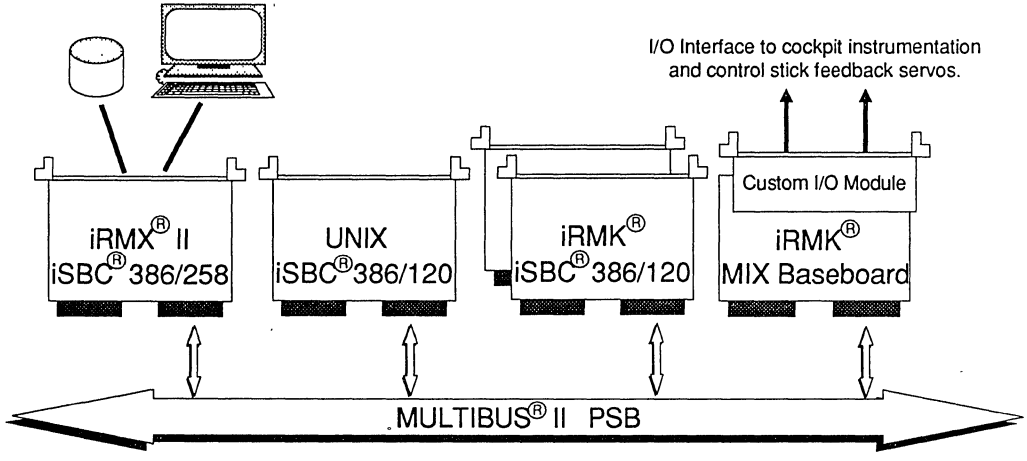


Figure 2. Flight Simulator Example

baseboard firmware for diagnostics, boot, and operating requirements without the need for modifying the baseboard EPROM contents. This enables any standard or special I/O modules to participate in BIST testing and boot procedures.

The demand for high performance software capabilities can be easily realized with the use of the iRMK® real-time kernel. This 32-bit kernel includes the transport and message passing support necessary for the intra-system communication used in completing the system level task. In this example, the MIX baseboard and each iSBC 386/120 performing simulation tasks runs the iRMK kernel.

UNIX System V/3.2 is run on the host iSBC 386/120 to provide the operator interface and flight record databasing functions of the system. These functions are typical of the capabilities of standard UNIX and exist as software packages supported by the UNIX operating system.

With the host running UNIX System V/3.2, the 386/120 boards running the iRMK kernel can take advantage of the capabilities of Intel's System V/iRMK. This version of iRMK is designed to operate in a UNIX hosted environment and provides a UNIX hosted development environment as well as a tested transport package for UNIX / iRMK communications. These capabilities greatly simplify the data collection and storage requirements of the system.

The remaining pieces of the system which must be developed by the system integrator are the custom MIX module, the iRMK tasks to perform the simulation, and the instructor interface routines for the UNIX subsystem. This document will discuss the MIX module portion of the design.

3. MODULE DEFINITION

3.1 Module Purpose

The MIX module to be designed must be capable of supplying many external digital I/O connections which can be quickly accessed and latched for reliable digital I/O performance. The module will also contain a serial interface to the MIX baseboard for debug purposes. These functional goals of the module can be realized in a simple 8-bit slave I/O MIX module design. To improve the aggregate I/O throughput of the design, the module architecture will be extended to support 32-bit accesses to the digital I/O channels.

This module design does not demonstrate the full capabilities of the MIX architecture as the module does not have MIX master capabilities. It does however, function as a useful example to demonstrate the considerations and issues involved in the design of a simple MIX module.

3.2 I/O Component Definition

In this module design example, a simple slave I/O device is used for digital I/O control. The 8255A Programmable Peripheral Interface component provides all of the I/O capabilities required of the external interface. The device provides three 8-bit parallel I/O ports programmable in three modes. The speed requirements of the digital I/O interface are met by utilizing the 82C55A CHMOS version of the chip. This version offers much better timing specifications for command widths and command recovery times as compared to the NMOS 8255A. In addition, the CHMOS device consumes much less power than its NMOS relative.

For the serial interface, the module will utilize the 82510 Asynchronous Serial Controller. This chip is chosen over the more common 82530 due to its proven high performance and its low power, CHMOS characteristics. In addition, this chip is being used on other Intel MIX modules such as the MIX 450 Terminal Concentrator and the MIX 560 Ethernet Controller.

3.3 Module Functional Blocks

The I/O chips chosen above have signal sets based on the 8086 architecture. Because the MIX interface supports an i386™ signal set, the module design must supply a signal conversion function for these components. Furthermore, the module must supply data transceivers, an I/O address decode scheme, and an interconnect EEPROM interface.

By separating the design into functional partitions, the task of transforming MIX signals into signals compatible with the I/O devices becomes a straight forward process.

The features described above are easily separated into five main functional blocks as shown in Figure 3. They are the MIX interface, the data transceivers, the control logic block, the I/O block, and the configuration data block.

The MIX interface block simply consists of the MIX connector and the terminating resistors required for the data bus and control signals. The data buffer block contains the data drivers which provide bidirectional data transfers and support the MIX interface TTL drive requirements. These devices will be set up in a byte swap configuration which allows BYTE, WORD, or DWORD accesses to the I/O devices.

The control logic block is made up of PALs which perform the I/O address decode and convert the MIX bus i386 family signals into read and write commands useable by the parallel and serial components. It also handles the READY logic for the module and generates the proper gate and direction signals for the data drivers. This block is by far the most complex portion of the module. The I/O block contains the parallel and serial chips as well as the drivers and connectors required to interface with external devices. The final block is the configuration data block. This block contains a serial EEPROM device which provides dynamic configurability of the baseboard's interconnect space during power up.

4. DESIGN TOOLS USED

Several engineering tools are used in this project to shorten the design cycle, simplify logic design requirements, and provide a means of simulation before the module is

prototyped. Because this is a relatively simple design, engineered by a single person, it is desirable to find engineering tools which can run on a common desktop PC. Many PC/AT compatible design tools exist which offer the capabilities needed for this type of board design.

The tools purchased from third parties for this project include a schematic capture package and a PAL design and simulation utility.

4.1 OrCAD Schematic Capture Package

OrCAD Systems Corporation provides a CAE package which includes a schematic capture design tool and a PCB layout utility. The schematic capture tool is used in this module design.

The OrCAD schematic package offers hierarchical or flat file schematic design, common component libraries, annotation, back annotation, cleanup, cross referencing, error checking, netlist, and parts list utilities. The completeness of this package allows the entire module design to be completed on a simple PC/AT computer.

4.2 PALASM 2 PAL Specification Utility

The control portion of this MIX module must perform address decode functions as well as generate chip selects, provide transceiver gate signals, and handle the READY circuitry. While these functions can be designed using discrete logic, they are more appropriately and efficiently handled using a PAL or EPLD device.

A PAL programming utility is required to input PAL logic equations and convert those equations into fusemaps which a PAL programmer understands. Such a utility is provided by Monolithic Memories, Inc. This utility is called PALASM 2. It is a follow on to MMI's PALASM software which assembles PAL design specification code, generates PAL device fuse patterns in JEDEC format, checks for syntax and assembly errors, and provides a means for simulating the PALs operation based upon given input states.

5. MODULE LOGIC DESIGN

To complete the logic design for this MIX module, the hierarchical capabilities of the schematic capture package are used in conjunction with the preliminary block diagram shown previously in Figure 3. This procedure allows the design to be broken down into simpler functional design tasks. As shown on sheet 1 of the module schematic (Appendix A), the five functional blocks of Figure 3 have been expanded to eight. A capacitor block has been added to provide the decoupling needed in any TTL design. The MIX block has been broken into the MIX interface block and termination block, and the I/O block has been divided into the Programmable Peripheral Interface (PPI) block and the Serial Interface block.

To start the design with the MIX interface block would not be productive as we have not yet defined the MIX signals which must be supported. Because this design is a MIX slave, the module need only support the MIX signals required to complete its I/O function. To understand which signals are required for this module, the module is designed from the I/O devices back to the MIX interface.

As mentioned earlier, the I/O block defined in Chapter 3 has been divided into two blocks. This is done because the module utilizes two distinctly different I/O devices. We therefore

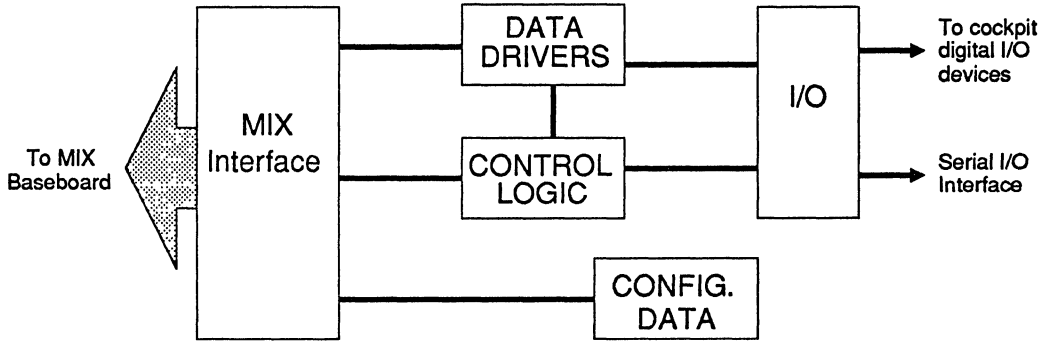


Figure 3. MIX Module Block Diagram

divide the I/O block into a parallel I/O block and a serial I/O block.

5.1 Parallel I/O Block

The 82C55A PPI will serve as the central I/O device for this block. This 8-bit device provides 24 programmable I/O pins which may be individually programmed in 2 groups of 12. It allows direct bit set/reset capabilities and is fully TTL compatible.

As shown in Figure 4, the 82C55A requires 8086 compatible signals for read and write control. The conversion from 386 to 8086 compatible signals as well as the chip select generation will be handled in the Control Block of the design. Aside from RESET, the two remaining chip control signals are A0 and A1. These will be obtained from the MIX interface signals MXA2* and MXA3* respectively. This addressing scheme will have the effect of separating the internal chip registers by 4 bytes.

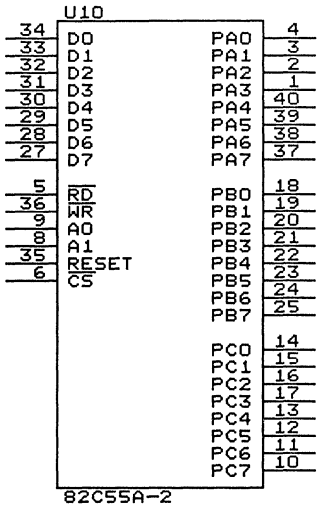


Figure 4. 82C55A PPI

As stated in section 3.1, the MIX module is to provide a 32-bit data path to the parallel I/O interface. To accomplish this, four 82C55A components operate in parallel. Except for the select lines, the control signals for each chip may be wire ORed together. This is shown on sheets 8 and 9 of the module schematic (Appendix A). To facilitate BYTE, WORD, and DWORD accesses to the digital I/O interface, the data drivers in the Data Buffer Block are designed in a byte swap configuration. Finally, the Control Block decodes the type of access being made and generates the proper chip select(s) to the peripheral chips.

Because the PPI is capable of only 2.5 mA of DC drive, the parallel I/O block also incorporates line drivers between the PPI and the target I/O devices. In this design, only ports A and C of the PPI are used. This gives 64-bits of digital I/O and uses up to 16 line driver components.

As shown on page 9 of the module schematic, four 74F245 Octal Transceivers drive the port A signals. Two signals from port B enable and configure the drive direction for the transceivers. These fast devices provide 64 mA of low level output current on the "B" side of the driver. This level is sufficient to drive most digital I/O needs.

Eight 14 pin DIP sockets are used for installation of the port C signal drivers. This allows the user to configure port C with a variety of Quad Two-Input drivers or terminators. This design also uses a 64 pin DIN connector to provide external connection to the 32 port A signals and their associated ground lines. Eight 74ALS00 devices drive the port C outputs to illuminate four banks of eight LEDs. This is shown on sheet 11 of the module schematic.

Although Port C of the parallel I/O block in this design is set up for display (LEDs), the block can be easily modified to provide 64 or 96 (using port B) lines of external I/O.

5.2 Serial I/O Block

The Serial Block for this module consists of the 82510 Asynchronous Serial Controller, a crystal oscillator, and a RS-232 line driver. The 82510 is an eight bit device and is chosen for this design because of its performance advantages over the more common 82530 chip. These advantages come from on chip FIFOs and higher clock frequency capabilities. This chip also has a power down mode for very low power consumption when not in use.

The RS-232 electrical interface for the module is designed as a three wire circuit. As shown on sheet 12 of the schematic, the J1 D-connector carries only TXD, RXD, and ground. Because the port is designed to be used for debug purposes, the remaining six modem signals are considered unnecessary.

The 82510 and required signals are shown in Figure 5. The control signal set for the 82510 differs little from that of the 82C55A, so design integration is simple. The chip requires an additional address signal, A2, which is derived from the MXA4* signal. As in the case of the 82C55A, the address line connections cause the internal registers of the component to be DWORD aligned.

One interrupt line also comes from the 82510 to signal FIFO conditions, errors, etc. This interrupt line is routed through an inverter and a series terminating resister to the LCLINT* pin of the MIX interface since this is the only interrupt on the module. The MIX interface termination requirements are found in the *MIX Module Design Specification* (Intel p/n 500729-001).

The chip also requires a clock source for baud rate generation. The 82510 allows two modes of operation for the clock source. The first mode uses an external oscillator to internally generate its system clock while the second uses an externally

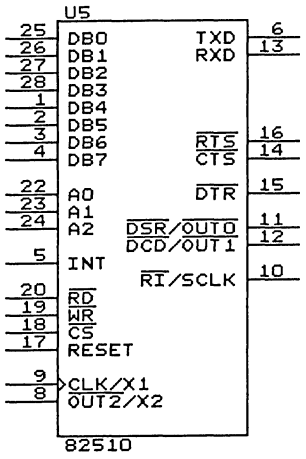


Figure 5. 82510 ASC

generated clock. This module design opts for the former for several reasons. First, an internally generated clock has lower cost and is electrically "cleaner" than using an external clock. Furthermore, the low power sleep mode of the chip is easily utilized with an internally generated clock. This mode requires the RTS pin to be driven low during a reset. This is provided by the CLKMOD signal shown on sheet 12 of the schematic. In this mode, we will use an external 18.432 MHz crystal which results in a 9.216 MHz internal clock (/2). This crystal frequency provides a 0% error baud rate generation for standard baud rate values.

The serial driver used is an MC145406 component. This device is RS-232 compatible and supports both transmit and receive signal voltage conversions on a single chip. Because only the Transmit Data and Receive Data signals are driven, a single MC145406 is sufficient for this application.

Now that the I/O blocks have been specified and the data line requirements known, we may proceed to design the data driver block of the MIX module.

5.3 Data Driver Block

As stated in the design requirements, the module must support 32-bit accesses to the four PPI devices as well as eight bit accesses to the serial device. Restricting parallel port accesses to only 32-bits would limit the baseboard software to 32-bit code or limit the number of PPI devices accessible with BYTE or WORD transfers. To solve this problem, each PPI will be accessible as a separate I/O device or all PPIs will be accessible as a single 32-bit I/O device.

This structure requires a byte swap mechanism in the data driver design. The design allows each PPI and the 82510 ASC to be accessed on the data signals MXD0 - MXD7. Furthermore, to obtain a "32-bit" parallel port, one PPI (PPI1) is accessible on MXD8 - MXD15, one (PPI2) on MXD16 - MXD23, and one (PPI3) on MXD24 - MXD31. This configuration requires 7 data transceivers as shown on sheet 5 of the schematic.

When accessing the four PPIs as a single 32-bit device, data transceivers U7, U9, U19, and U29 (schematic sheet 5) must be enabled at the same time. When accessing the PPIs as separate 8-bit devices, the U7, U8, U18, or U28 transceiver is individually enabled to provide a connection between the PPI being accessed and data signals MXD0 - MXD7.

The "G" pin of the transceiver stands for "gate" and is used to enable the driver. The data driver setup shown on sheet 5 requires 5 gate signals to enable the proper transceivers for a PPI transfer cycle. The signal GAT32* is activated during a 32-bit parallel port access. GATB0* transfers data between the U10 PPI and the low data byte on the MIX interface, GATB1* does the same for the U20 PPI, and so on. These signals, along with the data direction signal, are generated by the Control Logic Block of the module design.

5.4 MIX Interface Block

As stated previously, a MIX slave module need only support those MIX interface signals which allow it to complete its I/O function and remain MIX compatible. Now that the I/O device and data drivers blocks have been defined, we can determine the MIX interface signals needed to support this simple slave design.

The 130 signals on the MIX interface can be broken up and viewed as 11 functional groups. This eases the task of selecting the necessary signals. Aside from the Spares group, the MIX slave module signal groups are listed in Table 1. Notice that a slave module does not utilize any of the signals of the Arbitration Group as it will never need control of the interface.

The address group consists of 30 address lines, four byte enables, and three select lines. Because the module is a simple slave without memory mapped devices, only I/O address space need be considered. A MIX module is allotted 1K bytes of I/O address space in either the second, third, or fourth Kbyte in the baseboard's I/O space, depending on the stack position of the MIX module. Luckily, the baseboard's decode logic determines which module is selected and signals that module with a MXSELx* signal, seen as LCLSEL* on the module. This way, the baseboard assigns a module's base I/O address at 400H, 800H, or 0C00H and a module is designed based only on address offsets within that 1K bytes of space. In this scheme however, the module must use the LCLSEL* signal as a qualifier in the address decode scheme to assure it is the module being accessed. To support 1K bytes of address space, we need only support MXA2* through MXA9* and the byte enable signals. The module must also input its LCLSEL* signal to aid in the decode scheme. Furthermore, although the module does not act on the remaining address and MXSELx* signals, it must route the signals from the primary to secondary

Function	Signal Name	Signal Type	Used
Address	MXA[31:28]*	Pass-through	N
	MXA[27:7]*	Bussed	N
	MXA[6:2]*	Bussed	Y
	MXBE[3:0]*	Bussed	Y
	LCLSEL*	Owned	Y
	MXSEL[2:1]*	Pass-through	N
Data	MXD[31:0]	Bussed	Y
Status	MXMIO	Bussed	Y
	MXWR	Bussed	Y
	MXDC	Bussed	Y
Control	MXCYC*	Bussed	Y
	MXCMD*	Bussed	Y
	MXBS16*	Bussed	N
	MXWAIT*	Bussed	Y
Configuration	LCLID*	Owned	Y
	MXID[2:1]*	Pass-through	N
	SCLK	Bussed	Y
	SDIN	Bussed	Y
	SDOUT	Bussed	Y
	MXRST*	Bussed	Y
Interrupt	LCLINT*	Owned	Y
	MXINT[2:1]*	Pass-through	N
DMA	LCLDRQ*	Owned	N
	MXDRQ[2:1]*	Pass-through	N
	LCLDACK*	Owned	N
	MXDACK[2:1]*	Pass-through	N
Option	LCLOP*	Owned	N
	MXOP[2:1]*	Pass-through	N
Spares	Reserved	Bussed	N
Power	+5 VDC	Bussed	Y
	GND	Bussed	Y
	+12 VDC	Bussed	Y
	-12 VDC	Bussed	Y

Table 1. MIX Slave Module Signal Set

sides of the module. This is true for all signals of the MIX interface.

Because this module has been defined to support 32-bit accesses to the parallel I/O devices, all 32 data lines must be supported. All signals in the status group must also be supported to determine the type of access being made to the module.

In the Control group, the signals MXCYC* and MXCMD* are used to determine the state of a MIX transfer cycle. These two signals must be supported by even the simplest of modules. These signals are very useful in generating I/O commands to devices as well as chip select signals. The other signals in the Control group, MXBS16* and MXWAIT*, need only be used by modules which cannot support all 32-bit byte enable combinations or cannot accommodate full speed MIX transfers.

The MXBS16* is used to inform the baseboard that the 32-bit access being made is not supported by the module. This is the same as the BS16* signal for the i386 processor. This MIX module will support 32-bit access to the four PPI device group but not the individual PPIs or 82510 ASC. Therefore, the MXBS16* signal is generated for any access made on the module except an access to the PPI0 address since this is the address we use to enable all four PPIs during a 32-bit access.

To determine the need for the MXWAIT* signal, we must investigate the requirements of the I/O devices being used on the module and understand the result of using the MXWAIT* signal. The MXWAIT* signal can only be used on the MIX interface to extend the cycle time of the MIX transfer being made (i.e., extend the length of the MXCMD* and MXCYC* active time). This signal is very similar to the iSBX MWAIT* signal. Because of the effect of this signal, we need only consider the command duration time requirements of the device. Timing needs for parameters such as address or data setup, hold, and command recovery timings are not affected by simply activating the MXWAIT* signal. Meeting these requirements will necessitate additional control circuitry between the MIX interface and the I/O device.

Timing requirements for TTL components, of course, vary with the device. Typically, as is the case in this module design example, it is the peripheral devices which have the longest timing requirements. A full speed MIX transfer cycle is specified as 175 ns which corresponds to a 73 ns MXCMD* pulse width. If we use the MXCMD* signal to generate our RD* and WT* commands, we would have a minimum command pulse of 73 ns. Although the current MIX baseboard does not support a full speed MIX transfer, it is desirable to design the MIX module to the capabilities of the MIX specification (not the baseboard spec) so that the module will be compatible with future MIX baseboards. For this reason, the decision to support MXWAIT* is based upon a 73 ns command pulse time.

In the case of the PPI and the 82510 ASC, the parameters we are concerned with are the I/O Read and Write command hold times. The 82510 requires a minimum RD* active width of 2 T_{CY} + 65 ns which is 282 ns. Already we see we cannot support a full speed MIX transfer and will therefore need to use the MXWAIT* signal.

Support of all signals in the configuration group is a requirement of the MIX specification. These signals allow the baseboard and other PSB agents to determine the types of MIX modules present in any given MIX stack.

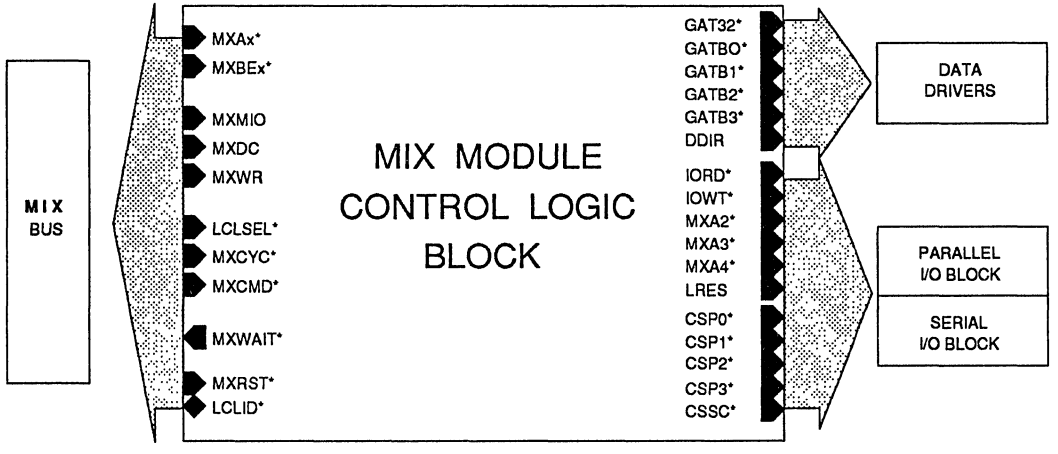


Figure 6. Mix Module Control Block

The Interrupt group of the MIX interface consists only of a single interrupt line per module. This interrupt line supports cascading which facilitates the use of a slave interrupt controller on a MIX module. Because this module has only one interrupt signal source (the serial controller), the cascading option is not needed. The remaining signal groups include the DMA and Option lines. This module does not utilize baseboard DMA resources nor does it require an additional interrupt signal, so these signals need not be supported. As spares go, MIX modules are required to bus the signals for future use but may not drive the signals.

It is a requirement of the MIX specification to connect to all power pins of any voltage which is used on the module. Because this module supports RS-232 communications, all available voltage levels are utilized. The module must therefore connect to all +5V, +12V, -12V, and GND pins.

As shown in Table 1, the final summary of supported signals for this module is designated with a "Y" in the "Used" column. Now that the MIX interface signal support and I/O blocks are defined, the next task is to design the block which converts the MIX compatible signals to commands understood by the I/O devices being used. This is accomplished in the Control Logic Block of the Module.

5.5 Module Control Logic Block

As shown in Figure 6, the MIX module control logic block resides between the MIX interface and the module's I/O and data driver devices. Because this block is similar to that which may be utilized on other MIX slave modules, detailed attention will be given to its design.

The control block has three basic functions; translating MIX signals into commands and timings understood by the I/O devices, performing address decode operations, and handling the module reset functions. The reset function is the simplest of the two, so it will be designed first. The reset function on the MIX interface is driven by the baseboard's microcontroller and, as stated in the module design specification, is guaranteed to be a minimum of 10 us in duration. The requirements of all devices used in this module design are at least an order of magnitude less than this. For this reason, we need only invert the MXRST* signal to supply the active high RESET required by the peripheral devices.

Another requirement of MIX modules is the support of the MXIDx* / MXEECSx* signal. During reset, the module must drive its LCLID* signal to inform the baseboard it is present. We can generate this signal by simply inverting the inverted MXRST* signal during a reset (this is accomplished with a tri-state inverter which drives the signal only during reset). Note that we could not simply connect the MXRST* to the LCLID* signal, or simply invert the MXRST* signal at all times since LCLID* takes on the function of LCLEECS* after reset. If we had simply tied MXRST* to LCLID*, an access to the serial EEPROM would result in a MXRST* being active on the MIX bus. If we did not tri-state the LCLID* driver, the baseboard MXEECSx* driver and our LCLID* driver would attempt to drive the same signal to opposite polarities. Also note that this design meets all reset timing requirements as listed in the *MIX Module Design Specification.*

The address decode functions and command conversion logic are the next portions of the control block which must be designed. Because of the relative isolation between these two functions, they can be partitioned into two designs, however, proper timing between the two functions must be ensured.

Typical operation for a peripheral chip is to set up address and chip select signals to the device, set up the data (for a write), and then pulse the read or write command. To do this, we can construct two logic sub-blocks we will call the I/O control and I/O decode blocks.

The I/O control block will be responsible for generating the RD* and WT* pulses to the chips, and controlling the MXWAIT* and MXBS16* functions.

The I/O decode block therefore has the responsibility of performing all the address decode operations and generation of the chip selects and data driver gate signals. These functions are derived from the states of the address and byte enable signals during a valid transfer cycle. The functions in both blocks are best handled by PAL or PLD logic design due to the device's logic flexibility and the number of input signals required of the functions.

5.5.1 I/O Decode PAL

The main function of this block is the generation of the peripheral device chip select signals. Each addressable device on the module is assigned a unique I/O offset address by this PAL. A device's chip select is only activated when the MIX baseboard accesses the chip's assigned I/O address via the MXA[6:2]* and LCLSEL* signals.

In addition to a valid address, it is wise to ensure the proper byte enable combination exists to provide valid data to the chip being selected. Therefore, our chip selects will be qualified by three things; one, a valid MIX transfer cycle is present; two, the proper I/O address is selected; and three, a valid byte enable combination exists. Because each of the peripheral devices require MXA3* and MXA2* for internal register access, we need not use these in qualifying accesses to a specific chip. This will force our address decode to place the peripheral devices on DWORD boundaries.

The addresses for each chip may be arbitrarily chosen between I/O offsets 0 and 3F0H. Note that as specified in the *MIX Module Design Specification*, I/O address offsets 3F0H through 3FEH are reserved. This module utilizes address lines MXA[6:2]* for its decode function. Because this module does not decode address lines nine through seven, the I/O chips are mirrored in the address map every 128 bytes. The first occurance of the chips' I/O addresses are shown in Table 2.

I/O OFFSET	DEVICE
0xH	Byte PPI (PPI0)
	Word PPI (PPIs 0,1)
	Dword PPI (PPI 0-3)
1xH	Byte PPI (PPI1)
2xH	Byte PPI (PPI2)
3xH	Byte PPI (PPI3)
4xH - 5xH	Byte 82510 ASC

Table 2. I/O Addresses

A secondary function of this block is the control of the data transceiver gate signals.

There are seven data transceivers in the Data Driver Block of the module which have enable pins ("G") which must be driven during peripheral accesses.

Of the seven data drivers three are only used for 32-bit transfers. These may be hard wired together and enabled as a group during a 32-bit access (This signal will be called GAT32*). The four remaining gate signals are used during byte wide accesses to the PPIs (We will call these GATB[3:0]* based on which byte of the module data bus the driver controls).

The data driver gate signal /GATB0 enables the data driver which transfers data from the low byte of the MIX data bus to the low byte of the module's data bus. PPI0 and the 82510 reside on this bus. This driver should be enabled on four access occasions. Three involve byte, word, or dword accesses to PPI0 and the fourth is for a byte access to the serial chip. The /GAT32 signal enables three data drivers which transfer data from the upper three bytes of the MIX data bus to the corresponding upper three bytes of the module's data bus. This signal should be activated only during word or dword accesses to PPI0. By providing simultaneous chip selects to the other PPIs, we can make the baseboard believe it is making an access to a single word or dword device when in actuality, two or four PPIs are being utilized.

The remaining gate signals (GATB[3:1]*) allow the upper three PPI devices to be accessed via the low byte of the MIX data bus. These three signals drive data between the low byte of the MIX data bus and one of the upper MIX module data bus bytes (1, 2 or 3). The signals are only activated during byte accesses to PPIs 1, 2 or 3.

Because the data timing for our devices must conform to the MIX interface timing requirements as well as the timing requirements of the peripheral chips, we must activate these drivers in a manner which meets the minimum data turn-on and minimum data hold time requirements. This can be accomplished by dividing the enable function between read and write accesses to the module. During a read operation, the module must tri-state its data drivers within 40 ns of the MXCMD* signal being deactivated. During write accesses, the 82C55A requires data be held at least 30 ns after the RD* or WT* signal is removed. Both needs are met by varying the gate parameters based on whether a read or write access is occurring.

From these definitions we find the total number of signals used by the decode logic is 12 inputs and 10 outputs. A common 22V10 PAL facilitates this logic design perfectly.

The PAL signal definitions are shown in Table 3. Note that the chip select generation scheme for the PPI devices will also allow us to make 16 bit accesses to the two PPIs which occupy the low word of the data bus (U10 and U20) without modification to the data driver scheme.

The actual PAL source code for this block is listed in Appendix B under the title "IODCD PAL".

SIGNAL	DESCRIPTION
/A6 - /A4	Inputs from MXA9* - MXA4*
/BE3 - /BE0	Inputs from MXBE3* - MXBE0*
/LCLSEL	MIX i/f input
/MXCMD	MIX i/f input
/MXCYC	MIX i/f input
MXWR	MIX i/f input
/GATB0	Byte 0 data driver gate signal
/GATB1	Byte 1 data driver gate signal
/GATB2	Byte 2 data driver gate signal
/GATB3	Byte 3 data driver gate signal
/GAT32	Bytes 1,2 & 3 drivers gate
/CSP3	PPI3 chip select
/CSP2	PPI2 chip select
/CSP1	PPI1 chip select
/CSP0	PPI0 chip select
/CSSC	82510 chip select

Table 3. I/O Decode PAL Signals

5.5.2 I/O Control PAL

As discussed earlier, the I/O control portion of the design is responsible for translating MIX compatible signals into commands recognized by the peripheral devices. It must also provide the MXWAIT* support for proper command timing and MXBS16* support for proper data accesses to the peripherals.

Since the devices supported on this module will act only on I/O read and I/O write commands, this PAL must translate the MIX Status group signal states into either IORD* or IOWT* commands to the 82510 and PPI devices. This is accomplished by decoding the states of MXMIO, MXDC, MXWR, LCLSEL*, MXCYC*, and MXCMD*. As an added check, we will qualify our decodes with the state of the MXRST* signal to ensure chip selects are not generated from the arbitrary signal states present during reset.

Because of the pulse width and command recovery timing requirements of the 82510 and the 82C55A we must support the MXWAIT* signal of the MIX interface. This signal will directly fulfill our pulse width needs and indirectly provide the command recovery time needed by the peripheral devices. As stated in the *MIX Module Design Specification*, the MXWAIT* signal must be asserted within 40 ns of the MXCYC* signal being activated. This requirement will require asynchronous and synchronous state machines to be linked in the PAL logic.

The I/O Control Block will also drive the MXBS16* signal on the MIX interface. A MIX module must drive this signal when it cannot support 32-bit accesses to its devices. This module design will activate this signal on all accesses except for those to the PPI0 address since this address is the one used for our 32-bit PPI access. To distinguish a PPI0 access, the control PAL will input several of the address lines.

An additional function not yet mentioned is the direction signal generation required by the data transceivers. Because these transceivers are only enabled during valid command cycles, we simply drive the direction pins with the MXWR signal. Note that the stub length requirements of the MIX interface and the drive requirements of the transceivers necessitate the use of a driver to produce the direction signal. This driver is a simple hex noninverter and may be implemented external to the PAL design.

As specified, our PAL design will require twelve inputs (including a clock signal for the MXWAIT* machine) and four outputs. From these requirements a 22V10 PAL device may again be used. This device allows twelve input and ten output pins. Although we have only defined four outputs, additional output pins will be consumed by the WAIT state machine design. A particularly nice feature of the 22V10 device is the ability to configure the outputs as either registered or simple combinatorial output pins. This is useful since we have both asynchronous and synchronous needs for this PAL design.

The first consideration in our PAL design is the tri-state needs of the MIX bus signals. The MIX specification dictates that the MXWAIT* and MXBS16* signals be tri-stated while a MIX cycle is not occuring. Furthermore, these signals must be in their logical false state before they are tri-stated. To facilitate this, we define a signal called IOCYC.* This signal is a combination of inputs which represent a valid I/O cycle (read or write) is occuring. The actual input signals can then be used to activate and deactivate the MXWAIT* and MXBS16* signals, while the IOCYC output is used to tri-state them. This provides 15 ns (the prop delay of the PAL) between the signals being deactivated and being tri-stated.

The IORD* and IOWT* signals can build upon the IOCYC* signal by using IOCYC* as a qualifier. Once IOCYC* is valid, the MXWR signal is used to distinguish between read and write accesses, and the MXCMD* signal is used to allow address and chip select setup times to the peripheral devices. The read and write commands will require one additional qualifier to provide the command recovery timing needs for the PPI chips. This addition will be discussed during the WAIT state machine design.

The MXBS16* function is a fairly simple signal to generate for this design. The module will only allow 32-bit accesses to the address specified for PPI0. This address, along with the Byte Enable signals, is used in the decode logic to enable all PPIs simultaneously when a 32-bit access is made to PPI0. Therefore, this PAL will not drive the MXBS16* signal during an access to PPI0. This is assured by verifying an access to PPI0 is not occurring.

Because of the propagation delays involved in PAL decode, we are not able to use the chip selects generated by the I/O Decode PAL for the generation of MXBS16*. The 15 ns delay of the chip select would result in a MXBS16* glitch during a PPI0 access. We can however, use the address lines MXA[6:4]* as signal qualifiers. These signals are specified to be glitch free during valid transfer cycles and will indicate which device is selected when MXCYC* is activated. We therefore use the address lines with the MXCYC* line in generating MXBS16*. Note that if the PAL offered more inputs, the signals MXA[9:7]* could also be used which would cover the full 1 kilobyte address range of the module.

The last requirement placed on the MXBS16* signal is that it be tri-stated within 46 ns of MXCYC* going inactive. This is so because the MXBS16* signal is bussed on the MIX interface. Since our 22V10 PAL device allows us to individually assign tri-state conditions to outputs, we can use IOCYC* to tri-state the MXBS16* output. The allows us to force MXBS16* inactive (high) with the removal of the MXCYC* signal and tri-state the line 15 ns (the propagation delay of the device) later when IOCYC* is deactivated.

The final function provided by this PAL is the MXWAIT* support. As discussed previously, the 82510 device requires a command pulse width of 282 ns for a read operation and 232 ns for a write. The 82C55A PPI device requires pulse widths of only 150 ns for a RD and 100 ns for a WR. Although we could include logic to distinguish PPI accesses from 82510 accesses, a generic MXWAIT* which supports the worst case time is much simpler to design. Therefore, a state machine must be designed which will produce a pulse greater than or equal to our maximum pulse width constraint of 282 ns.

Our peripheral devices also specify command recovery requirements which are not met by a full speed MIX cycle. The command recovery is the amount of inactive time between

RD* or WT* pulses to single chip. For the 82510 this value is 120 ns. The PPI requires 200 ns. As stated in the *MIX Module Design Specification*, all modules MUST provide hardware support for command recovery needs. This support can be rolled into the MXWAIT* state machine by designing in a number of inactive states in which the RD* or WT* signals cannot be activated. To assure these times are met and to assure we meet the MXWAIT* activation time requirements, we will utilize an asynchronous state machine to generate the actual MXWAIT* signal, coupled with a synchronous state machine which "turns off" the asynchronous machine. We will also use the output of our synchronous machine as a qualifier in the IORD* and IOWT* signal generation. This will give us the command recovery time needed for proper operation of the 82510 and 82C55A chips.

The asynchronous machine will define the conditions which must be present for the MXWAIT* signal to be asserted. In the synchronous state machine design, we must define three things; the clock frequency for the PAL device, the number of states in which SWAIT (synchronous WAIT) is active, and the number of states SWAIT is inactive. By qualifying IORD* and IOWT* with an active SWAIT, the added inactive SWAIT states provide our recovery time.

For the asynchronous machine, we define an idle state for a reset state or otherwise idle module (no transfer occurring) condition. Once the module is accessed and MXCYC* is asserted, we must provide MXWAIT* within 40 ns. A second state is therefore defined with MXWAIT* active. This state is entered when IOCYC* is active. With the PAL device used, we know MXWAIT* will be activated a maximum of 30 ns after MXCYC* is activated (15 ns from MXCYC* to IOCYC* and 15 ns from IOCYC* to MXWAIT* coming out of tri-state). This second state is held until the synchronous machine sets SWAIT active. Once this occurs, the machine moves into a third state which also has MXWAIT* active. This state is held until the synchronous machine indicates the delay time has expired (falling edge of SWAIT). The asynchronous machine then deactivates MXWAIT* and enters its final state until the cycle has completed. This state exists so that the loop is not repeated during the same MIX transfer cycle. Once IOCYC* goes inactive, the idle state is reentered.

To design the synchronous machine, we need to know the requirements for pulse widths and recovery times. As state previously, our longest pulse width requirement comes from an 82510 read cycle (282 ns) and our worst case recovery time requirement comes from back to back 82C55A accesses (200 ns). With these in mind, the equation $n / f = T_{WAIT}$ is used to determine the number of active and inactive WAIT cycles, "n", used in our synchronous state machine. The value "n" will vary based on the clock frequency, "f", of the PAL, and the required cycle time, "T_{WAIT}".

For the active cycles, T_{WAIT} represents the maximum command pulse width. For the inactive cycles, T_{WAIT} is the maximum command recovery time. Since both of these values are divisible by 40 ns we will use an "f" of 25 MHz (1/25 MHz = 40 ns). For this frequency we need 7 active states and 5 inactive states. Remember these active and inactive states represent states where MXCMD* is active or inactive respectively. We must therefore add an additional state for both the active and inactive transitions which waits for MXCMD* to transition to its desired state.

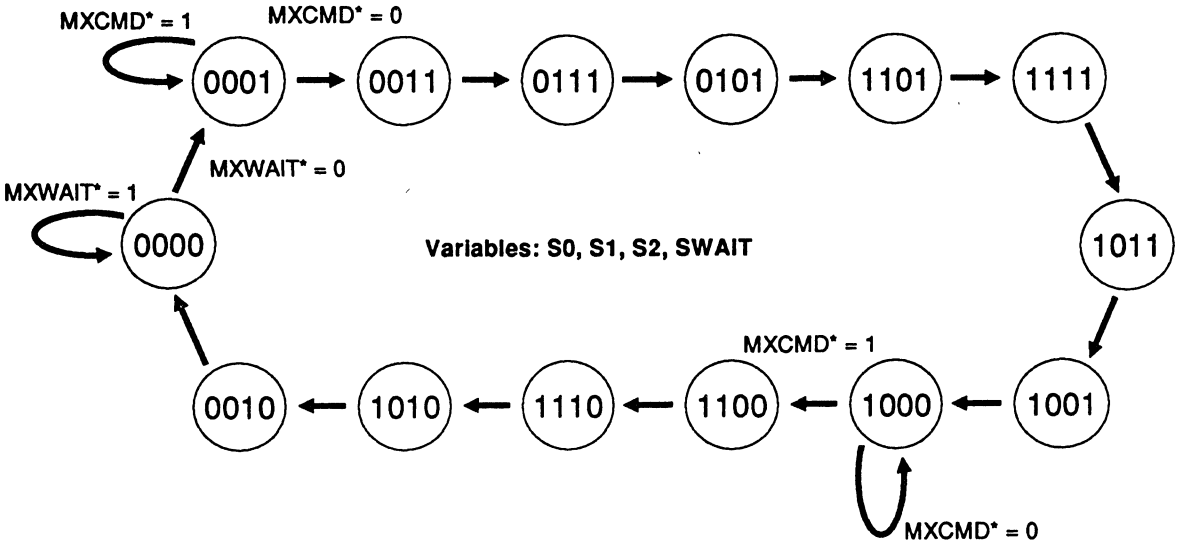
Two state diagrams are shown in Figure 7. The top represents the synchronous machine while the bottom represents the asynchronous machine. For the synchronous machine, four state variables are defined: S0, S1, S2, and SWAIT. The "0000" state is the machine's idle state. Once a cycle is initiated on the module and the asynchronous machine asserts MXWAIT*, the synchronous machine moves through its states as shown in the diagram. Notice that the 0000, 0001, and 1000 states require MXWAIT* or MXCMD* to transition before the next state is entered. The MXWAIT* requirement ensures the cycle is only entered once per valid module transfer cycle. This is accomplished without requiring the synchronous machine to be in a specific state when a MIX cycle starts. Note that a new transfer cycle could start anywhere between the 1100 and 0000 states of the synchronous machine. The MXCMD* loops exist to ensure the command widths and recovery timing needs are met. Note that SWAIT is asserted with MXWAIT* active but the clocked cycle is not executed until MXCMD* is activated. This is desirable since MXCMD* is the signal we need to extend by our T_{WAIT} value. Similarly, once SWAIT is deactivated, the machine will wait for MXCMD* to be deactivated before continuing with its cycle. By only allowing the IORD* and IOWT* commands to be activated while SWAIT is active, we satisfy our recovery time requirements by assuring SWAIT is inactive for our second (recovery time) T_{WAIT} value after MXCMD* is released.

SWAIT is the synchronous variable which is linked with the asynchronous machine. Regardless of the state of SWAIT, MXWAIT* will be activated with the onset of a valid MIX cycle (IOCYC* active). Once activated however, MXWAIT* will remain active until the falling edge of SWAIT. Once deactivated, MXWAIT* will not be reactivated until the falling edge of IOCYC* is detected.

With this machine, MXWAIT* will always be activated within 40 ns of MXCYC* being activated, however, the IORD* or IOWT* command will not be activated until recovery requirements from the previous access are met (SWAIT goes active).

Remember that the MXWAIT* signal is tri-stated when IOCYC* is inactive. This is necessary because of the bussed nature of the MXWAIT* signal and is accomplished in the PAL specification with the IOCYC* signal. The requirement that MXWAIT* be in its inactive (HIGH) state before it is tri-stated is also met with this design because MXWAIT* will go inactive while both MXCMD* and MXCYC* are asserted. MXWAIT* is then tri-stated by the IOCYC* signal going inactive.

SYNCHRONOUS WAIT MACHINE



ASYNCHRONOUS WAIT MACHINE

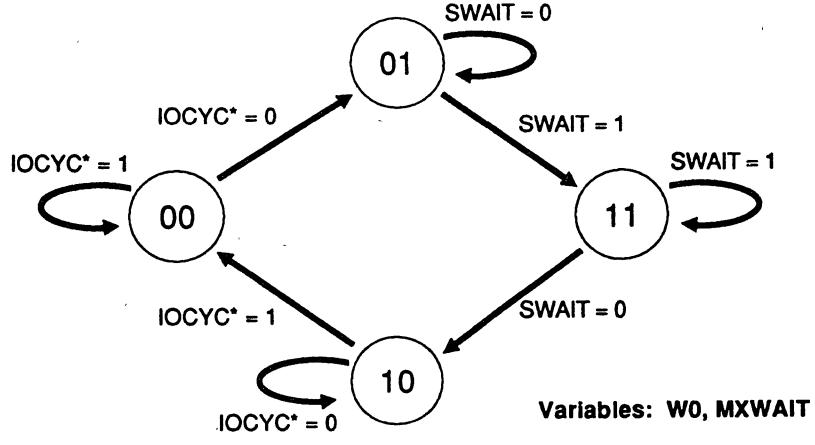


Figure 7. MXWAIT Machine State Diagram

Although the W0 and Sx variables are only used as internal variables to the PAL device, they must be assigned to actual pins on the 22V10 PAL device, bringing the number of output pins on the device to ten. Furthermore, the Sx and SWAIT variables are defined in the PAL source as registered (synchronous) outputs via the ":=" syntax.

From the asynchronous machine states, we can assign a truth table as shown in Table 4. Note that Table 4 gives values for CYC which is the logical state of the IOCYC* signal (IOCYC*=0 - CYC=1). Also "SW" represents the state of SWAIT and WAIT represents the logical value of MXWAIT (i.e. WAIT=1 - MXWAIT*=0). This leads to the equations for W0 and MXWAIT as listed in the PAL source code in Appendix B.

The synchronous machine follows a similar design, although in the synchronous case there are six variables instead of four. The PAL source for the synchronous machine is also listed in Appendix B.

Current State				Next State	
W0	WAIT	CYC	SW	W0	WAIT
0	0	0	0	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	1	0	0	1
0	1	1	1	1	1
1	1	1	1	1	1
1	1	1	0	1	0
1	0	1	0	1	0
1	0	0	0	0	0

Table 4. WAIT State Machine Truth Tables

For the I/O Control PAL's 25 MHz clock source, a 50 MHz crystal oscillator is routed through a D flip-flop which provides a divide by 2 function. The output of the flip-flop is then tied to the clock input pin of the PAL.

The final piece of the I/O control block which must be designed is the logic which supplies the address lines to the peripheral devices. Because the devices require active high address lines, we need only invert the MXA4* through MXA2* signals and route them appropriately. Note that the 82510 is the only device which requires the MXA4* signal.

5.6 Configuration Data Block

The serial EEPROM on the MIX module provides a means with which the baseboard microcontroller may download configuration data during power up. This allows the baseboard's interconnect space to reflect specific configuration data of the MIX modules present on the stack, while allowing the microcontroller firmware to remain standard.

The hardware requirements of the module's serial EEPROM block consist of a single device, a 128 byte 93C46 serial EEPROM. This device is available in an eight pin skinny dip package which may be connected directly to the MIX interface signals. Of the 8 pins, 2 are N/C ("no-connect") pins.

The firmware portions of the serial EEPROM are discussed in Chapter 6 of this document.

5.7 Simulation

Because this module is an extremely simple design and is being prototyped in a wire wrap version, hardware simulation is considered superfluous. In this case, a trace routing or device interaction problem will not cost additional fab production cycles or considerable CAD problems.

The PAL devices however, are easily simulated using the PALASM 2 software. This software includes a state simulator which allows the designer to set the trace mode for various PAL signal outputs and encode various input signal states into a line equation. By changing the input states, the designer can examine the output conditions generated. This is true for both the asynchronous and synchronous PALs designed for this module.

In the simulation for the IODCD PAL, we should investigate both valid and invalid accesses to the peripheral chips. In the PALASM source file for the I/O decode PAL (IODCD) in Appendix B, there is a section marked SIMULATION. The TRACE ON statement lists the signals which are to be examined and the SETF statements are used to change the states of the input signals.

The first SETF statement sets up an address to the PAL of 1xH, which corresponds to PPI1. The next statement specifies a word access is occurring, and finally an indication of a valid I/O cycle is given. Finally, the IOCMD signal is activated to indicate either an IORD* or IOWT* command is active. As shown in the selective trace listing, these inputs result in no chip selects since PPI1 may only be accessed as a byte wide device or in combination with PPI0.

The next group of statements changes the byte enables to a byte access which results in the /CSP1 signal being activated. The next changes the address to 0xH which is the address of PPI0. This leads to /CSP0 and /GATB0 being activated. Note the widths of the gate signals during both read and write accesses. They are longer during writes to provide the data hold requirements of the peripheral devices. The remaining portion of the simulation sets up the byte enables to represent WORD and DWORD transfers to the PPIs. The word transfer results in the /CSP0, /CSP1, /GATB0, and /GAT32 signals being activated, while the DWORD transfer also activates the /CSP2 and /CSP3 signals. These accesses are therefore equivalent to having 16 or 32-bit PPIs.

The simulation for the I/O control PAL follows a similar process. This simulation however, allows the designer to simulate clock ticks as well as input signal state changes. The

simulation is set up by placing the PAL in a reset state. The clock is cycled, then the reset is removed and an I/O read status is set. The LCLSEL*, MXCYC*, and MXCMD* signals are then set to simulate the states the PAL would normally encounter. To check the MXBS16* function, the PPI0 address is simulated on address lines /A6* through /A4*. The PAL is clocked several times to examine the operation of the MXWAIT* and MXBS16* outputs. As shown in the selective trace listing, the WAIT state machines step through their states as designed (MXWAIT* = 0 corresponds toMXWAIT = 1), with the MXWAIT* signal active for four clock pulses. Furthermore, the listing shows MXBS16* inactive throughout the cycle. This is desirable since 32-bit accesses are allowed for PPI0. Also we see IOCYC* go active with the MXCYC* signal and the IORD* signal go active when MXCMD* is activated. This occurs because the SWAIT signal is already active when MXCMD* is activated.

A second MIX transfer cycle is simulated to represent a full speed transfer cycle. In this cycle, we want to assure the proper WAIT machine function for command recovery and further examine the MXBS16* function. To do this, the address of PPI1 is presented to the PAL which should force MXBS16* to activate. We desire this since 32-bit cycles are only allowed for the PPI0 address offset. We also present active MXCYC* and MXCMD* signals to the PAL very soon after the end of the previous transfer cycle. This means MXCYC* has a short inactive time between cycles 1 and 2. This second simulation changes the polarity of the MXWR signal to represent a write request.

Once MXCYC* is activated we see both MXBS16* and MXWAIT* immediately activated. This ensures us we provide MXBS16* and MXWAIT* within the allotted 40 ns. In this cycle, when the command signal (MXCMD*) is activated the IOWT* signal does not come on. This occurs because the synchronous machine has not yet cycled through all of its inactive states. In the selective trace listing we see the synchronous machine continue through its inactive states even though there is a valid transfer cycle occurring. Once the machine reaches the 0001 state, IOWT* is activated. This has the effect of holding off the write command until the synchronous machine is ready to acknowledge it. Therefore, this is the command recovery mechanism built into the WAIT machine.

With acceptable PAL code simulations completed we can turn our attention to firmware requirements and module prototyping.

6. MODULE FIRMWARE

The MIX architecture has several firmware requirements for modules to assure proper interaction with MIX compatible baseboards and other modules. These requirements place the responsibility of supplying necessary configuration data and BIST tests on the module. While the module is responsible for supplying it, configuration data for the MIX modules is treated as an extension of the baseboard's interconnect space. This allows a MIX baseboard to use standard microcontroller code while allowing PSB agents to determine the type of MIX modules present via the baseboard's interconnect space. This structure allows dissimilar MIX modules to stack together without the need for modifying the baseboard's interconnect configuration data. Module resident BIST code allows the baseboard to test the hardware on the modules without the need for custom firmware on the baseboard. This allows different modules to stack together without the necessity of custom BIST code on the baseboard.

6.1 Serial EEPROM

The requirement placed on modules for interconnect space is the support of the National Microwire™ standard serial EEPROM. This is a 128 byte device which is arranged in a 64 x 16 bit configuration which allows random access of 64 word registers over four signal lines. It is programmed to contain two interconnect records which describe the configuration of the MIX module. Each interconnect register is a two byte value. The MSB contains the register protection information and the LSB contains the function's value. The baseboard's 8751 microcontroller downloads this information during power up and appends it to its own interconnect data. The *MIX Module Design Specification* describes the format which the EEPROM firmware must follow in order to function with the baseboard's 8751.

The standard format has two interconnect records, the Hardware Extension Record and the Module Specific Record as shown in Table 5. A custom module uses the standard template as a guide to defining the module's interconnect data. Required values for the registers are included in Table 5. Those registers listed as TBS in the value column have register values "To Be Specified" by the module designer.

This module defines its registers as shown in the "EX" column of Table 5. The vendor ID, 0001 (0900, 0901), represents the Intel vendor number for MULTIBUS II. Intel administers the assignment of specific vendor numbers to MULTIBUS II board manufacturers.

This Module's ID has been defined to be "TM_MIX_MOD", with the "T" occupying register 6 and the "D" occupying register 0FH. Note that the ASCII equivalent of the string is placed in the Module ID registers.

Because this module is a wire wrap prototype, all revision level registers are given the value of 0. Finally, since this module was not designed with EPROM devices or self test capabilities, its BIST support level is defined as 0. This level means either the module hardware is not tested during power up or the baseboard firmware is responsible for supplying the module BIST code. In the case of this module, the former is true. A BIST support level of 0 implies that the Jump Table Pointer registers are assigned a value of 0.

Now that the register values are defined, the EEPROM device must simply be programmed and installed on the module. All download functions are handled by the baseboard microcontroller and do not concern the module designer.

Register Offset	Description	Access Rights Local	Global	Value	Ex
	HARDWARE EXTENSION RECORD				
0	H/W Ext. Record Type	R/O	R/O	0914H	0914H
1	Record Length	R/O	R/O	0913H	0913H
2	H/W Ext. Type (LSB)	R/O	R/O	0910H	0910H
3	H/W Ext. Type (MSB)	R/O	R/O	0900H	0900H
4	Vendor ID, Low	R/O	R/O	TBS	0901H
5	Vendor ID, High	R/O	R/O	TBS	0900H
6-0FH	Module ID, Char 1-10	R/O	R/O	TBS	*
10H	H/W Test Rev Level	R/O	R/W1	TBS	2900H
11H	Number of Records	R/O	R/O	0901H	0901H
12H	RFU	R/O	R/O	0900H	0900H
13H	RFU	R/O	R/O	0900H	0900H
14H	RFU	R/O	R/O	0900H	0900H
	MODULE SPECIFIC RECORD				
15H	Mod. Specific Record Type	R/O	R/O	09F0H	09F0H
16H	Record Length	R/O	R/O	TBS	0909H
17H	EPROM Entry Table Ptr., MSB	R/O	R/O	TBS	0900H
18H	EPROM Entry Table Ptr., Byte 3	R/O	R/O	TBS	0900H
19H	EPROM Entry Table Ptr., Byte 2	R/O	R/O	TBS	0900H
1AH	EPROM Entry Table Ptr., LSB	R/O	R/O	TBS	0900H
1BH	RFU	R/O	R/O	0900H	0900H
1CH	H/W Rev Level	R/O	R/W1	TBS	2900H
1DH	BIST Test Support Level	R/O	R/W1	TBS	2900H
1EH	RFU	R/O	R/O	0900H	0900H
1FH	RFU	R/O	R/O	0900H	0900H
20H	RFU	R/O	R/O	0900H	0900H
21H-3FH	Module Specific Info	Module Defined			

Table 5. MIX Module Interconnect Template

The 93C64 device can be programmed using a standard EEPROM programmer which supports the device. Because the device is electrically erasable, it is possible for the EEPROM to be programmed by the baseboard's microcontroller. At the time of this module design, this support was not included in the baseboard 8751 firmware.6.2 BIST Support Levels

The MIX architecture provides several BIST support levels as described in the *MIX Module Design Specification.* Inclusion of EPROMs and BIST firmware on modules allows power up verification of the module hardware and provides diagnostic capabilities when module hardware problems are encountered.

This module does not incorporate EPROM or BIST support in its design. This is defined as BIST support level 0. Although level 0 allows the stipulation that the MIX baseboard carry the module BIST code in its EPROM, this module design assumes no power up testing is performed on the module hardware. This option was chosen because of the simplicity of the module hardware used in the design. Furthermore, addition of module EPROMs would have further complicated the I/O control block portion of the design, which would have lessened the simple example intentions of this module design.

7. MODULE PROTOTYPING

Once the module is fully defined and its schematic has been completed, a final check of the modules electrical characteristics can be made. The main concern here is the power requirements of the module. We should ensure the devices used in this design meet the power consumption requirements listed for the MIX interface. Once we have assured these characteristics, layout and actual wire wrapping and debug can be done.

7.1 Electrical Considerations

Before the actual module assembly begins, we must check the module design against the electrical specifications for the MIX interface. The two major issues here involve the drive characteristics of the devices which connect to the MIX signals and the total maximum power consumption of the

module. Both of these parameters may affect how the module operates with the MIX baseboard or other modules.

The drive characteristic check deals with the type of driver used on the MIX bus. The MIX specification states that no module may use open collector devices to drive the MIX bus signals. A quick check of the module device specifications assure us that the module meets this requirement.

The second area of concern is the power consumption of the module. The *MIX Module Design Specification* states that each module may consume up to 3 Amps of +5 VDC. Again, by going through each device's spec sheet we can compute the maximum and typical power consumption of the module.

In reviewing the components used for this module, we find the maximum current draw on the +5 V power is just under 2.5 Amps for the entire module, including the LEDs. Although we find this simple module is surprisingly near the 3 Amp limit, a typical module design would not implement 32 LEDs which draw about 25 mA each.

7.2 Wire Wrap Layout

Now that the electrical requirements of the MIX module are verified, actual prototyping will be done. The layout portion of the prototyping requires knowledge of the signal paths and component interconnection. The schematic capture program used in this design contains a program which generates a netlist from the schematic output. With this netlist and the MIX MOD2 Breadboard Module, we are ready to begin the layout.

In laying out components for the MIX design, careful attention should be given to the trace length requirements given in the *MIX Module Design Specification.* Although this is merely a wire wrap version of the final product, adhering to the trace length rules may save some debugging headaches later. The Breadboard module offers the standard MIX module width and MIX interface footprint, however, the board length is much longer than a standard module. This added length is provided to allow an engineer to prototype surface mount designs using DIP components which require more design area.

The module layout follows the signal paths between the MIX interface and I/O devices. The termination resistors and data transceivers are placed nearest the MIX connector. The two PAL devices and EEPROM also reside adjacent to the connector due to their direct connection to interface signals. The PPI devices and 82510 are placed next, and the output drivers, LEDs, and connectors are placed at the edges of the board. This layout structure provides a natural signal propagation from the MIX interface out to the edges of the module and ensures the trace lengths are kept to a minimum. Once the components are placed, they may be connected via wire wrap.

7.3 Module Debug

Once the module has been completely wired up, it is ready for verification and debug. Although most designers are confident to immediately initiate a "smoke test", it is wiseto verify the device connections with the netlist provided by the schematic capture package. This is time consuming but is easily done using a simple ohmmeter. Once this has been checked out, the module may be installed on the baseboard for final debug.

The final validation for this module was performed on a MIX 386/M0xPP Pre-production baseboard with a MIX MOD1 Test Module in module slot 0 and the wire wrap module in slot 1. In this stacking configuration, the wire wrap module's I/O address space ranges from 0800H to 0C00H. This means the I/O address offsets encoded in the IODCD PAL are added to the base address of 0800H to obtain the actual addresses for the module's devices. For example, the baseboard 386 microprocessor would access PPI0 at 098xH.

To facilitate ease of baseboard I/O access, firmware was placed on the MIX baseboard which contained a DMON 386 monitor configured (with the proper 82510 port address) to run with MIX MOD1 in module slot 0. This provides a serial interface for the baseboard and removes the need for additional system boards (except the CSM/001) for the module debug.

In debugging the board, we must check out several things, access to the boards serial EEPROM, parallel port functionality, and 82510 functionality. The serial EEPROM interface is verified by reading the baseboards interconnect space. DMON386 provides an input interconnect (ii) function which aids in this task. The *MIX 386/M0xPP Baseboard User's Guide* shows the first H/W Extension Record should begin at interconnect offset 65H. This register can be checked for a value of 14H. If the 14H is found, the module ID registers (6BH - 74H) can be read. If the module's ID is present (in ASCII), the serial interface is functioning.

The parallel ports may be verified by using the LEDs designed on the module's PPI port C or by using a logic analyzer. Using the LEDs requires setting up port C of the PPI as a simple output port and then making output accesses to port C to toggle the LEDs. To use a logic analyzer, direct accesses to the PPIs may be checked, or the port outputs may be checked. To verify the port A connector, ports A and B must be configured as an output port since bits 0 and 1 of port B control the direction and enable pin of the port A drivers.

The addressing scheme used in this module places the port A data bus at x0H, the port B data bus at x4H, port C's data at x8H, and the PPI's control port at xCH. For the PPI verification, all PPI ports are set up as simple output channels. This is done by writing 80H to ports 098CH, 099CH, 09ACH, and 09BCH. As shown on sheet 10 of the schematic, port B bits 0 and 1 affect the function of the port A driver chip. Therefore, a 3 is output to 0984H to enable the port A drivers and set their direction as outputs. Data is then output to both ports A and C to verify the module operation.

The 82510 device may be similarly checked out. With the module in slot 1, the 82510 internal registers begin at I/O address 09C0H. The chip can be verified by setting its 35 internal registers to configure a loopback mode to verify data transmit and receive data consistency.

Once the 82510, PPIs, and EEPROM operation have been verified, a high level of design confidence is achieved, and production cycles may proceed as needed.

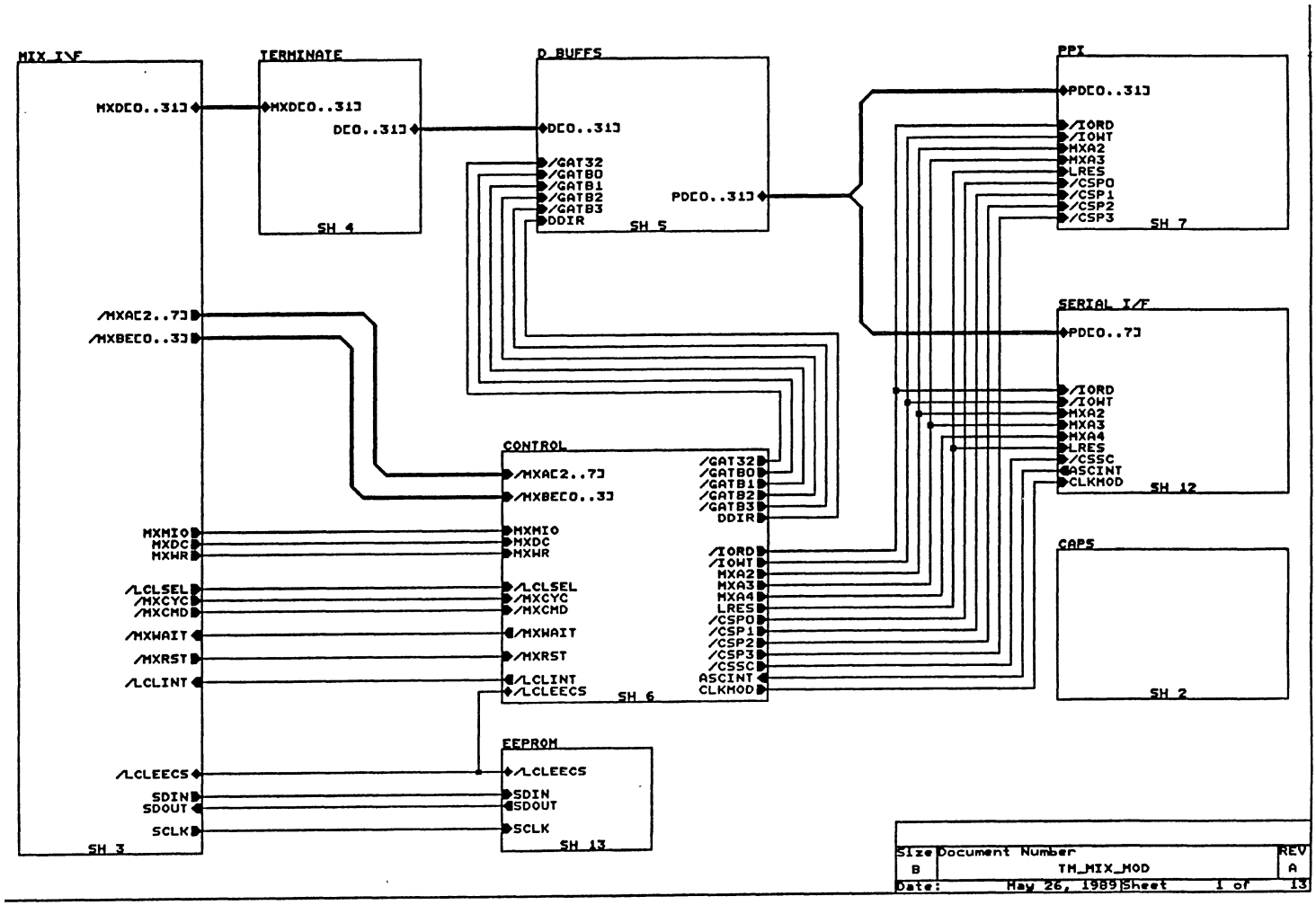
8. CONCLUSIONS

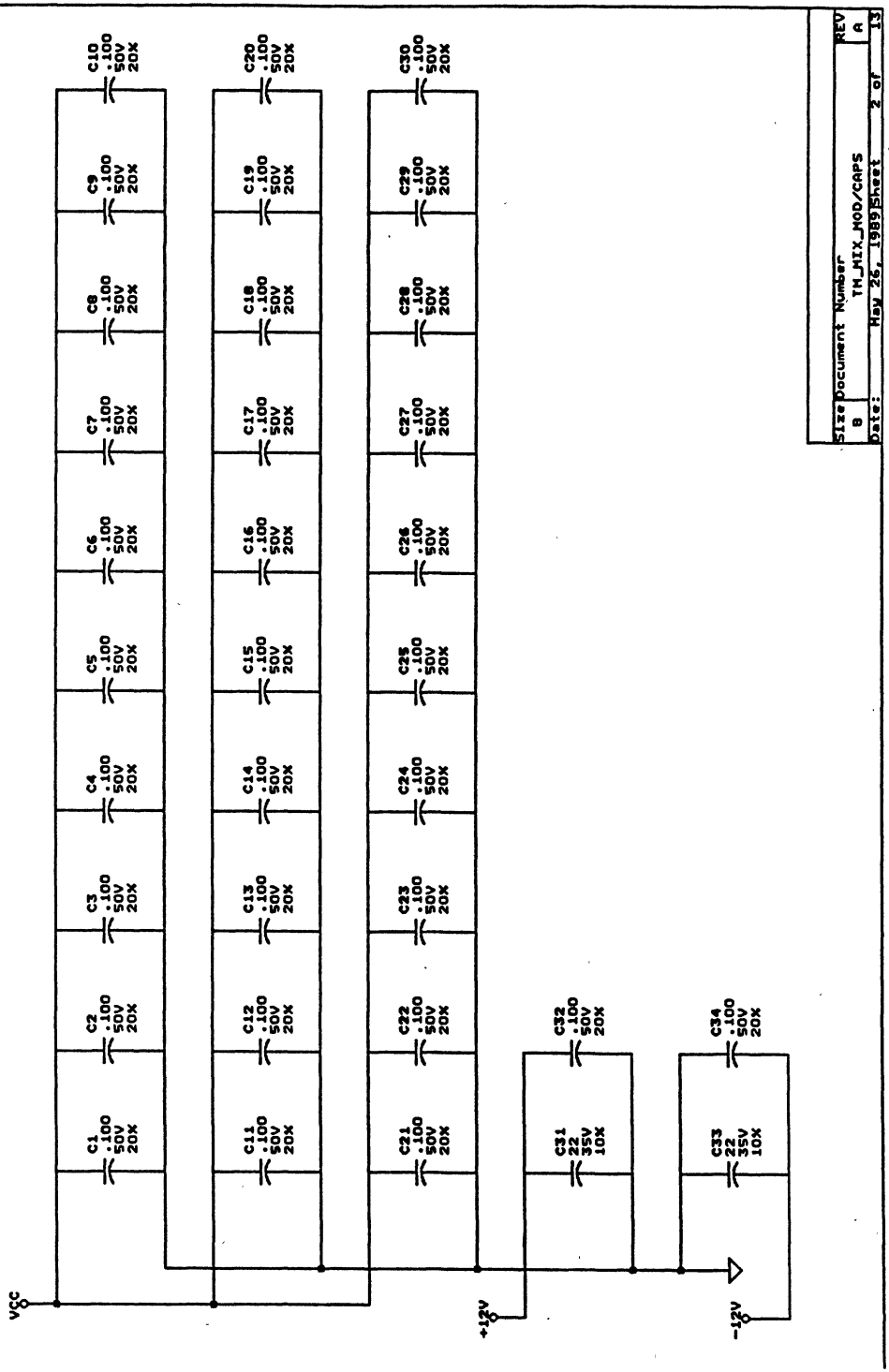
Although this module design is a simple one and has been done simply for display and example purposes, it conveys the considerations required in a MIX module design effort. Furthermore, the control and data drivers blocks of the module may be used as a design guide in many simple slave I/O module designs.

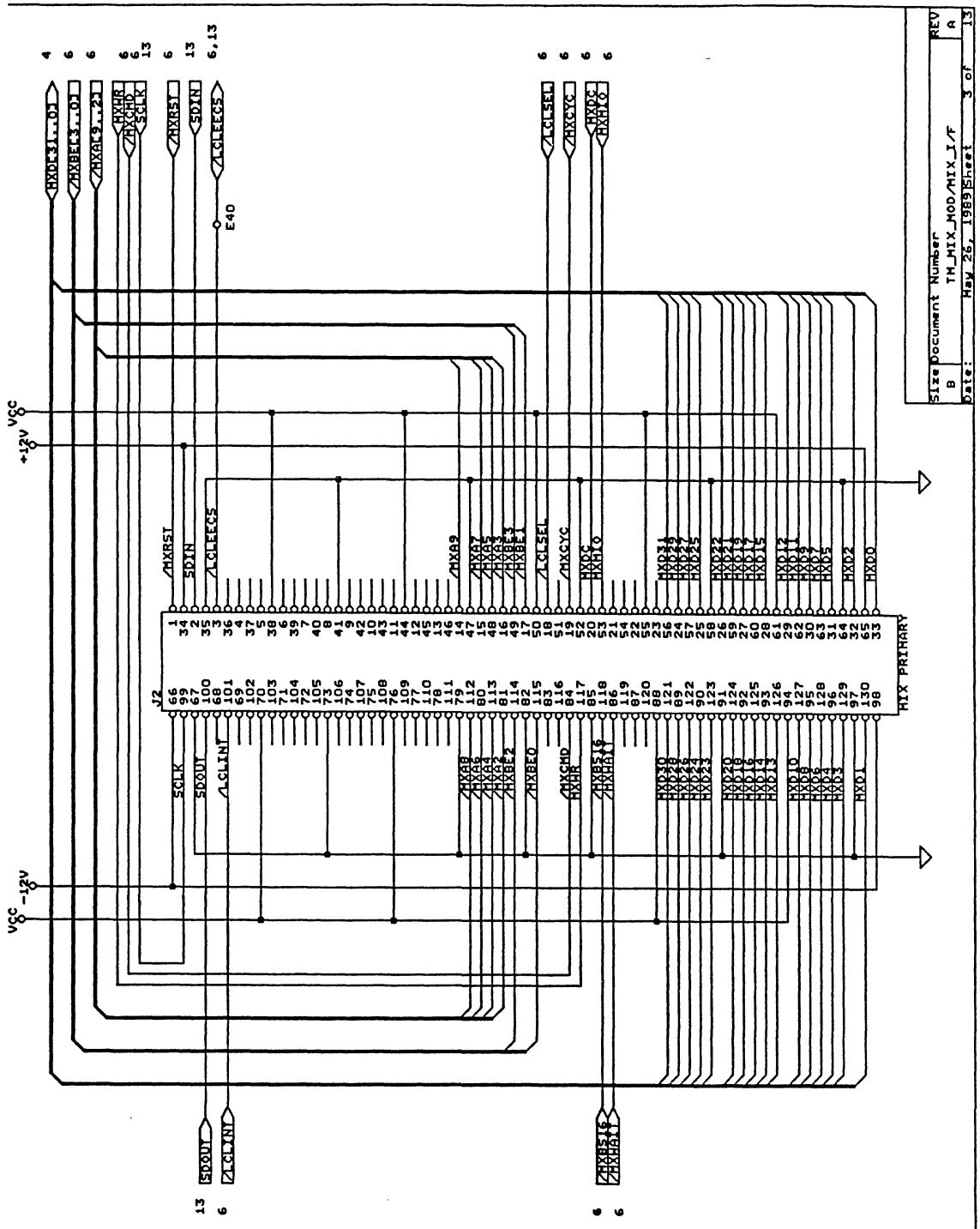
The module schematics and PAL source codes have been included in Appendices A and B as an aid to designing simple I/O modules and as a basis for more complex module designs.

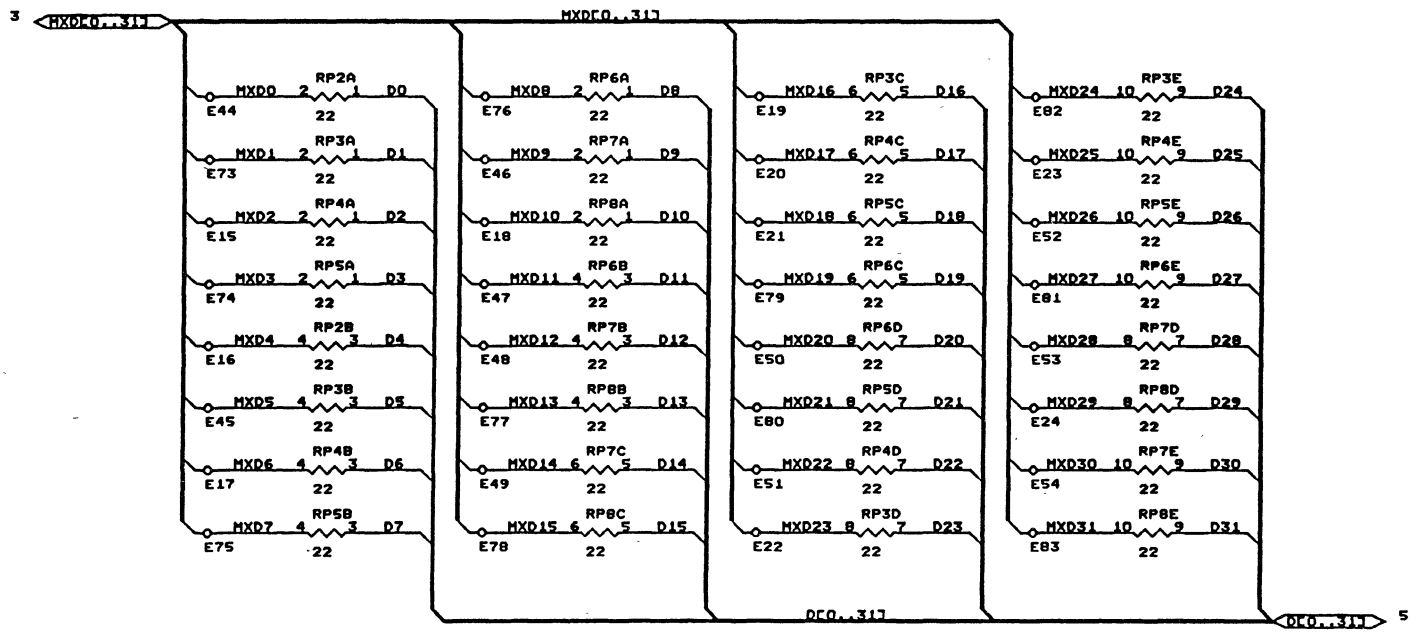
APPENDIX A

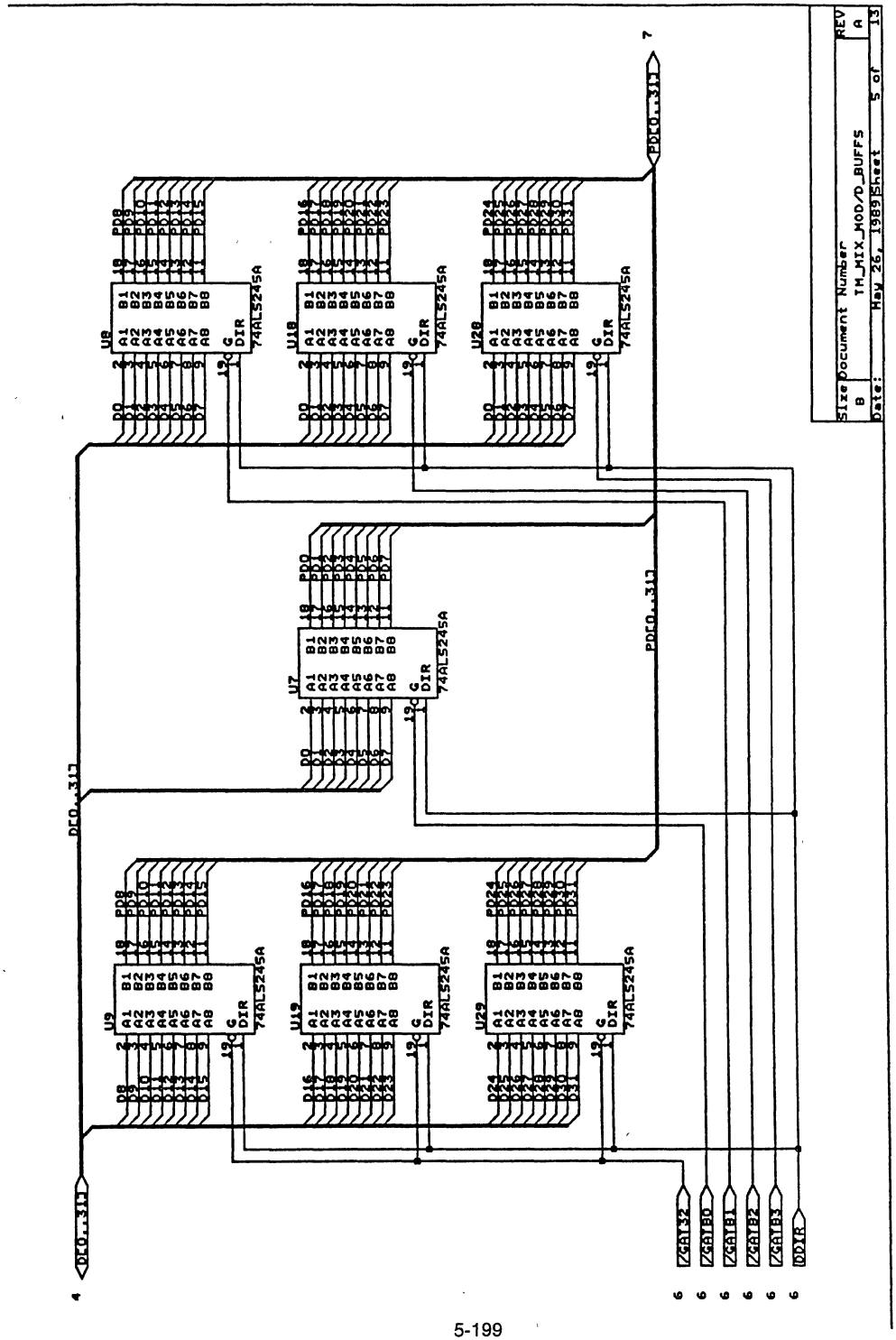
MIX MODULE SCHEMATICS

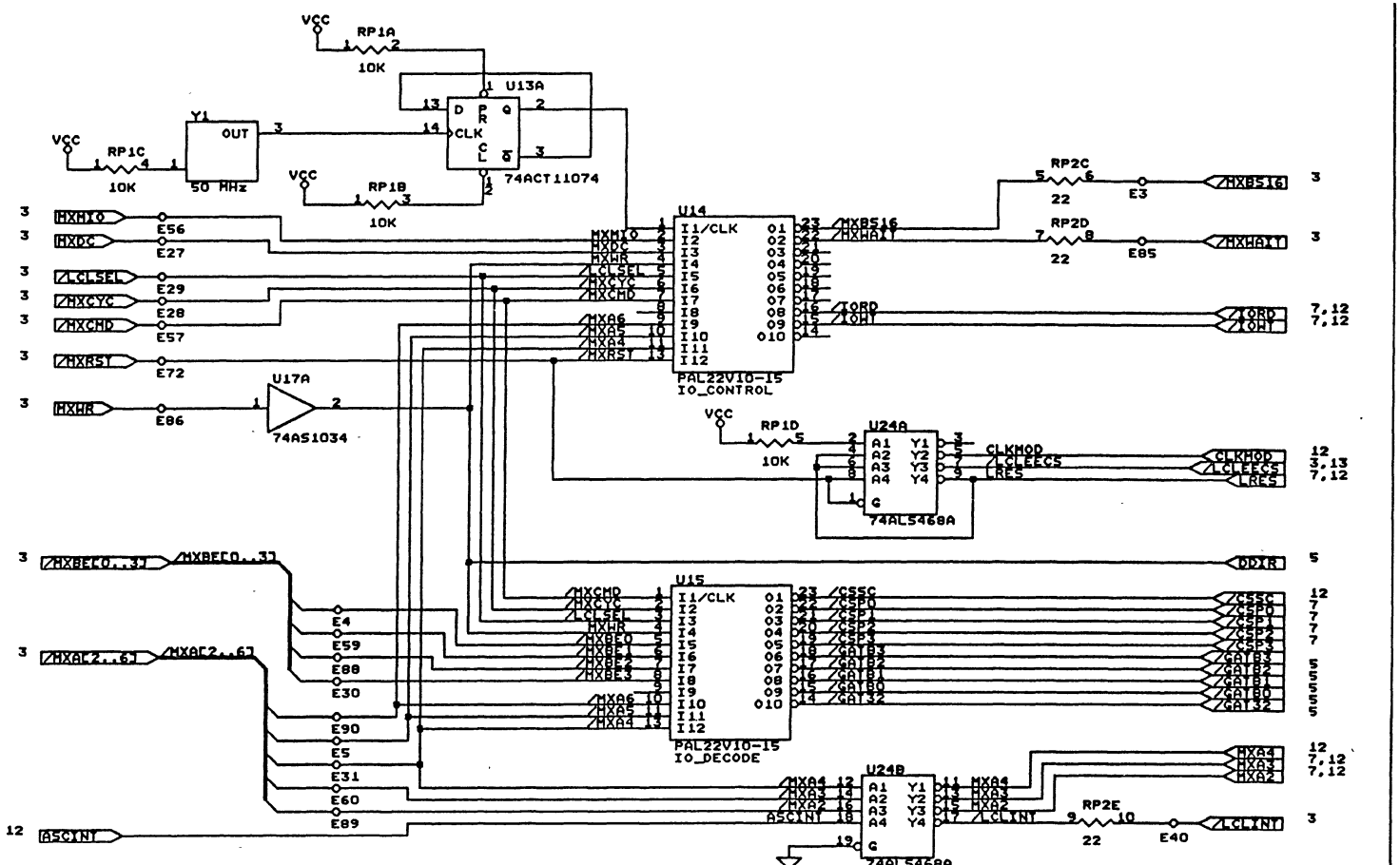

MIX I\F
MXDCO..31J
/MXAC2..7J
/MXBECO..3J
MXMIO
MXDC
MXWR
/LCLSEL
/MXCYC
/MXCMD
/MXWAIT
/MXRST
/LCLINT
/LCLEECS
SDIN
SDOUT
SCLK
SH 3
TERMINATE
MXDCO..31J
DCO..31J
SH 4
D BUFFS
DCO..31J
/GAT32
/GATB0
/GATB1
/GATB2
/GATB3
DDIR
PDCO..31J
SH 5
CONTROL
/MXAC2..7J
/MXBECO..3J
MXMIO
MXDC
MXWR
/LCLSEL
/MXCYC
/MXCMD
/MXWAIT
/MXRST
/LCLINT
/LCLEECS
/GAT32
/GATB0
/GATB1
/GATB2
/GATB3
DDIR
/IORD
/IOWT
MXA2
MXA3
MXA4
LRES
/CSP0
/CSP1
/CSP2
/CSP3
/CSSC
ASCINT
CLKMOD
SH 6
EEPROM
/LCLEECS
SDIN
SDOUT
SCLK
SH 13
PPI
PDCO..31J
/IORD
/IOWT
MXA2
MXA3
LRES
/CSP0
/CSP1
/CSP2
/CSP3
SH 7
SERIAL I/F
PDCO..7J
/IORD
/IOWT
MXA2
MXA3
MXA4
LRES
/CSSC
ASCINT
CLKMOD
SH 12
CAPS
SH 2
Size B
Document Number TM_MIX_MOD
REV A
Date: May 26, 1989 Sheet 1 of 13


C10 .100 50V 20X
C9 .100 50V 20X
C8 .100 50V 20X
C7 .100 50V 20X
C6 .100 50V 20X
C5 .100 50V 20X
C4 .100 50V 20X
C3 .100 50V 20X
C2 .100 50V 20X
C1 .100 50V 20X
C20 .100 50V 20X
C19 .100 50V 20X
C18 .100 50V 20X
C17 .100 50V 20X
C16 .100 50V 20X
C15 .100 50V 20X
C14 .100 50V 20X
C13 .100 50V 20X
C12 .100 50V 20X
C11 .100 50V 20X
C30 .100 50V 20X
C29 .100 50V 20X
C28 .100 50V 20X
C27 .100 50V 20X
C26 .100 50V 20X
C25 .100 50V 20X
C24 .100 50V 20X
C23 .100 50V 20X
C22 .100 50V 20X
C21 .100 50V 20X
C32 .100 50V 20X
C31 22 35V 10X
C34 .100 50V 20X
C33 22 35V 10X
VCC
+12V
-12V
Size Document Number REV
B TM_MIX_MOD/CAPS A
Date: May 26, 1989 Sheet 2 of 13

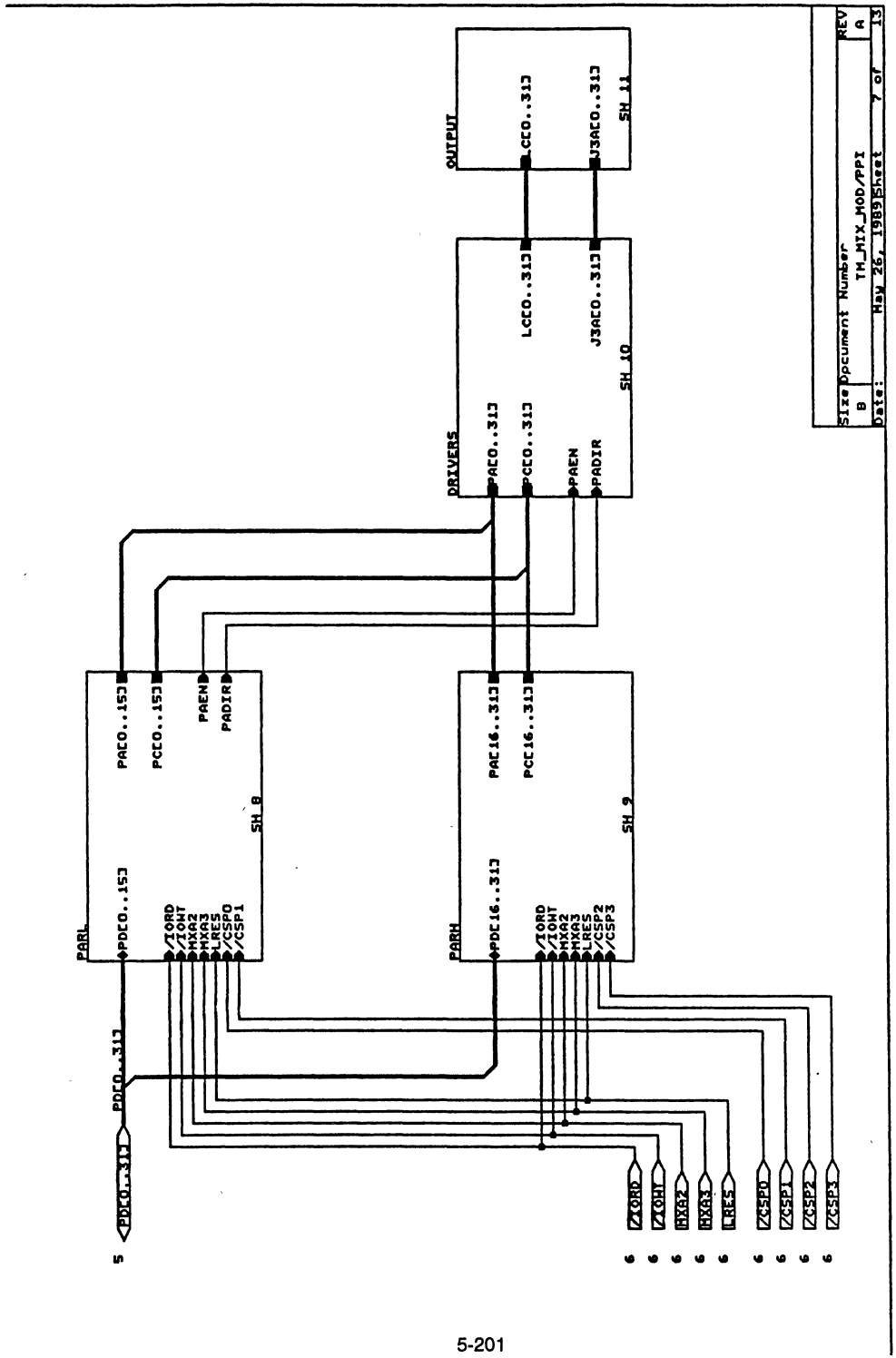

MIX PRIMARY
J2
Size Document Number REV
B TM_MIX_MOD/MIX_I/F A
Date: May 26, 1989 Sheet 3 of 13

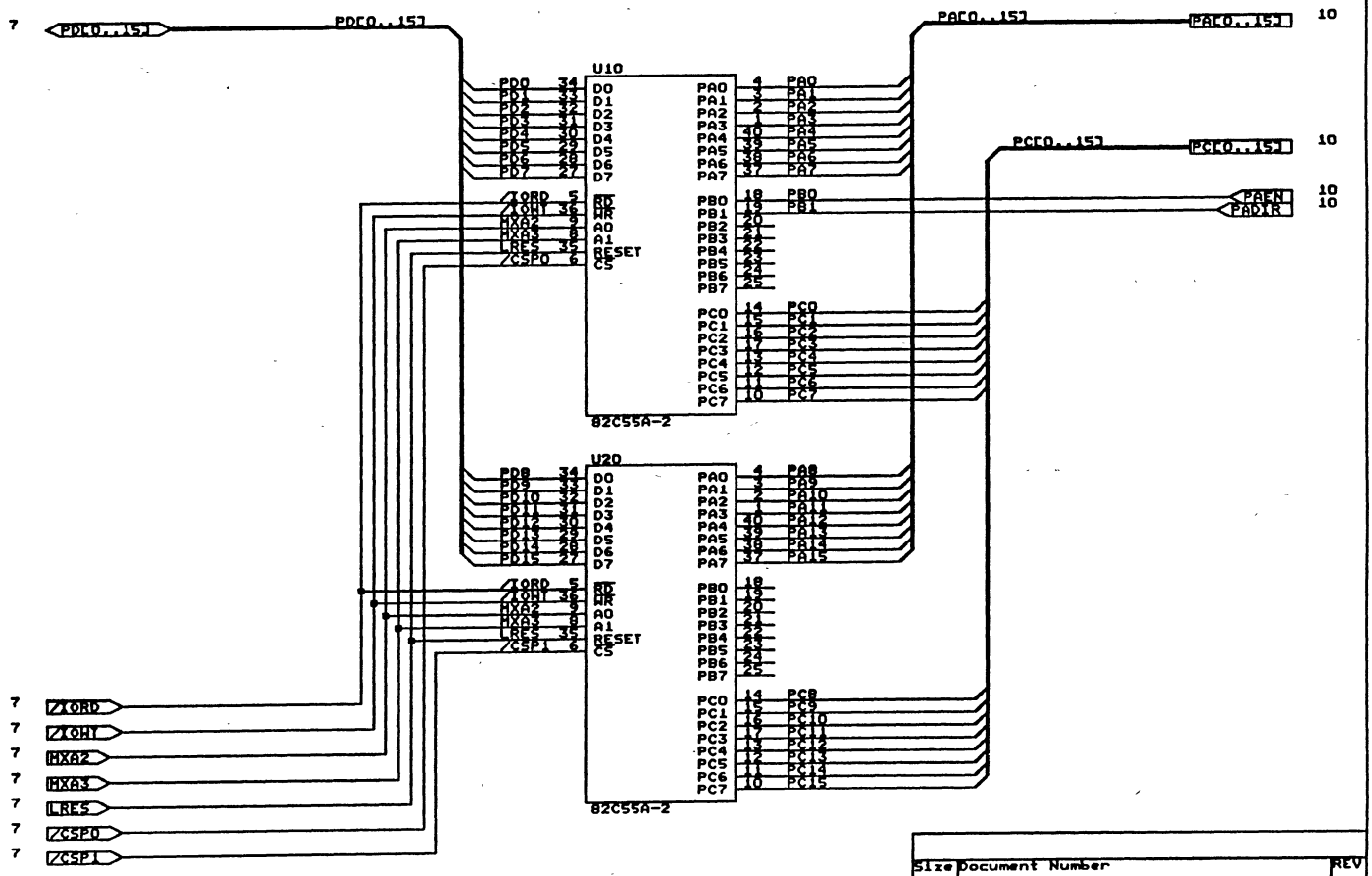

3
MXDI0..31
MXDI0..31
RP2A
MXD0 2 1 D0
E44
22
RP3A
MXD1 2 1 D1
E73
22
RP4A
MXD2 2 1 D2
E15
22
RP5A
MXD3 2 1 D3
E74
22
RP2B
MXD4 4 3 D4
E16
22
RP3B
MXD5 4 3 D5
E45
22
RP4B
MXD6 4 3 D6
E17
22
RP5B
MXD7 4 3 D7
E75
22
RP6A
MXD8 2 1 D8
E76
22
RP7A
MXD9 2 1 D9
E46
22
RP8A
MXD10 2 1 D10
E18
22
RP6B
MXD11 4 3 D11
E47
22
RP7B
MXD12 4 3 D12
E48
22
RP8B
MXD13 4 3 D13
E77
22
RP7C
MXD14 6 5 D14
E49
22
RP8C
MXD15 6 5 D15
E78
22
RP3C
MXD16 6 5 D16
E19
22
RP4C
MXD17 6 5 D17
E20
22
RP5C
MXD18 6 5 D18
E21
22
RP6C
MXD19 6 5 D19
E79
22
RP6D
MXD20 8 7 D20
E50
22
RP5D
MXD21 8 7 D21
E80
22
RP4D
MXD22 8 7 D22
E51
22
RP3D
MXD23 8 7 D23
E22
22
RP3E
MXD24 10 9 D24
E82
22
RP4E
MXD25 10 9 D25
E23
22
RP5E
MXD26 10 9 D26
E52
22
RP6E
MXD27 10 9 D27
E81
22
RP7D
MXD28 8 7 D28
E53
22
RP8D
MXD29 8 7 D29
E24
22
RP7E
MXD30 10 9 D30
E54
22
RP8E
MXD31 10 9 D31
E83
22
MXDI0..31
DI0..31
5
Size B
Document Number TM_MIX_MOD/TERMINATE
REV A
Date: May 26, 1989
Sheet 4 of 13

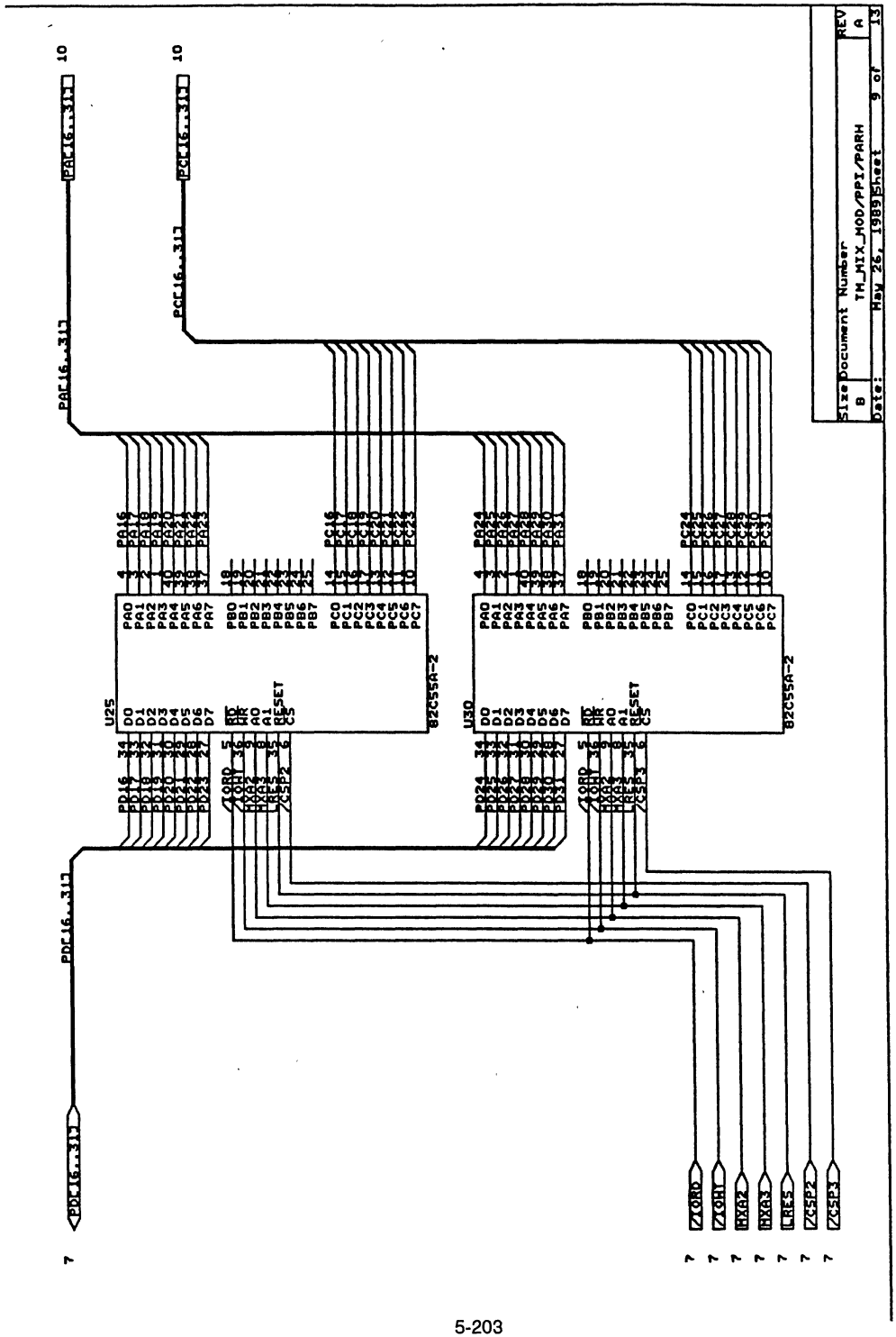

Size Document Number REV
B TM_MIX_MOD/D_BUFFS A
Date: May 26, 1989 Sheet 5 of 13
74ALS245A
U8
U18
U28
U7
U9
U19
U29
PD0..31
DD0..31
ZDAT32
ZDAT80
ZDAT81
ZDAT82
ZDAT83
DDIR

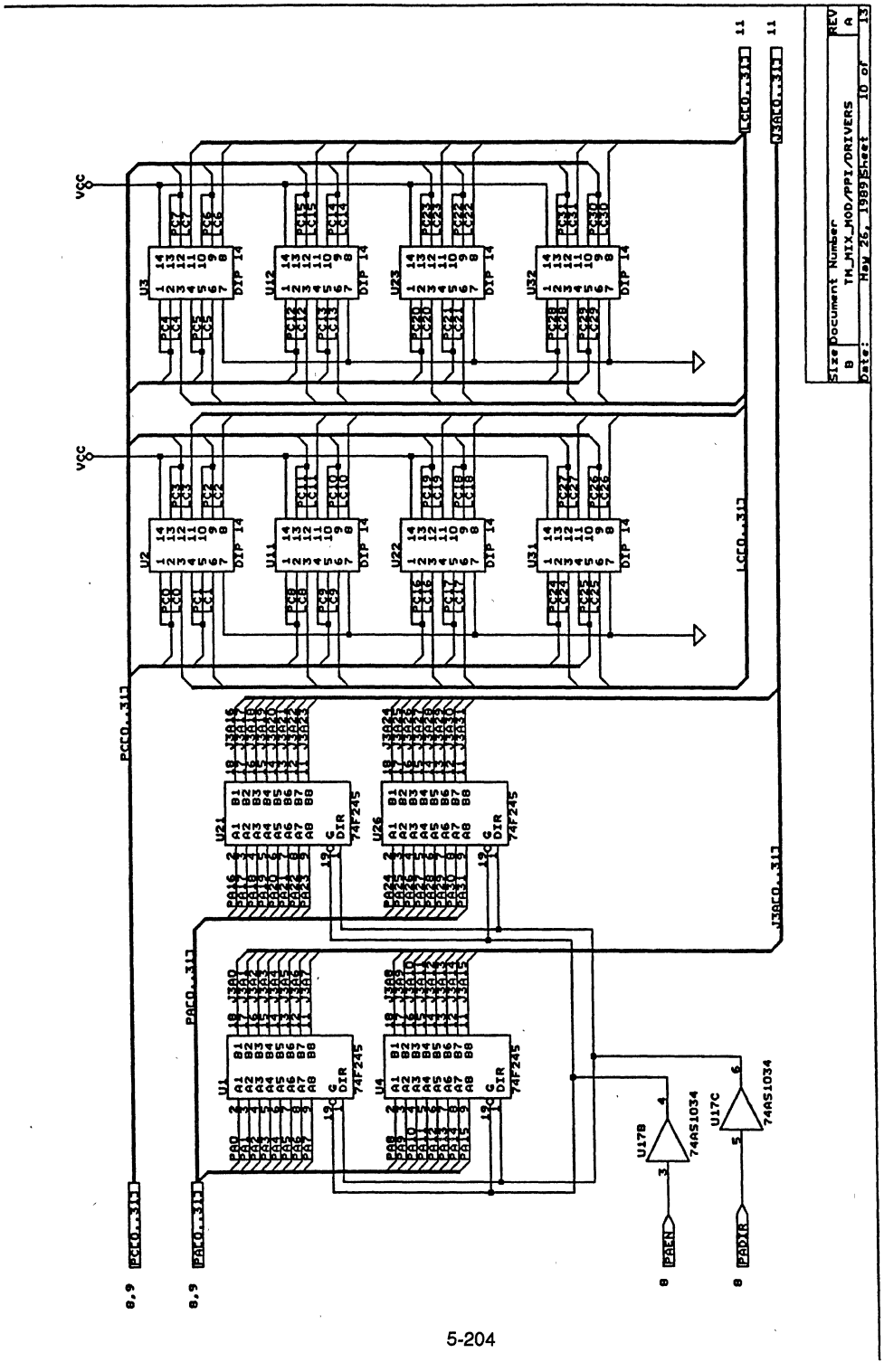


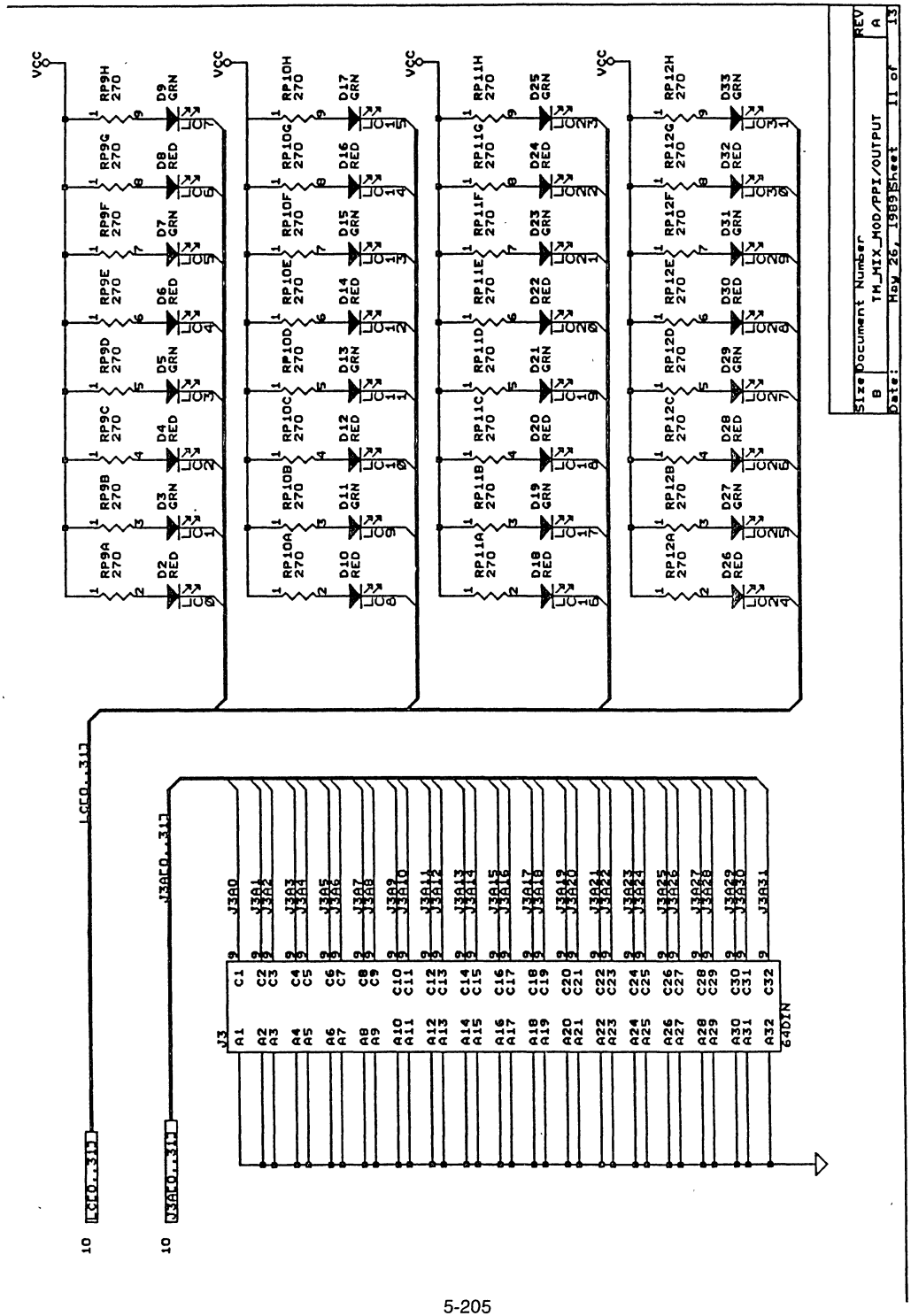
Size	Document Number	REV
B	TM_MIX_MOD/CONTROL	A
Date: May 26, 1989	Sheet 6 of 13	

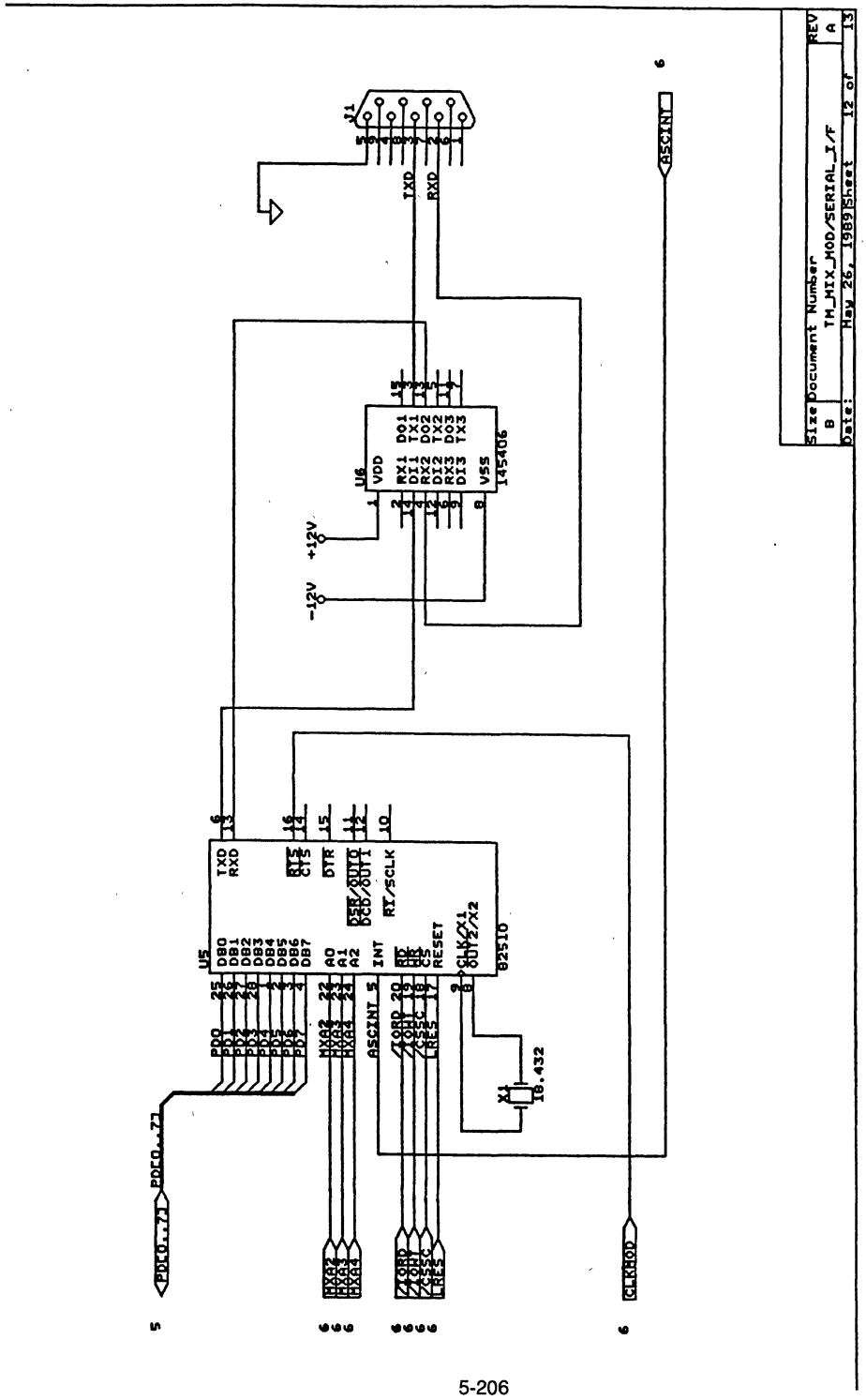

OUTPUT
LCC0..31J
J3AC0..31J
SH_11
DRIVERS
PAC0..31J
PCC0..31J
PAEN
PADIR
SH_10
PARL
PDC0..15J
PAC0..15J
PCC0..15J
/IORD
/IOWT
MXA2
MXA3
LRES
/CSP0
/CSP1
SH_8
PARH
PDC16..31J
PAC16..31J
PCC16..31J
/CSP2
/CSP3
SH_9
Size
B
Document Number
TM_MIX_MOD/PPI
Date:
May 26, 1989
Sheet
7 of
13
REV
A

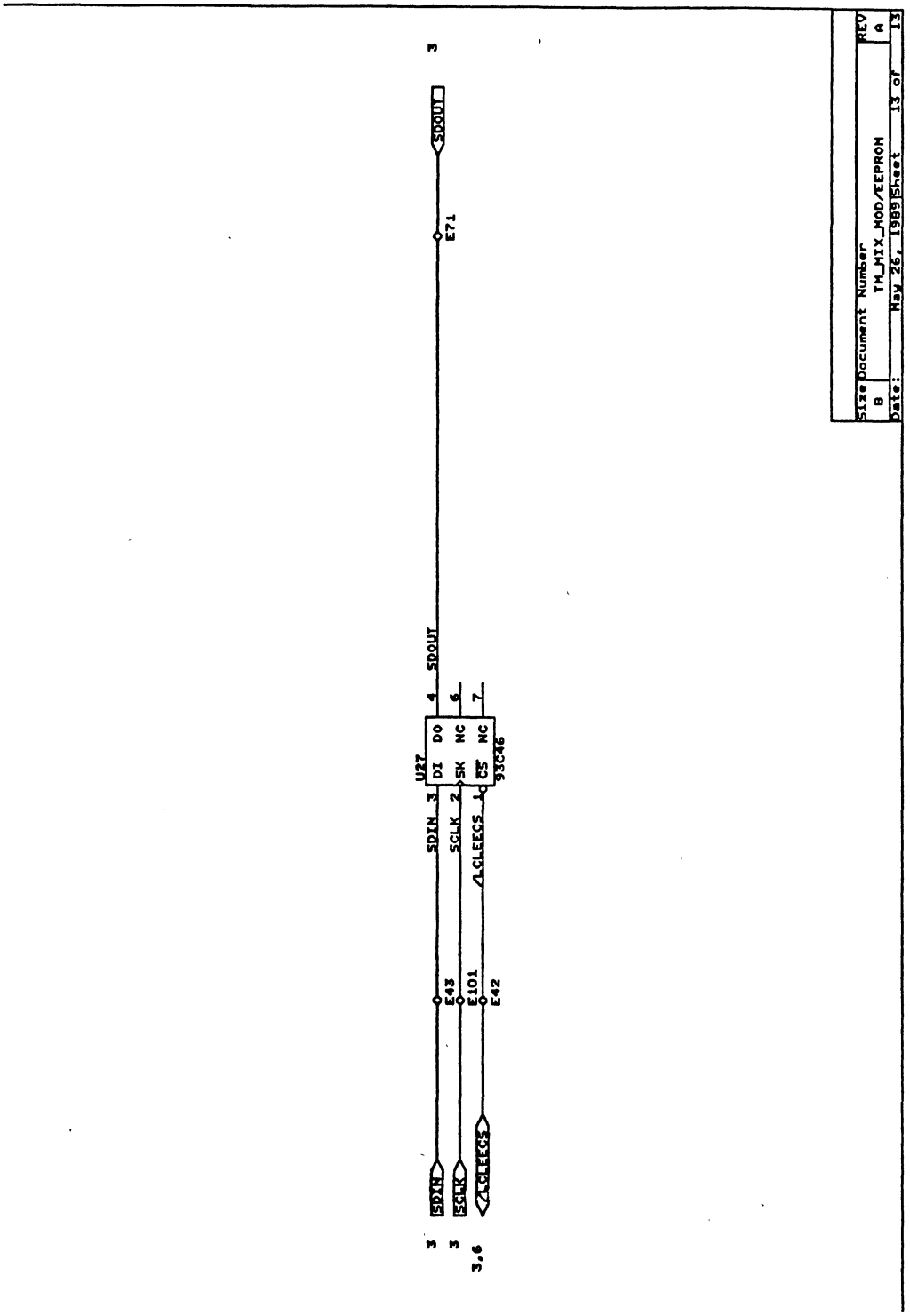

PDI0..15]
PDC0..15]
U10
PD0 34
PD1 33
PD2 32
PD3 31
PD4 30
PD5 29
PD6 28
PD7 27
D0
D1
D2
D3
D4
D5
D6
D7
/IORD 5
/IOWT 36
HXA2 9
HXA3 8
LRES 35
/CSP0 6
RD
WR
A0
A1
RESET
CS
PA0 4
PA1 3
PA2 2
PA3 1
PA4 40
PA5 39
PA6 38
PA7 37
PB0 18
PB1 19
PB2 20
PB3 21
PB4 22
PB5 23
PB6 24
PB7 25
PC0 14
PC1 15
PC2 16
PC3 17
PC4 13
PC5 12
PC6 11
PC7 10
82C55A-2
PAC0..15]
PAI0..15]
PCC0..15]
PCI0..15]
PAEN
PADIR
U20
PD8 34
PD9 33
PD10 32
PD11 31
PD12 30
PD13 29
PD14 28
PD15 27
/IORD 5
/IOWT 36
HXA2 9
HXA3 8
LRES 35
/CSP1 6
PA8 4
PA9 3
PA10 2
PA11 1
PA12 40
PA13 39
PA14 38
PA15 37
PC8 14
PC9 15
PC10 16
PC11 17
PC12 13
PC13 12
PC14 11
PC15 10
/IORD
/IOWT
HXA2
HXA3
LRES
/CSP0
/CSP1
Size B
Document Number TM_MIX_MOD/PPI/PARL
REV A
Date: May 26, 1989 Sheet 8 of 13


PAC16..31
PCC16..31
PAC16..31
PCC16..31
PDC16..31
PDC16..31
U25
U30
82C55A-2
82C55A-2
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
D0
D1
D2
D3
D4
D5
D6
D7
RD
WR
A0
A1
RESET
CS
ZCORD
ZCOWR
HXA2
HXA3
LRES
ZCSP2
ZCSP3
Size
B
Document Number
TM_MIX_MOD/PPI/PARH
REV
A
Date:
May 26, 1989
Sheet
9 of 13


VCC
VCC
U3
U12
U23
U32
U2
U11
U22
U31
DIP 14
PCC0..3]1
PAC0..3]1
LCC0..3]1
U38C0..3]1
U21
U26
U1
U4
74F245
U17B
U17C
74AS1034
PAEN
PADIR
Size
Document Number
TM_MIX_MOD/PPI/DRIVERS
REV
A
Date:
May 26, 1989
Sheet
10 of
13


VCC
RP9A 270
RP9B 270
RP9C 270
RP9D 270
RP9E 270
RP9F 270
RP9G 270
RP9H 270
D2 RED
D3 GRN
D4 RED
D5 GRN
D6 RED
D7 GRN
D8 RED
D9 GRN
RP10A 270
RP10B 270
RP10C 270
RP10D 270
RP10E 270
RP10F 270
RP10G 270
RP10H 270
D10 RED
D11 GRN
D12 RED
D13 GRN
D14 RED
D15 GRN
D16 RED
D17 GRN
RP11A 270
RP11B 270
RP11C 270
RP11D 270
RP11E 270
RP11F 270
RP11G 270
RP11H 270
D18 RED
D19 GRN
D20 RED
D21 GRN
D22 RED
D23 GRN
D24 RED
D25 GRN
RP12A 270
RP12B 270
RP12C 270
RP12D 270
RP12E 270
RP12F 270
RP12G 270
RP12H 270
D26 RED
D27 GRN
D28 RED
D29 GRN
D30 RED
D31 GRN
D32 RED
D33 GRN
LCS0..3]
J3A0..3]
10
J3
64DIN
Size B
Document Number TM_MIX_MOD/PPI/OUTPUT
REV A
Date: May 26, 1989 Sheet 11 of 13


J1
TXD
RXD
ASCINT
U6
VDD
RX1 DO1
DI1 TX1
RX2 DO2
DI2 TX2
RX3 DO3
DI3 TX3
VSS
145406
+12V
-12V
U5
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
TXD
RXD
RTS
CTS
DTR
DSR/OUT2
DCD/OUT1
RI/SCLK
A0
A1
A2
INT
RD
WR
CS
RESET
CLK/X1
OUT2/X2
82510
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
MXA2
MXA3
MXA4
ASCINT_5
X1
18.432
PDC0..7
CLK40D
Size
B
Document Number
TM_MIX_MOD/SERIAL_I/F
Date:
May 26, 1989
Sheet
12 of
13
REV
A


3
SDOUT
E71
SDOUT
4
U27
DI DO
SK NC
CS NC
93C46
SDIN 3
SCLK 2
CLEECS 1
6
7
E43
E101
E42
SDIN
SCLK
CLEECS
3
3
3,6
Size Document Number REV
B TM_MIX_MOD/EEPROM A
Date: May 26, 1989 Sheet 13 of 13

APPENDIX B

MIX MODULE PAL CODE

```
Title    :IOCTL PAL
Pattern  :U14
Rev      :1.02
Author   :ERIK A STEEB
Company  :INTeL Corporation
Date     :10/2/89

              Monolithic Memories PAL22V10-15

                    **********   **********
                    *          ***          *
              IOCLK ** 1                 24 ** Vcc
                    *                      *
              MXMIO ** 2                 23 ** MXBS16
                    *                      *
               MXDC ** 3                 22 ** MXWAIT
                    *                      *
               MXWR ** 4                 21 ** W0
                    *                      *
             LCLSEL ** 5                 20 ** SWAIT
                    *                      *
              MXCYC ** 6                 19 ** S2
                    *                      *
              MXCMD ** 7                 18 ** S1
                    *                      *
                 NC ** 8                 17 ** S0
                    *                      *
                 A6 ** 9                 16 ** IORD
                    *                      *
                 A5 ** 10                15 ** IOWT
                    *                      *
                 A4 ** 11                14 ** IOCYC
                    *                      *
                Gnd ** 12                13 ** MXRST
                    *                      *
                    ************************
```

```
; created by eas
; (C) - COPYRIGHT INTeL Corporation, 1989
 TITLE IOCTL PAL
 PATTERN U14
 REVISION 1.02
 AUTHOR ERIK A STEEB
 COMPANY INTeL Corporation
 DATE 10/2/89

;Revision 1.02 corrects a problem in the synchronous wait machine
;which brought the machine back to the 0000 state when IOCYC went
;inactive.

CHIP IO_CONTROL PAL22V10

;PINS  1      2      3      4       5       6       7      8    9
      IOCLK MXMIO  MXDC MXWR /LCLSEL /MXCYC /MXCMD  NC  /A6

;PINS 10     11     12     13      14      15      16    17   18
     /A5    /A4     GND /MXRST /IOCYC /IOWT  /IORD  S0   S1

;PINS 19     20     21     22      23    24    25
      S2   SWAIT    W0 /MXWAIT /MXBS16 VCC INIT

STRING IO_CYCLE '/MXRST * /MXMIO * MXDC * LCLSEL * MXCYC'

EQUATIONS

IOCYC   =  IO_CYCLE

IORD    =  IO_CYCLE * /MXWR * MXCMD * SWAIT
        +  IORD  *  MXCMD

IOWT    =  IO_CYCLE *  MXWR * MXCMD * SWAIT
        +  IOWT  *  MXCMD

MXBS16.TRST = IOCYC                  ; IO_CYCLE + 15 ns Tri-State

MXBS16  =  IO_CYCLE *  A6            ; BS16* asserted any access
        +  IO_CYCLE *  A5            ; except to PPI 0.
        +  IO_CYCLE *  A4
```

```
;  *** SYNCHRONOUS PORTION OF WAIT MACHINE ***

INIT.RSTF = MXRST                ; Global reset of register outputs

S0      :=   S0 *  S1
         +   S0 * /S2
         +   S0 *  SWAIT
         +   S1 * /S2 * SWAIT

S1      :=  /MXCMD *  S0 * /S2 * /SWAIT
         +   S0 *  S1 * /S2
         +  /S0 *  S1 *  S2
         +  /S0 *  S2 *  SWAIT
         +   S1 * /S2 *  SWAIT

S2      :=   S0 *  S1
         +   S0 *  S2 * /SWAIT
         +  /S0 * /S1 *  S2 *  SWAIT
         +  /S0 * /S1 *  SWAIT *  MXCMD

SWAIT   :=  /S0 *  SWAIT
         +   S0 *  S1 *  SWAIT
         +   S0 * /S1 *  S2 *  SWAIT
         +   IO_CYCLE *  MXWAIT * /S0 * /S1 * /S2 * /SWAIT

;  *** ASYNCHRONOUS PORTION OF WAIT MACHINE ***

W0       =  IO_CYCLE * /SWAIT * W0
         +  IO_CYCLE *  SWAIT * MXWAIT

MXWAIT.TRST = IOCYC          ;( IO_CYCLE + 15 ns TRI-STATE )

MXWAIT   =  IO_CYCLE *  SWAIT * MXWAIT
         +  IO_CYCLE * /W0

;
;  DESCRIPTION
;
;  The IOCYC variable is used by the tri-state outputs for MXBS16* and

;  MXWAIT* to facilitate the "logical false before tri-state" rule.
;
```

```
SIMULATION

TRACE_ON  /IORD /IOWT /IOCYC /MXCYC /MXCMD S0 S1 S2 SWAIT W0 /MXWAIT
/MXBS16
SETF /IOCLK MXRST /MXCMD /LCLSEL /MXCYC
CLOCKF
CLOCKF
SETF /MXRST /MXMIO MXDC /MXWR    ; set I/O read status
SETF /A6 /A5 /A4                 ; set PPI0 address
CLOCKF
CLOCKF
SETF LCLSEL                      ; select the module
CLOCKF
CLOCKF
SETF MXCYC                       ; start a valid cycle
CLOCKF                           ; provide plenty of clocks
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
SETF MXCMD                       ; issue the command
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
SETF /MXCMD                      ; release the command
CLOCKF
SETF /MXCYC                      ; end transfer cycle
SETF /MXRST /MXMIO MXDC MXWR     ; set I/O write status
SETF /A6 /A5 A4                  ; set PPI1 address
CLOCKF
SETF MXCYC                       ; begin transfer cycle
CLOCKF
SETF MXCMD                       ; issue command
```

```
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
SETF /MXCMD                    ; release command
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
SETF /MXCYC                    ; end cycle
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
SETF /LCLSEL                   ; deselect module
CLOCKF
CLOCKF
TRACE_OFF
```

```
PALASM SIMULATION, V2.23 - MARKET RELEASE (2-1-88)
(C) - COPYRIGHT MONOLITHIC MEMORIES INC, 1988
PALASM SIMULATION SELECTIVE TRACE LISTING

Title     : IOCTL PAL                  Author    : ERIK A STEEB
Pattern   : U14                        Company   : INTeL Corporation
Revision  : 1.02                       Date      : 10/2/89

PAL22V10
Page :  1
          g c cgg  c   cg  c  c g  c  c  c   c  c  c
/IORD     HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOWT     HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOCYC    HHHHHHHHHH HHHHHHHHHH LLLLLLLLLL LLLLLLLLLL
/MXCYC    HHHHHHHHHH HHHHHHHHHH LLLLLLLLLL LLLLLLLLLL
/MXCMD    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
 S0       LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
 S1       LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
 S2       LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
 SWAIT    LLLLLLLLLL LLLLLLLLLL LLHHHHHHHH HHHHHHHHHH
 W0       LLLLLLLLLL LLLLLLLLLL LLHHHHHHHH HHHHHHHHHH
/MXWAIT   HHHHHHHHHH HHHHHHHHHH LLLLLLLLLL LLLLLLLLLL
/MXBS16   HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH

Page :  2
           c  c  c  c  c  c  c g  c  c  c   c  c  c
/IORD     HHHHHHHHHH HHHHHHHHHH LLLLLLLLLL LLLLLLLLLL
/IOWT     HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOCYC    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/MXCYC    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/MXCMD    HHHHHHHHHH HHHHHHHHHH LLLLLLLLLL LLLLLLLLLL
 S0       LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LHHHHHHHHH
 S1       LLLLLLLLLL LLLLLLLLLL LLLLLHHHHH HHHHHHHLLL
 S2       LLLLLLLLLL LLLLLLLLLL LLHHHHHHLL LLLLHHHHHH
 SWAIT    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
 W0       HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/MXWAIT   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/MXBS16   HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH

Page :  3
           c  c  c  c  cg  cg  gg  cg   c g  c  c  c
/IORD     LLLLLLLLLL LLLLHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOWT     HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHLL
/IOCYC    LLLLLLLLLL LLLLLLLLHH HHHHHLLLLL LLLLLLLLLL
/MXCYC    LLLLLLLLLL LLLLLLLLHH HHHHHLLLLL LLLLLLLLLL
/MXCMD    LLLLLLLLLL LLLLHHHHHH HHHHHHHHHH LLLLLLLLLL
 S0       HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHLLLLLLLL
 S1       LLLLLLLLLL LLLLLLHHHH HHHHHHHHLL LLLLLLLLLL
 S2       LLLLLLLLLL LLLLLLLLLL LLLHHHHHHH HHHHHLLLLL
 SWAIT    HHHLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLHH
 W0       HHHHHHHHHH HHHHHHHHLL LLLLLLLLLL LLLLLLLLHH
/MXWAIT   LLLHHHHHHH HHHHHHHHHZ ZZZZZZLLLL LLLLLLLLLL
/MXBS16   HHHHHHHHHH HHHHHHHHHZ ZZZZZZLLLL LLLLLLLLLL
```

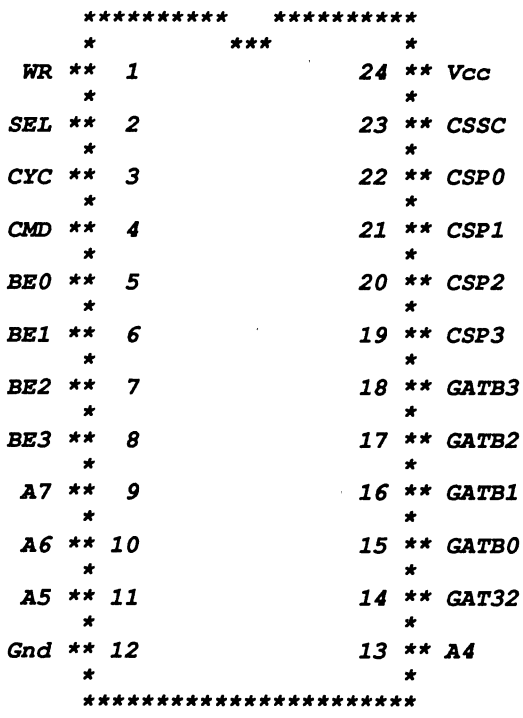
```
Page :  4
          c  c  c   c  c  c   c  c  c  c   c  c  c
/IORD     HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOWT     LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/IOCYC    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/MXCYC    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/MXCMD    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
 S0       LLLLLLLLLL HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
 S1       LLLLHHHHHH HHHHHHLLLL LLLLLLLLLL LLLLLLLLLL
 S2       LHHHHHHLLL LLLHHHHHHL LLLLLLLLLL LLLLLLLLLL
 SWAIT    HHHHHHHHHH HHHHHHHHHH HHLLLLLLLL LLLLLLLLLL
 W0       HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/MXWAIT   LLLLLLLLLL LLLLLLLLLL LLHHHHHHHH HHHHHHHHHH
/MXBS16   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL

Page :  5
          c  c  c   c  c  cg   c  c  c  c   c  c  cg
/IORD     HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOWT     LLLLLLLLLL LLLLLLLLHH HHHHHHHHHH HHHHHHHHHH
/IOCYC    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLH
/MXCYC    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLH
/MXCMD    LLLLLLLLLL LLLLLLLHHH HHHHHHHHHH HHHHHHHHHH
 S0       HHHHHHHHHH HHHHHHHHHH HHHHHHHHLL LLLLLLLLLL
 S1       LLLLLLLLLL LLLLLLLLLH HHHHHLLLLL LLLLLLLLLL
 S2       LLLLLLLLLL LLLLLLLLLL LLHHHHHHHH HLLLLLLLLL
 SWAIT    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
 W0       HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHL
/MXWAIT   HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/MXBS16   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLH

Page :  6
          c  c  c   c  c  c   c  cg  c   c
/IORD     HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HH
/IOWT     HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HH
/IOCYC    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HH
/MXCYC    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HH
/MXCMD    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HH
 S0       LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LL
 S1       LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LL
 S2       LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LL
 SWAIT    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LL
 W0       LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LL
/MXWAIT   ZZZZZZZZZZ ZZZZZZZZZZ ZZZZZZZZZZ ZZ
/MXBS16   ZZZZZZZZZZ ZZZZZZZZZZ ZZZZZZZZZZ ZZ
```

```
Title    :IODCD PAL
Pattern  :U15
Rev      :1.01
Author   :ERIK A STEEB
Company  :INTeL Corporation
Date     :5/24/89
```

Monolithic Memories PAL22V10-15



```
; created by eas
; (C) - COPYRIGHT INTeL Corporation, 1989
 TITLE IODCD PAL
 PATTERN U15
 REVISION 1.01
 AUTHOR ERIK STEEB
 COMPANY INTeL Corporation
 DATE 5/24/89

CHIP IO_DECODE PAL22V10

;PINS   1    2    3    4    5    6    7    8    9   10  11
        WR /SEL /CYC /CMD /BE0 /BE1 /BE2 /BE3  NC /A6 /A5

;PINS  12    13   14     15     16     17
       GND  /A4 /GAT32 /GATB0 /GATB1 /GATB2

;PINS   18     19    20    21    22    23  24   25
      /GATB3 /CSP3 /CSP2 /CSP1 /CSP0 /CSSC VCC INIT

STRING  CYCLE    'SEL * CYC'
STRING  GATE_RD  'SEL * CYC * /WR * CMD'
STRING  GATE_WT  'SEL * CYC *  WR'

EQUATIONS

GATB0  = GATE_RD * /A6 * /A5 * /A4
       + GATE_WT * /A6 * /A5 * /A4
       + GATE_RD *  A6 * /A5
       + GATE_WT *  A6 * /A5

GATB1  = GATE_RD * /A6 * /A5 *  A4 * /BE1 *  BE0
       + GATE_WT * /A6 * /A5 *  A4 * /BE1 *  BE0

GATB2  = GATE_RD * /A6 *  A5 * /A4 * /BE1 *  BE0
       + GATE_WT * /A6 *  A5 * /A4 * /BE1 *  BE0

GATB3  = GATE_RD * /A6 *  A5 *  A4 * /BE1 *  BE0
       + GATE_WT * /A6 *  A5 *  A4 * /BE1 *  BE0

GAT32  = GATE_RD * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
       + GATE_WT * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
       + GATE_RD * /A6 * /A5 * /A4 * /BE3 * /BE2 *  BE1 * BE0
       + GATE_WT * /A6 * /A5 * /A4 * /BE3 * /BE2 *  BE1 * BE0
```

```
CSP0    = CYCLE   * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
        + CYCLE   * /A6 * /A5 * /A4 * /BE3 * /BE2 *  BE1 * BE0
        + CYCLE   * /A6 * /A5 * /A4 * /BE3 * /BE2 * /BE1 * BE0

CSP1    = CYCLE   * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
        + CYCLE   * /A6 * /A5 * /A4 * /BE3 * /BE2 *  BE1 * BE0
        + CYCLE   * /A6 * /A5 *  A4 * /BE3 * /BE2 * /BE1 * BE0

CSP2    = CYCLE   * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
        + CYCLE   * /A6 *  A5 * /A4 * /BE3 * /BE2 * /BE1 * BE0

CSP3    = CYCLE   * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
        + CYCLE   * /A6 *  A5 *  A4 * /BE3 * /BE2 * /BE1 * BE0

CSSC    = CYCLE   *  A6 * /A5 * /BE3 * /BE2 * /BE1 * BE0


 ; DESCRIPTION;:
;
; NAME CHANGES       SCHEMATIC  to  PIN LIST
;
;                      /LCLSEL        /SEL
;                      /MXCYC         /CYC
;                      /MXCMD         /CMD
;                       MXWR           WR
;                      /MXBEx         /BEx
;                      /MXAx          /Ax
;
;  Read and write accesses distinquished in "GAT" equations to meet
;  82C55A-2 data hold times for writes and MIX spec data off times
;  for reads.
;
; I/O DECODE SCHEME AS FOLLOWS:
;
;  Byte  Parallel Port @ offset xxx000xxxx binary (first = 0xH)
;  Byte  Parallel Port @ offset xxx001xxxx binary (first = 1xH)
;  Byte  Parallel Port @ offset xxx010xxxx binary (first = 2xH)
;  Byte  Parallel Port @ offset xxx011xxxx binary (first = 3xH)
;  Word  Parallel Port @ offset xxx000xxxx (Combo of two byte ports)
;  DWord Parallel Port @ offset xxx000xxxx (Combo of four byte ports)
;  Serial Port         @ offset xxx10xxxxx  (first = 4xH & 5xH)
;
;  NOTE: Address lines /MXA9, /MXA8, and /MXA7 not used so decode
;        is mirrored every 128 bytes.
```

```
SIMULATION

TRACE_ON  /SEL /CYC /CMD WR /CSP0 /CSP1 /CSP2 /CSP3 /CSSC
          /GATB0 /GATB1 /GATB2 /GATB3 /GAT32
SETF      SEL
SETF      /WR /A6 /A5 A4                ; Set PPI1 address read
SETF      /BE3 /BE2 BE1 BE0             ; Word access
SETF      CYC                           ; Begin valid cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      /BE1                          ; Byte access
SETF      CYC                           ; Begin cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      WR                            ; Write access
SETF      CYC                           ; Begin cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      /A4                           ; Set PPI0 address
SETF      CYC                           ; Begin cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      BE1                           ; Word access
SETF      CYC                           ; Begin valid cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      /WR BE3 BE2                   ; Dword access read
SETF      CYC                           ; Begin valid cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      A6 A5                         ; Set invalid address read
SETF      CYC                           ; Begin valid cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      /BE3 BE2 BE1 /BE0             ; Set invalid byte enable
SETF      CYC                           ; Begin valid cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
TRACE_OFF
```

```
PALASM SIMULATION, V2.23 - MARKET RELEASE (2-1-88)
(C) - COPYRIGHT MONOLITHIC MEMORIES INC, 1988
PALASM SIMULATION SELECTIVE TRACE LISTING

Title     : IODCD PAL                Author    : ERIK STEEB
Pattern   : U15                      Company   : INTeL Corporation
Revision  : 1.01                     Date      : 5/24/89

PAL22V10
        gggggggggg gggggggggg gggggggggg gggggggggg
/SEL    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/CYC    XXXLLLHHLL LHHLLLHHLL LHHLLLHHLL LHHLLLHHLL
/CMD    XXXXLHHHHL HHHHLHHHHL HHHHLHHHHL HHHHLHHHHL
 WR     XLLLLLLLLL LLHHHHHHHH HHHHHHHLLL LLLLLLLLLL
/CSP0   XHHHHHHHHH HHHHHHHHLL LHHLLLHHLL LHHHHHHHHH
/CSP1   XXHHHHHHLL LHHLLLHHHH HHHLLLHHLL LHHHHHHHHH
/CSP2   XHHHHHHHHH HHHHHHHHHH HHHHHHHHLL LHHHHHHHHH
/CSP3   XHHHHHHHHH HHHHHHHHHH HHHHHHHHLL LHHHHHHHHH
/CSSC   XHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/GATB0  XHHHHHHHHH HHHHHHHHLL LHHLLLHHHL HHHHHHHHHH
/GATB1  XXHHHHHHHL HHHLLLHHHH HHHHHHHHHH HHHHHHHHHH
/GATB2  XHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/GATB3  XHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/GAT32  XHHHHHHHHH HHHHHHHHHH HHHLLLHHHL HHHHHHHHHH

Page :  2
        gg
/SEL    LL
/CYC    LH
/CMD    HH
 WR     LL
/CSP0   HH
/CSP1   HH
/CSP2   HH
/CSP3   HH
/CSSC   HH
/GATB0  HH
/GATB1  HH
/GATB2  HH
/GATB3  HH
/GAT32  HH
```

ENHANCING SYSTEM PERFORMANCE WITH THE MULTIBUS® II ARCHITECTURE

Although the MULTIBUS® II architecture can accommodate systems with a wide range of performance, systems that take advantage of its multiprocessing capabilities can achieve new performance levels while maintaining reasonable price/performance ratios. Today, multiprocessing provides an easy path to increased functionality and processing power largely because of the availability of inexpensive memory and CPUs.

The low cost of high-performance microprocessors and RAM chips has drastically altered the cost dynamics of systems design. The material cost of a CPU and its memory are typically a small portion of the total system cost, in sharp contrast to mini and mainframe computers where the cost of the CPU and memory is the majority of system cost. The decreased cost factor means today's designer can optimize a system's price/performance by dedicating a CPU to each function in the system.

This product brief will discuss the MULTIBUS II multiprocessing capabilities and their user benefits. The capabilities include:

- A high-speed local environment
- An efficient burst transfer capability
- A hardware-based message passing facility

Higher Performance Through Multiprocessing

The key to high performance in multiprocessing systems is allowing all of the processors to run concurrently in their own private environments. For this to occur, each functional module must contain its own CPU, memory and I/O resources. It also means that the system bus is primarily used for passing commands and data between modules.

A system using this approach might consist of a host processing board and intelligent disk controller, a terminal concentrator and LAN controller boards (Figure 1). Each

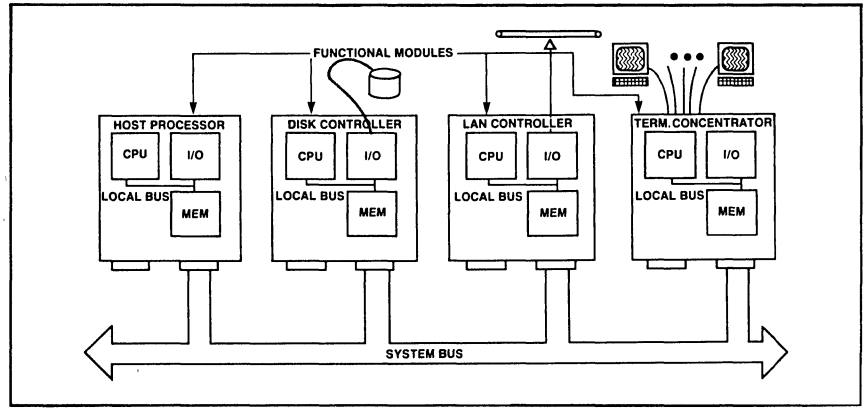


Figure 1. Functional Partitioning is the Distribution of CPU, Memory & I/O Resources to Support Different Functions in a System

functional module would contain the resources required to perform its assigned function. Further, each module would operate over its own private local bus which is decoupled from the system bus. This enables the modules to operate concurrently with each other and leaves the system bus open for communication between the intelligent modules.

High-Speed Local Environment Optimizes On-Board Resources

In multiprocessing systems, performance is optimized when all execution code and data is accessed in a local environment. The most important performance factors in a local environment are the CPU clock speed, the number of CPU clocks per instruction, the CPU instruction set, and the number of memory wait states. While the CPU choice dictates the CPU performance factors, the bus architecture can assist in providing a good CPU-memory and I/O environment.

The MULTIBUS II architecture provides a high-speed local environment through its moderate size board form factor and a local memory bus extension. The MULTIBUS II board form factor is the Eurocard Standard 233mm by 220mm (9.1"×9.0"), chosen because it allows most functional modules to completely fit on one board. This factor is critical to system performance because on-board resources can be optimized to run at their full potential without impacting the system bus. A smaller board size would force a particular function onto multiple boards with a resulting decrease in performance.

Burst Transfers

A key development to optimizing the iPSB bus for multiprocessor communications is the high-speed burst transfer capability. Since address information is transferred over the bus only once for the entire burst, performance is greatly enhanced.

The synchronous handshake capabilities of the iPSB bus nearly double the speed of burst transfers compared to traditional asynchronous handshakes (Figure 2). Burst

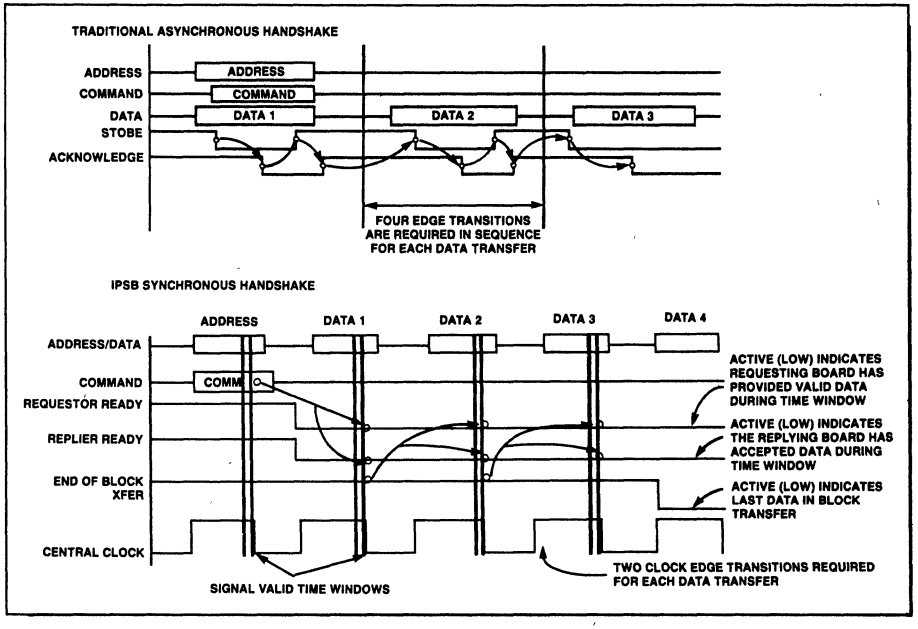


Figure 2. iPSB Synchronous Handshake Compared to Asynchronous Handshake

transfers allow boards to transfer blocks of data over the iPSB bus at speeds up to 40 Mbytes/s. This speed approaches the limit of what can be expected from TTL technology when propagation across a 20-slot backplane is required.

In the iPSB bus, a burst transfer consists of one address clock followed by multiple data transfers. The receiving board takes care of actual memory location placement (ie., auto-increments the memory address, as necessary). The actual speed of the burst transfer will depend on the abilities of the communicating boards. For example, burst transfers from an intelligent board to dual-port memory will typically be only marginally faster than single-cycle writes, due to the long access times from the system bus side of dual-port memory boards.

To achieve the true performance benefits of burst transfers, each board needs the ability to send and receive small bursts at the full bandwidth of the system bus. This can be accomplished by bus interface logic containing high-speed buffers and the ability to format and send 32-bit-wide data bursts.

In the MULTIBUS II architecture, the interface bus logic to the iPSB is defined with burst capability in a message-passing scheme. This ensures that boards developed by various manufacturers will all be able to communicate compatibly at tremendous speeds.

Message passing, as defined in the MULTIBUS II protocol, allows modules to communicate directly. In other words, one module sends a message (data) over the iPSB bus to the address of another module. This differs from the normal CPU functions of reading or writing only from memory or I/O.

Since conventional CPUs do not contain facilities to perform direct CPU-to-CPU communication, additional hardware logic is required. The hardware can be thought of as a coprocessor to the primary CPU, e.g., a coprocessor that adds the function of direct module-to-module communication at speeds many times that which the primary CPU could perform. The coprocessor logic for message passing resides in the bus interface.

An example best illustrates how message passing works (Figure 3). Assume Board A wants to send 1 Kbyte of data to Board B. First, the CPU on Board A would instruct its message passing unit to send 1 Kbyte of data (with the assistance of a DMA device), beginning at a particular location in local memory, to Board B. Next, the message passing coprocessor on Board A takes over so the CPU

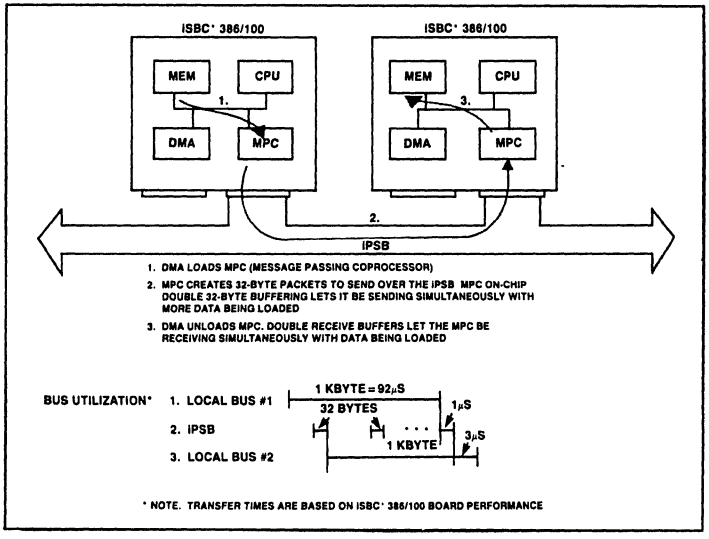


Figure 3. A Message Passing Example

can perform other processing. At this point, the DMA device loads the data into the message passing coprocessor on Board A. Once enough data has been loaded (typically 32 bytes), the coprocessor arbitrates for the bus and sends the first packet of data as a burst transfer to the message-passing logic on Board B.

While the message passing logic on Board B is unloading the first packet out of its high-speed buffers into local RAM, the message-passing logic on Board A is reading the next piece of data into its high-speed buffers. Meanwhile, the system bus is free of traffic and available for another pair of boards to communicate over.

The message-passing logic on Board A continues to build and send small packets of data to Board B's message-passing logic, and Board B continues to unload this data into its local memory until the entire 1 Kbyte has been transferred. At the completion of the transfer, the message-passing logic on both boards interrupts their respective CPUs to notify them that the transfer is complete.

Summary

Five important performance benefits result from the MULTIBUS II multiprocessing capabilities and specifically from hardware-assisted message passing. First, all single-cycle memory/IO transfers can be designed to occur in local CPU environments. These environments are optimized for single-cycle transfers over their local memory buses and usually run at few or no wait states, compared to substantial wait state delays over a system bus.

Second, transfers over the iPSB bus can be done as burst transfers between message-passing logic containing high-speed buffers, thereby transferring data at the maximum bus data rate. Third, the iPSB bus is not in use between data packets and is available for other traffic. Fourth, each CPU does not need direct access into the other board's local environment. That is, no dual port memory (which is slower than single port memory) is required. And fifth, each CPU is available to process other tasks while the data transfer is occurring.

INCREASING SYSTEM RELIABILITY WITH THE MULTIBUS® II BUS ARCHITECTURE

System reliability is more than just mechanical factors like Eurocard and DIN connectors. It involves many design factors often overlooked in traditional buses. The MULTIBUS®II bus architecture addresses the problem of system reliability not only from a mechanical point of view, but from protocol and electrical factors as well. This product brief will discuss how the following MULTIBUS II features resolve specific reliability problems while enhancing overall system reliability:

- Synchronous Timing
- Bus Parity
- Protocol Error Handling
- Bus Timeout
- Power Sequencing
- Eurocard/DIN Connectors
- Front Panel Design
- Backplane Design

INCREASING ELECTRICAL RELIABILITY

Synchronous Timing for Enhanced Noise Immunity

Traditional buses, such as MULTIBUS I and VME, are based on asynchronous timing where the edges or transitions of the bus-control signals cause the bus to perform its functions. Unfortunately, edge-sensitive timing is susceptible to external disturbances and noise. If noise causes a signal to look as though it made a transition, the transition is misinterpreted and a failure results.

The MULTIBUS II architecture addresses this problem by using synchronous sampling of all signal lines. Both the MULTIBUS II Parallel System Bus (iPSB) and the Local Bus Extension (iLBX™ II bus) employ synchronous sampling for enhanced noise immunity. The iPSB serves as a good example of the benefits of synchronous sampling.

In the iPSB bus, all signals (address, data, control, and arbitration) are driven and sampled with respect to a 10 MHz bus clock. The 10 MHz clock breaks the bus activity into 100ns increments with signals sampled at the end of each period. This method avoids looking at the signal while transitions caused by reflections and crosstalk are occurring. Therefore, signals are vulnerable only during the small sampling window.

Figure 1 shows the iPSB timing with the 100ns period divided into three intervals: driver timing, bus propagation, and receiver timing. The 40ns driver timing interval takes into account driver logic delays and the capacitive loading for a maximum of 20 loads spaced over 16.8 inches.

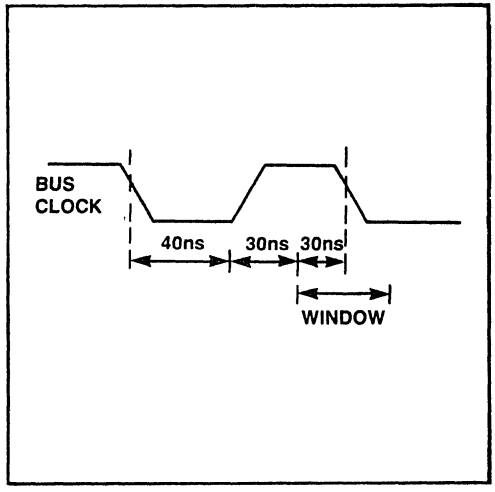


Figure 1. iPSB Timing, Showing Synchronous Sample Driving Stable Data Window

The bus propagation interval accounts for 25ns of signal transit time and 5ns of potential clock skew. A signal traveling on the backplane creates reflections on itself and crosstalk on other signals. The signal transit time allows the signal to propagate down and back on the backplane. It also allows time for crosstalk to subside. This guarantees that the signals have stabilized in spite of distance and interference from other signals.

The receiver interval consists of a 30ns receiver setup time plus 5ns of hold time which extends into the next cycle. This interval is the time the signal is stable prior to sampling on the falling edge of the clock.

Thus, the MULTIBUS II parallel bus timing creates a 65ns interval (driver timing plus bus propagation) when the bus is completely immune to noise or external disturbances. That means during 65% of the time interval, noise causing a transition or level change is simply ignored. It is only during the 35ns receiver setup and hold interval that the bus timing is vulnerable to noise. During this interval, however, the bus contains parity protection (to be discussed in another section).

Comparable Performance at Higher Speeds

A common complaint about synchronous buses is that fixed time increments limit performance compared to asynchronous buses. This may be true at slower bus clock speeds. However, at 10 MHz the differences diminish. If both an asynchronous and a synchronous bus use similar TTL technology for the bus drivers and receivers over the same backplane length, they possess roughly the same bus timing. In other words, the driver timing, bus propagation, and receiver intervals of both buses will be approximately the same with nearly equal performance. However, as we've seen, a synchronous bus offers a significant improvement in system reliability that easily justifies its use.

Guaranteed Electrical Compatibility

Synchronous sampling also has a less obvious benefit — guaranteed electrical compatibility among boards. The 100ns timing of the iPSB is based upon a worst-case environment of 20 boards over a backplane length of 16.8 inches (0.8 inch separation). All derating for loading, voltage margin, and skew is included. Thus, any number of boards, up to 20, are guaranteed to work together.

Electrical compatibility is much harder to achieve in asynchronous buses. Because they are edge-sensitive, asynchronous boards are naturally susceptible to changes in signal edge rates and timing. When the number of boards in a system change, edge rates and timing also change, in some cases adversely affecting system reliability.

The synchronous nature of the bus moves the point of synchronization to the local bus of each board. When two asynchronous CPUs communicate, synchronization between them occurs between each CPU and its interface. This provides a better electrical environment for dealing with reliability problems caused by metastability.

Bus Parity Versus Memory Parity

At this point, it is important to distinguish between BUS parity and MEMORY parity. (See Figure 2.) Both allow the detection of errors. Memory parity protects *data* while it is resident on a memory board. Bus parity, on the other hand, protects *address, control,* and *data* while in transit on the bus. In a sense, one complements the other in reliable systems. In both cases, it is possible to handle errors via retry or other mechanisms.

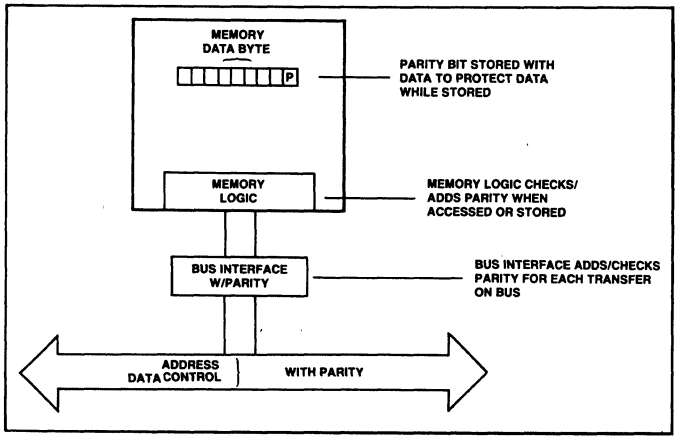


Figure 2. Parity Protects Address Data and Control from Errors which could be Incurred on the iPSB Bus

Bus parity in the MULTIBUS II architecture provides another level of electrical reliability by protecting the bus from noise and external disturbances during the receiver timing interval. It also protects the bus from failed interface components.

On the iPSB bus, the board driving the bus generates bus parity. Address and data lines use byte parity, while control lines use nibble (4-bit) parity. All receiving boards check parity during the receiver timing sampling interval. If an error is detected, the BUS ERROR line is activated. This stops activity on the bus and puts the bus into a predefined known state.

At this point, the system designer has a number of options: retry the transfer, swap in a hot spare, log the error, ignore it, or shut down the system gracefully. Which option he chooses depends on his specific system requirements. Basically, the protocol gives him the opportunity to evaluate the situation and take appropriate action.

PROTOCOL RELIABILITY

Board-to-Board Error Indications

Not all errors occur because of noise or component failure. Sometimes they occur when one board asks another to do something it is not capable of doing. Although traditional buses typically ignore these kinds of errors, they can cause system failure just as noise can. The MULTIBUS II architecture offers a solution.

In the iPSB bus protocol, when one board cannot perform the request, it simply informs the requesting board and allows it to attempt a retry. Five types of error indications are supported: data, transfer width, continuation, not-understood, and negative acknowledge.

A *data error* indicates that the replying board has detected an error with the requested data, for example a memory parity error. *Data transfer errors* occur when the replying board does not support the requested data width. For example, the requesting board might ask for a 32-bit transfer from an 8-bit device. After the replying board indicates the error has occurred, the requesting board can retry the transfer with an 8-bit width.

Although the iPSB bus protocol allows for burst transfers (multiple data cycles following one address cycle), not all boards need to support this capability. If a requesting board attempts a burst transfer with a board which does not support bursts, the replying board will return a *continuation error*. The requesting board can recover by simply retrying with the necessary address cycles.

Trying to write to a read-only memory board is a good example of a *transfer-not-understood error*. This type of error occurs when the replying board does not support the requested operation. As with other board-to-board errors, the requesting board many retry with another request.

The last kind of error, called a *negative acknowledge error,* occurs during a message transfer when resources are not available in the receiving board. This is used for flow control in the MULTIBUS II message passing protocol, a queue-based data movement protocol. Negative acknowledge errors instruct the requesting board to retry the operation at a later time, giving the replying board time to process the data in its queue.

Bus Timeout

Another protocol reliability feature in the MULTIBUS II architecture is the BUS TIMEOUT monitor in the Central Services Module (CSM). If a bus transfer fails to complete within a specified time (e.g., a failed board), the CSM, which monitors all bus activity, activates the BUS TIMEOUT line. This stops all bus activity and places the bus in a predefined known state for recovery. At this point, the error is logged and normal bus activity can resume. As an added feature, designers may define their own timeout error handling policy.

POWER SEQUENCING

The iPSB bus protocol also contains a mechanism for orderly handling of power-up and power-down sequencing. For normal power on/off and unexpected power failures, timing of the RESET, DCLOW, and PROTect signals coordinate the sequencing. The combination of the RESET and DCLOW lines signal whether the power-up operation is a warm or cold start of the system.

Once the system is running, the DCLOW signal (driven by the CSM) is used to indicate imminent loss of DC power (Figure 3). At this time, the system has a predetermined time to save state information. After that interval,

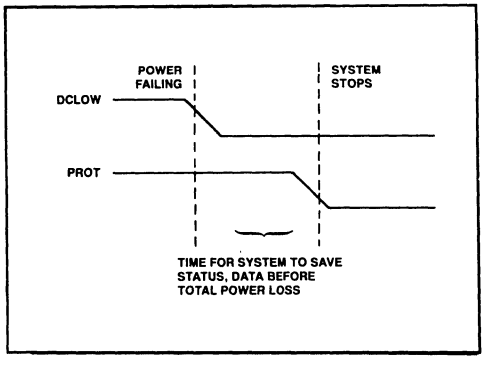


Figure 3. Power Failure Control Lines

the CSM activates the PROTect line which prevents transitions on bus lines from affecting the system during power loss.

MECHANICAL RELIABILITY

The MULTIBUS II mechanical specification is based upon the Eurocard form factor and DIN connectors. However, unlike traditional bus architectures, it goes beyond these mechanical standards with a front panel design that helps the system designer solve EMI (Electro-Magnetic Interference) and ESD (Electro-Static Discharge) problems.

Eurocard and DIN Connectors

The Eurocard family of mechanical specifications is noted for its high reliability in rugged and industrial environments. The MULTIBUS II specification calls out the two-connector 233mm by 220mm and single-connector 100mm by 220mm size boards. The two connector board contains almost the same board area as the 6.75 by 12 inch MULTIBUS I board. That is, it is large enough to allow the implementation of single-board computers with I/O, CPU, and memory onboard, even for 32-bit CPUs.

The DIN 41612 (also known as IEC 603.2) connectors are 96-pin two-piece connectors where each pin consists of a blade mating with two contact points on each side of the blade. This connector approach offers advantages over the board-edge style connectors. Among them are tighter dimensional tolerances, reduced sensitivity to vibration, improved protection from environmental contaminants, and a larger number of cycles for insertion and removal.

FRONT PANEL SYSTEM

The MULTIBUS II front panel system (Figure 4), while dimensionally compatible with standard Eurocard front panels, offers several important advantages.

(Note that while this front panel technology is different from normal Eurocard practice, the dimensioning is such that MULTIBUS II boards fit in any standard Eurocard packaging)

Standard Eurocard front panels make it difficult to comply with EMI and ESD regulations without the use of additional shielding. Adjacent front panels form small, narrow slits between boards which function like a slot antenna at some frequencies. Through these narrow slits, EMI can enter or exit the system and additional shielding is usually required.

To solve this problem, the MULTIBUS II front panel is U-shaped. From an EMI point-of-view, this makes the front panel electrically thicker. While the size of the slit between adjacent boards is the same as the standard Eurocard front panel, the electrically thicker front panel attenuates EMI which satisfies FCC EMI regulations and protects the system from external EMI.

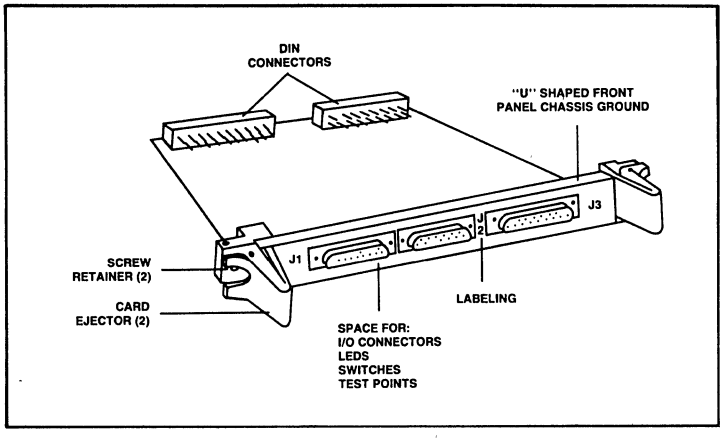


Figure 4. MULTIBUS® II Front Panel System

The U-shaped front panel also adds structural rigidity to the board and has captive retaining screws for securing the board to the system. Shielded I/O connectors located through the front panel eliminate the need for intermediary cables and connectors. In addition, the front panel is at chassis ground for protection against static discharge.

BACKPLANE DESIGN

Designed for reliability, the iPSB bus backplane consists of six layers — three signal layers sandwiched between three power and ground planes (Figure 5). The power and ground planes provide for good power distribution. Moreover, since they are in between each signal layer, they reduce the opportunity for crosstalk due to coupling between the signal layers.

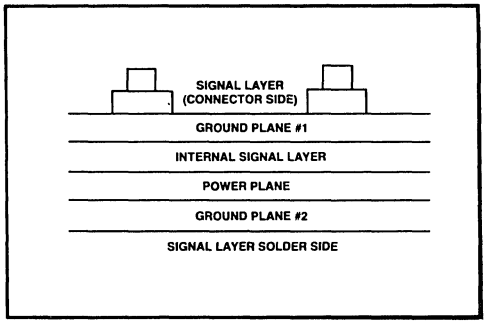


Figure 5. Backplane Design

On each signal layer, signal lines are laid out identically to minimize signal skew across the backplane. To control reflections, each signal line is passively terminated.

Both power and ground connections are evenly distributed across the connectors with 9 pins allocated for +5 volts and 15 for ground providing ample current and good ground return paths.

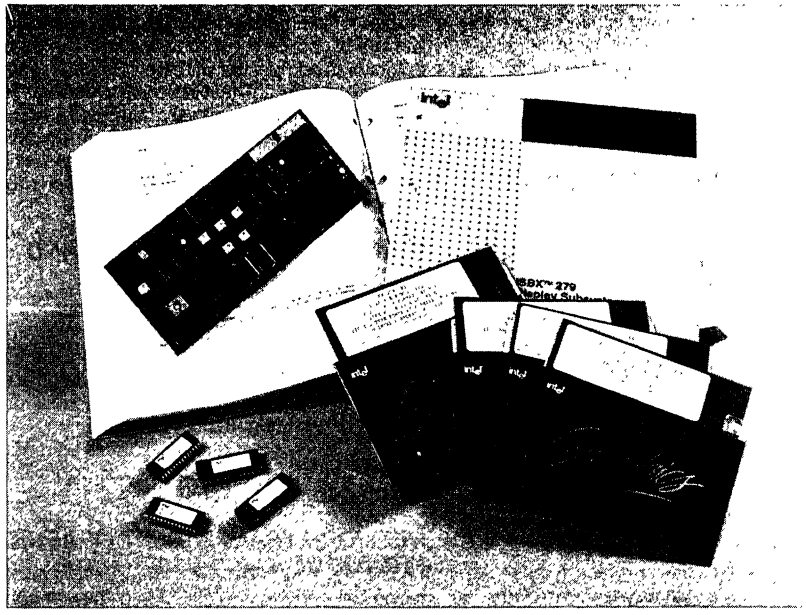
SUMMARY

Because the MULTIBUS II architecture addresses the problems of electrical, protocol and mechanical reliability, it is superior to traditional buses in achieving overall system reliability. Besides the mechanical reliability of its Eurocard form factor, DIN connectors, and backplane design, the MULTIBUS II electrical protocol is highly immune to noise and external disturbances because of its synchronous sampling and bus parity. In addition, the agent error capability catches common operational errors. Other operational concerns such as bus time-out and power sequencing are fully specified.

iSBX™ Expansion Modules

6

iSBX™ 279* DISPLAY SUBSYSTEM



HIGH-SPEED GRAPHICS/WINDOWING FOR iRMX® II SYSTEMS

The iSBX® 279 is a complete graphics subsystem designed to provide users of Intel's iRMX II real-time systems with advanced interactive graphics functions. Based on Intel's 82786 Display Processor, the iSBX 279 efficiently off-loads bitmap and window manipulation from the application CPU, preserving real-time system performance. High-speed windowing, ASCII terminal emulation with system console support, and powerful drawing commands are provided in a convenient system expansion package. Complete software support, including iRMX II Device Driver, Application Interface Libraries for C-286 and PL/M 286, and loadable fonts, provides a high-level, network transparent interface, allowing application portability across Intel's real-time platforms and shortening application development time.

FEATURES:

- Intel 82786 Display Processor
- iRMX II Operating System Device Driver
- High-Level Language Interface
- System Console Support Kits
- Windowed User Interface
- Terminal Emulator
- Standard Graphics Command Interpreter
- Network Transparent Graphics Protocol

intel

September, 1989
Order Number 280667-001

FEATURES

INTEL 82786 DISPLAY PROCESSOR

The Intel 82786 Display Processor is used to provide nearly instantaneous window manipulation. Together with the iSBX 279 firmware and iRMX II software, this allows multiple overlapping windows displaying graphical information or terminal sessions to be presented simultaneously.

iRMX II OPERATING SYSTEM DEVICE DRIVER

The iRMX II Interactive Configuration Utility provides the screens needed to configure the iSBX 279. The device driver is provided as part of the iSBX 279 upgrade kit. The device driver manages the device interface and performs I/O on behalf of application requests through device/file connections in the iRMX II IOS. The device driver is compatible with iRMX II Terminal Support. This speeds development, by allowing the programmer to remain unaware of the device interface, instead concentrating on the application code needed for the target system.

HIGH-LEVEL LANGUAGE INTERFACE

Application interface libraries are provided for C-286 and PL/M 286. The application interface is a rich set of graphics and windowing primitives that provide standard drawing functions with complete control of bitmaps and windows. This allows the application programmer to quickly begin writing sophisticated real-time graphics applications using a portable interface for iRMX II systems.

SYSTEM CONSOLE SUPPORT KITS

The iSBX 279 is designed to be the system console for iRMX II MULTIBUS® systems based on iSBC® 286/1X and iSBC 386/XX CPU's. EPROM's containing System Confidence Tests, System Debug Monitor, and Bootstrap Loader are provided to allow the iSBX 279 to operate as a system console.

WINDOWED USER INTERFACE

The user may interactively MOVE, RESIZE, PUSH, POP, and SELECT windows using the mouse and an easy-to-use menu provided for this purpose. Several terminal sessions and interactive graphics applications can be managed from a single console. Window and bitmap manipulation is performed locally by the iSBX 279, allowing complex user-interface operations to proceed in parallel with time-critical real-time tasks on the host-CPU.

TERMINAL EMULATOR

The terminal emulator allows existing applications to run in a window without modification. The terminal emulator is compatible with the iRMX II Human Interface, the AEDIT text editor, iRMX Virtual Terminal software, and other terminal oriented programs that can be configured to operate with a smart CRT. By using the virtual terminal capability, it is possible to access any host on an OpenNET™ network from a single display.

STANDARD GRAPHICS COMMAND INTERPRETER

The graphics command interpreter is an implementation of the Computer Graphics Interface (CGI), providing an interface that is consistent with current ISO-CGM and ANSI-CGI standardization efforts, while extending this interface to include window and bitmap manipulation functions. The interface is fully compatible with Intel's existing iVDI 720 R1.8 interface providing a direct upgrade for iSBC 186/78A applications.

NETWORK TRANSPARENT GRAPHICS PROTOCOL

Using OpenNET it is possible to display images stored on remote nodes, run interactive applications from any node on the network, allow multiple applications on several processors to share a single display, and access other displays from a local application processor. Network transparency allows distributed applications to be controlled from a single console.

INTEL QUALITY AND RELIABILITY

The components of the iSBX 279 subsystem are designed and manufactured in accordance with Intel's high quality standards. Quality is assured through rigorous testing in our state-of-the-art Environmental Testing Laboratory.

WORLDWIDE SERVICE AND SUPPORT

Intel provides support for repair or on-site service. Development support options include phone support, subscription service, on-site consulting, and customer training.

SPECIFICATIONS

Display

640 × 480 Pixels
25MHz Clock
60Hz Noninterlaced Frame Rate
31.5kHz Horizontal Rate
Up to 256 Colors from Palette of 16.7 Million[1]

Memory

1 Megabyte Bitmapped Memory
Bitmap Depths of 1,2,4, or 8 Bits

Interfaces Supported
IBM[2] PC/AT[3] Keyboard (or compatible)
PC-MOUSE (or compatible) locator device
NEC MultiSync[4] Analog-RGB Monitor (or compatible)
8-bit SBX Bus Interface (all signals TTL compatible)

Physical Characteristics			
Width:	7.5 in	= 19.05 cm.	
Length:	3 15 in	= 8 00 cm	
Height:	0.80 in.	= 2.03 cm.	iSBX 279 Only
	1.14 in.	= 2.89 cm.	With iSBC Host
Weight:	6.0 oz.	= 170.1 gm.	iSBX 279 Only

Electrical Requirements
+5 VDC ±5% @ ≈ 3.0 A
+12 VCD ±5% @ ≈ 0.025 A
−12 VDC ±5% @ ≈ 0.025 A

Environmental Requirements	
Operating Temperature:	0°C to 55°C @ 200 LFPM miminum air flow
Storage Temperature:	−40°C to +75°C
Humidity:	0% to 95%, non-condensing

ORDERING INFORMATION

Order Code	*Description*
SBX279	Display Subsystem (Board Only)
SXM2791286K	Complete iSBC 286/1X System Upgrade (Includes SBX279)
SXM2791386K	Complete iSBC 386/XX System Upgrade (Includes SBX 279)

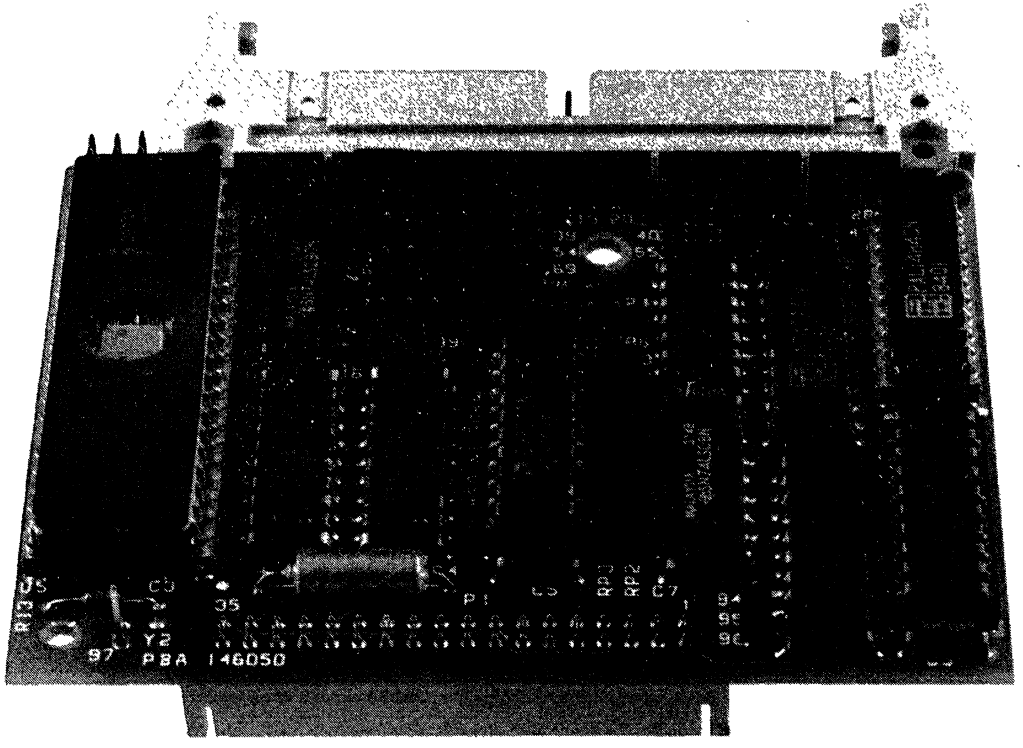
Note: iRMX II must be purchased separately. Kits include all software, firmware, and hardware needed to begin using the iSBX 279 Display Subsystem immediately. Software Royalty included in each iSBX 279.

[1]See the RGI279 Product Release Notes for restrictions on window size/placement when displaying 256 colors
[2]IBM is a registered trademark of International Business Machines, Inc
[3]PC/AT is a trademark of International Business Machines, Inc
[4]MultiSync is a trademark of NEC

iSBX™ 217C* ¼-INCH TAPE DRIVE INTERFACE MULTIMODULE™ BOARD

- **iSBX™ Bus Module Provides Tape Backup Capability for iSBC® 215 Generic Winchester Controller**
- **Configurable to Interface with up to Four QIC-02 Compatible or 3M HCD-75 Compatible Tape Drives**
- **Implements the QIC-02 with Parity Streaming Tape Interface Standard**
- **Supports Transfer Rates of 90K, 30K or 17K Bytes per Second Depending on Tape Speed**
- **Supported by iRMX™ 86 and XENIX** Operating Systems when Used on iSBC® 215 Generic Winchester Controller Board**
- **+5 Volt Only Operation**

The iSBX 217C ¼-Inch Tape Drive Interface module is a member of Intel's family of IEEE 959 iSBX I/O Expansion Bus products. This module is particularly useful for implementing cartridge tape back-up capability directly on the iSBC 215 Generic Winchester Disk Controller via DMA. The iSBX 217C bus board can also provide a low-cost tape storage interface for any Intel single board computer, having an iSBX bus connector, via programmed I/O. The iSBX 217C module interfaces with up to four streaming tape drives. Typically, these drives provide 20 to 45 megabytes of storage each. When used in conjunction with these drives and the iSBC 215 board, the module can transfer 20 megabytes of data from disk to tape in about fourteen minutes. Alternatively, the iSBX 217C board can interface with up to four 3M Company HCD-75 compatible start/stop tape drives, for those applications requiring access to individual data files on tape.



210817-1

*The iSBX™ 217C is also manufactured under product code piSBX™ 217C by Intel Puerto Rico, Inc.
**XENIX™ is a trademark of Microsoft Corporation.

September 1989
Order Number: 210817-003

SPECIFICATIONS

Compatibility

Host—Any iSBC signal board computer or peripheral controller with an iSBX connector. The iSBC 215 Generic Winchester Controller includes on-board firmware to support the iSBX 217C under either the iRMX 86 or XENIX Operating Systems. The firmware on the iSBC 215A and iSBX 215B Winchester Controllers cannot support the iSBX 217C module.

Drives—Any QIC-02 or 3M HCD-75 interface compatible cartridge ¼-inch magnetic tape drive.

Transfer Rate

90K (one byte every 11 microseconds), 30K (one byte every 33 microseconds) or 17K (one byte every 53 microseconds) depending on tape drive speed.

Equipment Supplied

iSBX 217C Interface Module
Reference Schematic

Controller-to-drive cabling and connectors are not supplied. Cables can be fabricated with flat cable and commercially-available connectors as described in the Hardware Reference Manual.

Nylon mounting bolts

Physical Characteristics

Width: 3.08 inches (7.82 cm)
Height: 0.809 inches (2.05 cm)
Length: 3.70 inches (9.40 cm)
Shipping Weight: 3.5 ounces (99.2 gm)
Mounting: Occupies one single-wide iSBC MULTIMODULE position on boards

Electrical Characteristics

Power Requirements: +5 VDC @ 1.5A

Environmental Characteristics

Temperature: 0°C to +55°C (operating) @200 LFM; −55°C to +85°C (non operating)

Humidity: Up to 90% relative humidity without condensation (operating); all conditions without condensation or frost (non-operating)

Reference Manual

D146704— iSBX 217C Board Hardware Reference Manual (NOT SUPPLIED)

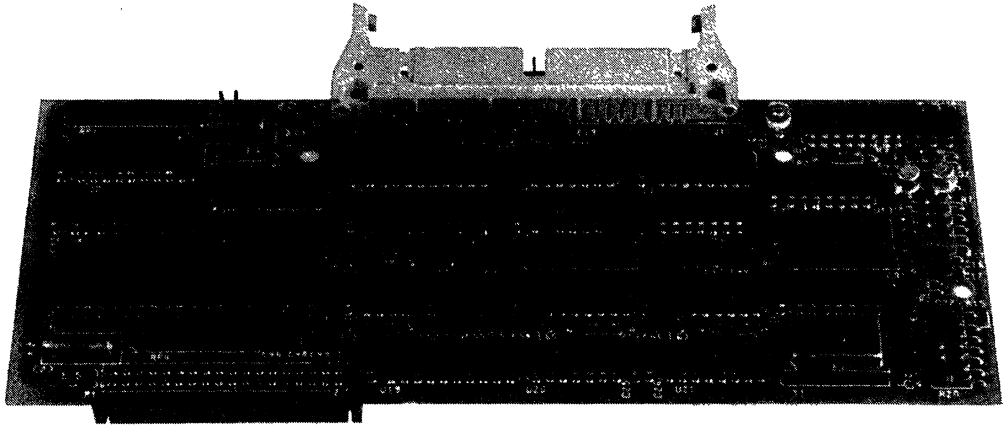
ORDERING INFORMATION

Order Code	Description
SBX217C	Cartridge ¼-inch Tape Drive Interface

iSBX™ 218A*
FLEXIBLE DISK CONTROLLER

- **IEEE 959 iSBX™ Bus Compatible 8" or 5.25" Floppy Diskette Controller Module**
- **Hardware and Software Compatible with iSBX 218 Module**
- **Controls Most Single/Double Density and Single/Double Sided Floppy Drives**
- **User Programmable Drive Parameters Allow Wide Choice of Drives**
- **Motor On/Off Latch Under Program Control**
- **Drive-Ready Timeout Circuit for 5.25 Inch Floppy Drives**
- **Phase Lock Loop Data Separator Assures Data Integrity**
- **Read and Write on Single or Multiple Sectors**
- **Single +5 Volt Supply Required**

The Intel iSBX™ 218A Flexible Disk Controller module is a software and hardware compatible replacement for the iSBX 218 module and provides additional features. The iSBX 218A module is a double-wide iSBX module floppy disk controller capable of supporting virtually any soft-sectored, single/double density and single/double sided floppy drives. The controller can control up to four drives. In addition to the standard IBM 3740 and IBM system 34 formats, the controller supports sector lengths up to 8192 bytes. The iSBX 218A module's wide range of drive compatibility is achieved without compromising performance. The operating characteristics are specified under user control. The controller can read and write either single or multiple sectors.

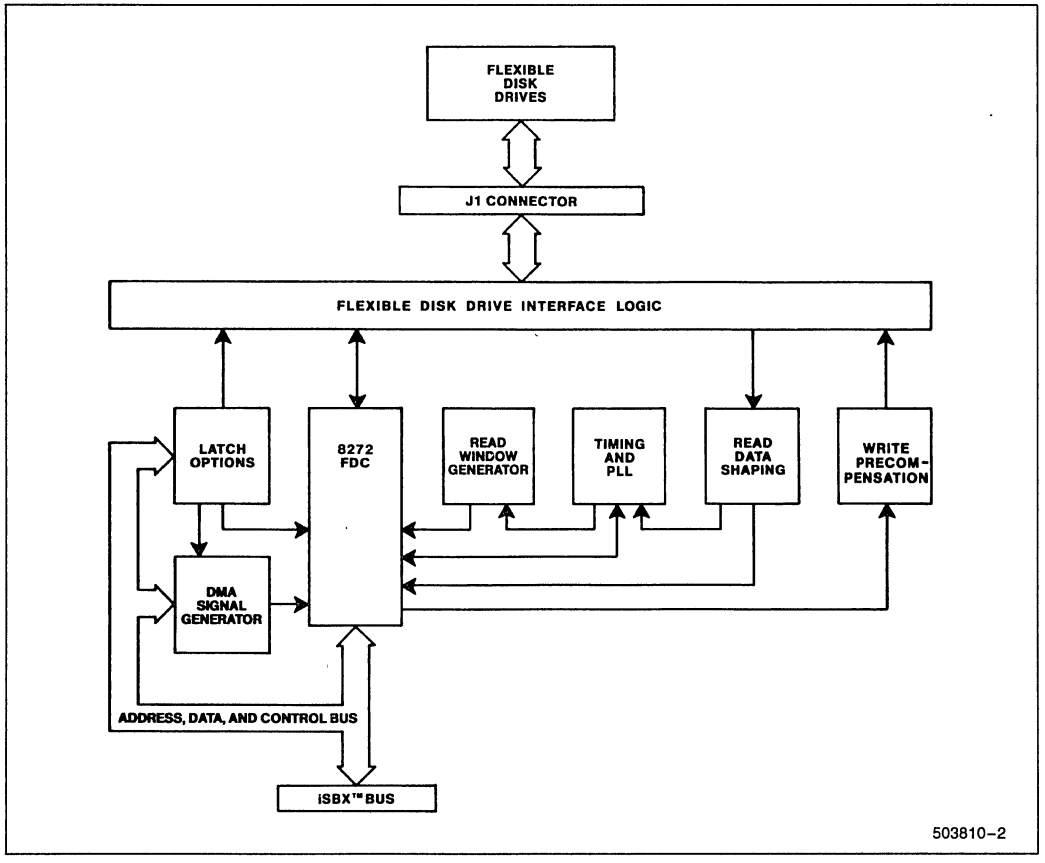


503810-1

*The iSBX™ 218A is also manufactured under product code piSBX™ 218A by Intel Puerto Rico, Inc.

October 1989
Order Number: 503810-003

BLOCK DIAGRAM



Block Diagram of iSBX™ 218A Board

SPECIFICATIONS

Compatibility

CPU—Any single board computer or I/O board implementing the iSBX bus interface and connector.

Devices—Double or single density standard (8") and mini (5¼") flexible disk drives. The drives may be single or double sided. Drives known to be compatible are indicated in the table to the right.

Standard (8")		Mini (5¼")	
Caldisk	**143M**	**Shugart**	**450/400**
Remex	**RFD 4000**	**Shugart**	**460/410**
Memorex	**550**	**Micropolis**	**1015-IV**
MFE	**700**	**Pertec**	**250**
Siemens	**FDD 200-8**	**Siemens**	**200-5**
Shugart	**SA 850/800**	**Tandon**	**TM-100**
Shugart	**SA 860/810**	**CDC**	**9409**
Pertec	**FD650**	**MPI**	**51/52/91/92**
CDC	**9406-3**		

Diskette—Unformatted IBM Diskette 1 (or equivalent single-sided media); unformatted IBM Diskette 2D (or equivalent double-sided).

Equipment Supplied

iSBX 218A Controller

Reference Schematic

Controller-to-drive cabling and connectors are not supplied with the controller. Cables can be fabricated with flat cable and commercially-available connectors as described in the iSBX 218A Hardware Reference Manual.

Nylon Mounting Screws and Spacers

Physical Characteristics

Width: 3.15 inches (8.0 cm)

Height: 0.83 inches (2.1 cm)

Length: 7.5 ounces (19.1 cm)

Weight: 4.5 ounces (126 gm)

Mounting: Occupies one double-wide iSBX MULTIMODULE™ position on boards; increases board height (host plus iSBX board) to 1.13 inches (2.87 cm).

Data Organization and Capacity

Standard Size Drives

	Double Density						Single Density					
	IBM System 34			Non-IBM			IBM System 3740			Non-IBM		
Bytes per Sector	256	512	1024	2048	4096	8192	128	256	512	1024	2048	4096
Sectors per Track	26	15	8	4	2	1	26	15	8	4	2	1
Tracks per Diskette	77			77			77			77		
Bytes per Diskette (Formatted, per diskette surface)	512,512 (256 bytes/sector) 591,360 (512 bytes/sector) 630, 784 (1024 bytes/sector)			630,784			256,256 (128 byte/sector) 295,680 (256 bytes/sector) 315, 392 (512 bytes/sector)			315,392		

Electrical Characteristics

Power Requirements: +5VDC @ 1.7A max.

Environmental Characteristics

Temperature: 0°C to +55° (operating); −55°C to +85°C (non-operating).

Humidity: Up to 90% Relative Humidity without condensation (operating); all conditions without condensation or frost (non-operating).

Reference Manual

145911— iSBX 218A Flexible Disk Controller Hardware Reference Manual (NOT SUPPLIED).

Reference manuals may be ordered from any Intel sales representative, distributor office, or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

Drive Characteristics

	Standard Size	Mini Size
	Double/Single Density	**Double/Single Density**
Transfer Rate (K bytes/sec)	62.5/31.25	31.25/15.63
Disk Speed (RPM)	360	300
Step Rate Time (Programmable)	1 to 16 ms/track in 1 ms increments	2 to 32 ms/track in 2 ms increments
Head Load Time (Programmable)	2 to 254 ms in 2 ms increments	4 to 508 ms in 4 ms increments
Head Unload Time (Programmable)	16 to 240 ms in 16 ms increments	32 to 480 ms in 32 ms increments

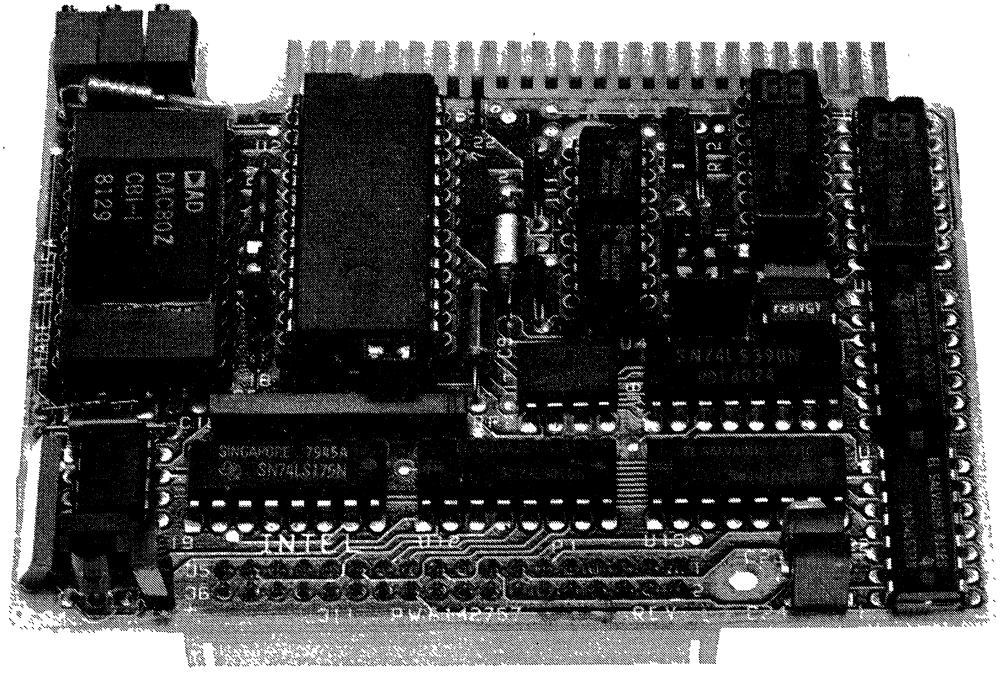
ORDERING INFORMATION

Order Code	Description
SBX218A	Flexible Disk Controller

iSBX™ 311*
ANALOG INPUT MULTIMODULE™ BOARD

- **Low Cost Analog Input Via iSBX™ Bus Connector**
- **8 Differential/16 Single-Ended, Fault Protected Inputs**
- **20 mV to 5V Full Scale Input Range, Resistor Gain Selectable**
- **Unipolar (0 to +5V) or Bipolar (−5V to +5V) Input, Jumper Selectable**
- **12-Bit Resolution Analog-To-Digital Converter**
- **18 KHz Samples Per Second Throughput to Memory**

The Intel iSBX 311 Analog Input MULTIMODULE board provides simple interfacing of non-isolated analog signals to any iSBC board having an IEEE 959 iSBX I/O Expansion Bus connector. The single-wide iSBX 311 plugs directly onto the iSBC board, providing data acquisition of analog signals from eight differential or sixteen single-ended voltage inputs, jumper selectable. Resistor gain selection is provided for both low level (20 mV full scale range) and high level (5 volt FSR) signals. Incorporating the latest high quality IC components, the iSBX 311 MULTIMODULE board provides 12 bit resolution, 11 bit accuracy, and a simple programming interface, all on a low cost SBX board.

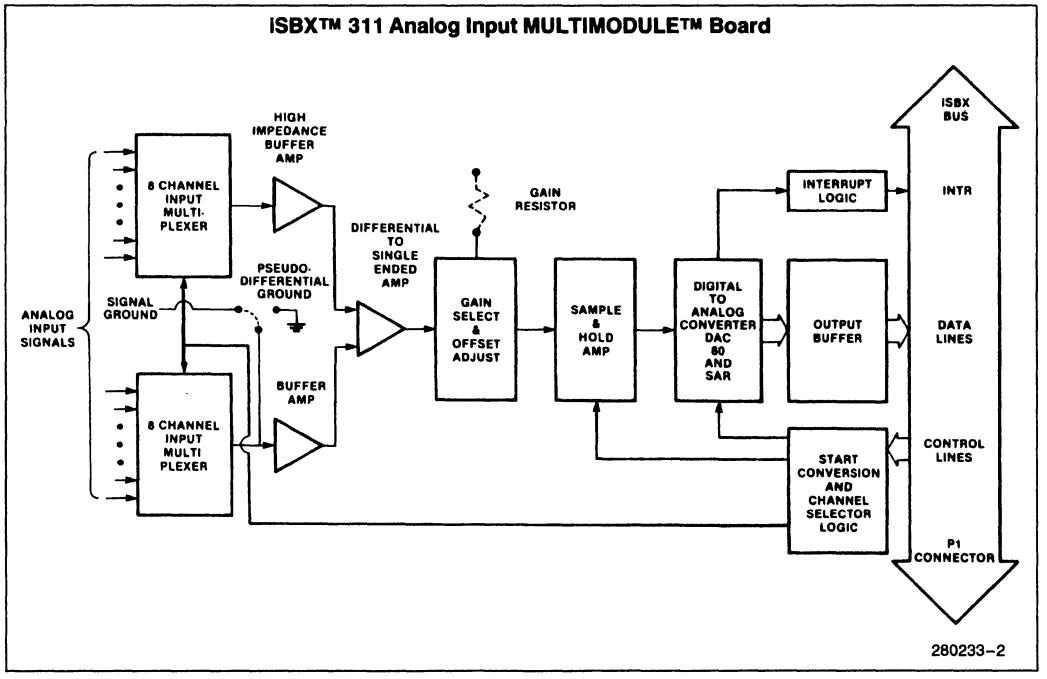


280233-1

*The iSBX™ 311 is also manufactured under product code piSBX™ 311 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280233-003

BLOCK DIAGRAM



SPECIFICATIONS

Inputs—8 differential. 16 single-ended. Jumper selectable.
Voltage Range—−5 to +5 volts (bipolar). 0 to +5 volts (unipolar). Jumper selectable.

Gain—User-configurable through installation of two resistors. Factory-configured for gain of X1.

Resolution—12 bits over full scale range (1.22 mV at 0–5V, 5 μV at 0–20 mV).

Accuracy—

Gain	Accuracy at 25°C
1	±0.035% ± ½ LSB
5	±0.035% ± ½ LSB
50	±0.035% ± ½ LSB
250	±0.035% ± ½ LSB

NOTE:
Figures are in percent of full scale reading. At any fixed temperature between 0° and 60°C, the accuracy is adjustable to ±0.035% of full scale.

Dynamic Error—±0.015% FSR for transitions.

Gain TC (at Gain = 1): 30 PPM per degree centigrade (typical); 56 PPM per degree centigrade (max).

Offset TC (in percent of FSR/°C):

Gain	Offset
1	0.0018
5	0.0036
50	0.024
250	0.116

Offset is measured with user-supplied 10 PPM/°C gain resistors installed.

Input Protection—±30 volts.

Input Impedance—20 MΩ (minimum).

Conversion Speed—50 ms (nominal).

Common Mode Rejection Ratio—60 db (minimum).

Sample and hold—sample time 15 ms.

Aperture—hold aperture time: 120 ns.

Connectors—

Interface	Pins (Qty)	Centers		Mating Connectors
		In	cm	
J1 8/16 Channels Analog	50	0.1	0.254	3m 3415-000

Physical Characteristics

Width: 9.40 cm (3.7 inches)

Length: 6.35 cm (2.5 inches)

Height: 2.03 cm (0.80 inch) MULTIMODULE board only

2.82 cm (1.13 inches) MULTIMODULE and iSBC board

Weight: 68.05 gm (2.4 ounces)

Electrical Characteristics (from iSBX connector)

V_{cc} = ±5 volts (±0.25V), I_{cc} = 250 mAmax
V_{dd} = +12 volts (±0.6V), I_{dd} = 50 mAmax
V_{ss} = −12 volts (±0.6V), I_{ss} = 55 mAmax

Environmental Characteristics

Operating Temperature: 0° to 60°C (32° to 140°C)

Relative Humidity: to 90% (without condensation)

Reference Manuals

142913— iSBX 311 Analog Input MULTIMODULE Board Hardware Reference Manual (order separately)

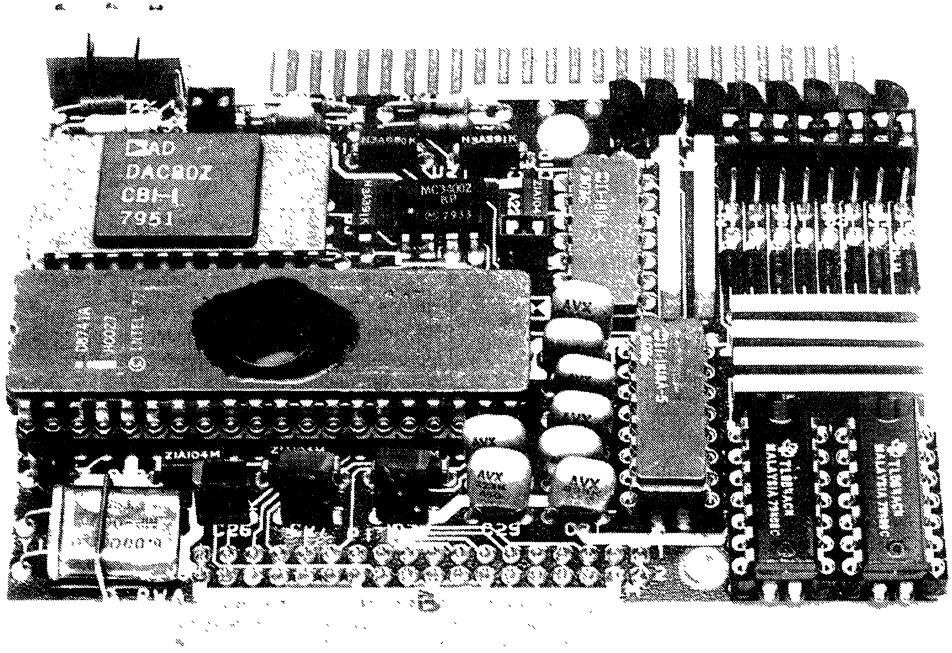
ORDERING INFORMATION

Order Code	Description
SBX311	Analog Input MULTIMODULE Board

iSBX™ 328*
ANALOG OUTPUT MULTIMODULE™ BOARD

- **Low Cost Analog Output Via iSBX™ Bus Connector**
- **8 Channel Output, Current Loop or Voltage in any Mix**
- **4–20 mA Current Loop; 5V Unipolar or Bipolar Voltage Output**
- **12-Bit Resolution**
- **0.035% Full Scale Voltage Accuracy @ 25°C**
- **Programmable Offset Adjust in Current Loop Mode**

The Intel iSBX 328 MULTIMODULE board provides analog signal output for any intelligent board having an IEEE 959 iSBX I/O Expansion bus connector. The single-wide iSBX 328 plugs directly onto the host board, providing eight independent output channels of analog voltage for meters, programmable power supplies, etc. Voltage output can be mixed with current loop output for control of popular 4–20 mA industrial control elements. By using an Intel single chip computer (8041) for refreshing separate sample-hold amplifiers through a single 12 bit DAC, eight channels are contained on a single SBX board for high density and low cost per channel. High quality analog components provide 12 bit resolution, and slew rates per channel of 0.1V per microsecond. Maximum channel update rates are 5 KHz on a single channel to 1 KHz on all eight channels.



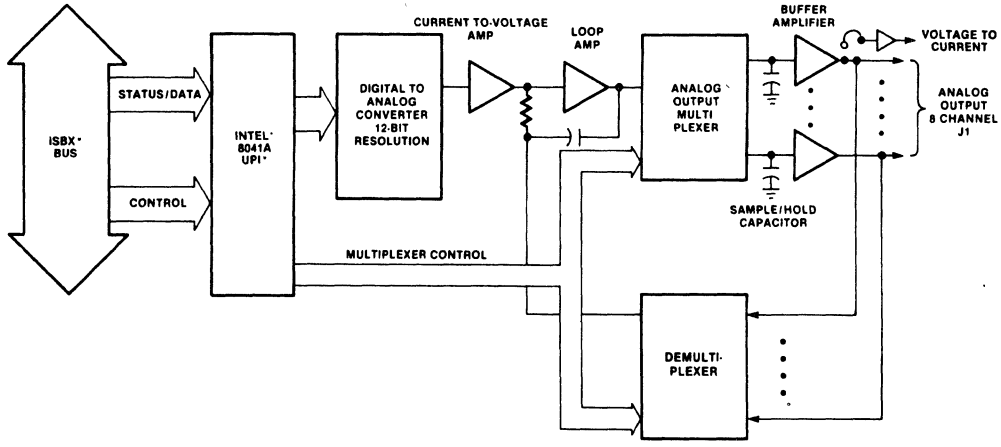
280234-1

*The iSBX™ 328 is also manufactured under product code piSBX™ 328 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280234-002

BLOCK DIAGRAM

iSBC® 328 Analog Output MULTIMODULE™ Board Block Diagram



280234-2

SPECIFICATIONS

Outputs — 8 non-isolated channels, each independently jumpered for voltage output or current loop output mode.

Voltage Ranges — 0 to +5V (unipolar operation) −5 to +5V (bipolar operation)

Current Loop Range — 4 to 20 mA (unipolar operation only)

Output Current — ±5 mA maximum (voltage mode-bipolar operation)

Load Resistance — 0 to 250Ω with on-board iSBX power. 1000Ω minimum with 30 VDC max. external supply

Compliance Voltage — 12V using on-board iSBX power. If supplied by user, up to 30 VDC max

Resolution — 12 bits bipolar or unipolar

Slew Rate — 0.1V per microsecond minimum

Single Channel Update Rate — 5 KHz

Eight Channel Update Rate — 1 KHz

Output Impedance — 0.1Ω. Drives capacitive loads up to 0.05 microfarads. (approx. 1000 foot cable)

Temperature Coefficient — 0.005%/°C

Refresh and Throughput Rates**	
Refresh 1 channel (no new data):	80 μs
Refresh all 8 channels (no new data):	650 μs
Update and refresh 1 channel with new data: firmware program 2	150 μs
for each additional channel	130 μs
Update and refresh 1 channel with new data: firmware program 1 or 3	200 μs
for each additional channel	155 μs
Update and refresh all 8 channels (all new data): firmware program 2	1,050 ms
per channel of new data	50 μs
Update and refresh all 8 channels (all new data): firmware program 1or 3	1,280 ms
per channel of new data	80 μs

**All times nominal

Accuracy—

Mode	Accuracy	Ambient Temp
Voltage-Unipolar, typical	± 0.025% FSR	@ 25°C
Voltage-Unipolar, maximum	± 0.035% FSR	@ 25°C
Voltage-Unipolar, typical	± 0.08% FSR	@ 0° to 60°C
Voltage-Unipolar, maximum	± 0.19% FSR	@ 0° to 60°C
Voltage-Bipolar, typical	± 0.025% FSR	@ 25°C
Voltage-Bipolar, maximum	± 0.035% FSR	@ 25°C
Voltage-Bipolar, typical	± 0.09% FSR	@ 0° to 60°C
Voltage-Bipolar, maximum	± 0.17% FSR	@ 0° to 60°C
Current Loop, typical	± 0.07% FSR	@ 25°C
Current Loop, maximum	± 0.08% FSR	@ 25°C
Current Loop, typical	± 0.17% FSR	@ 0° to 60°C
Current Loop, maximum	± 0.37% FSR	@ 0° to 60°C

Connectors—

Interface	Pins (Qty)	Centers in	Centers cm	Mating Connectors
J1 8/16 channels analog	50	0.1	0.254	3m 3415-000

Physical Characteristics

Width: 9.40 cm (3.7 inches)

Length: 6.35 cm (2.5 inches)

Height: 1.4 cm (0.56 inch) MULTIMODULE board only

2.82 cm (1.13 inches) MULTIMODULE and iSBC board.

Weight: 85.06 gm (3.0 ounces)

Electrical Characteristics

V_{CC} = ±5V (0.25V), I_{CC} = 140 mA max

V_{DD} = ±12V (±0.6V), I_{DD} = 45 mA max (voltage mode)

= 200 mA max (current loop mode

V_{SS} = −12V (±0.6V), I_{SS} = 55 mA max

Environmental Characteristics

Operating Temperature: 0° to 60°C (32° to 140°C)

Relative Humidity: to 90% (without condensation)

Reference Manuals

142914— iSBX 328 Analog Output MULTI-MODULE Board Hardware Reference Manual (Order Separately)

Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

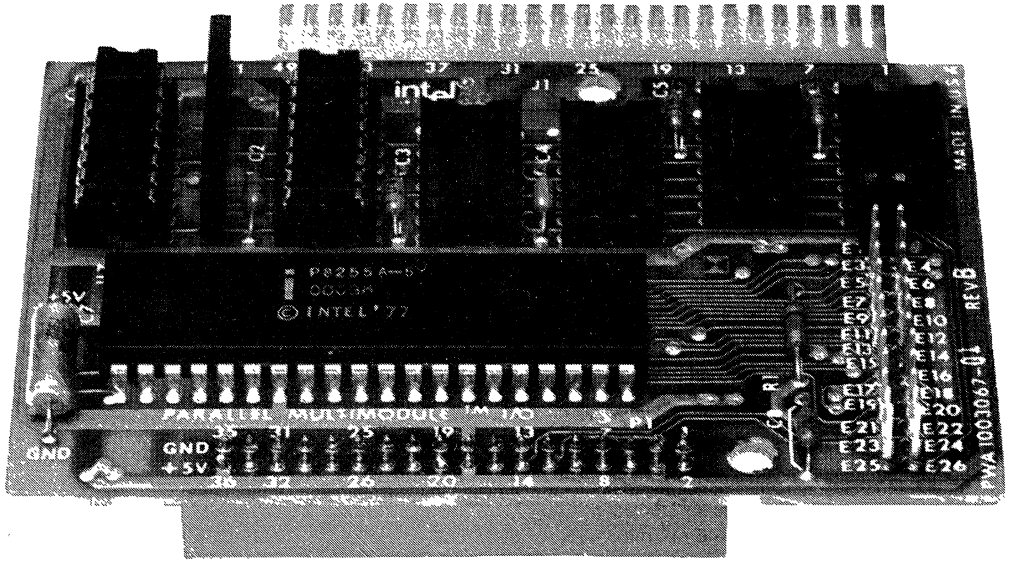
ORDERING INFORMATION

Order Code	Description
SBX328	Analog Output MULTIMODULE Board

iSBX™ 350*
PARALLEL I/O MULTIMODULE™ BOARD

- **iSBX™ Bus Compatible I/O Expansion**
- **24 Programmable I/O Lines with Sockets for Interchangeable Line Drivers and Terminators**
- **Three Jumper Selectable Interrupt Request Sources**
- **Accessed as I/O Port Locations**
- **Single +5V Low Power Requirement**
- **iSBX Bus On-Board Expansion Eliminates MULTIBUS® System Bus Latency and Increases System Throughput**

The Intel iSBX 350 Parallel I/O MULTIMODULE Board is a member of Intel's line of IEEE 959 iSBX I/O Expansion Bus products. The iSBX 350 module provides 24 programmable I/O lines with sockets for interchangeable line drivers and terminators. The iSBX board is closely coupled to the host board through the iSBX bus, and as such, offers maximum on-board performance and frees MULTIBUS system traffic for other system resources. In addition, incremental power dissipation is minimal requiring only 1.6 watts (not including optional driver/terminators).



280235-1

*The iSBX™ 350 is also manufactured under product code piSBX™ 350 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280235-002

SPECIFICATIONS

Parallel I/O Port Operation Modes

Port	Lines (qty)	Mode of Operation					Control
		Unidirectional				Bidirectional	
		Input		Output			
		Unlatched	Latched & Strobed	Latched	Latched & Strobed		
A	8	X	X	X	X	X	
B	8	X	X	X	X		
C	4	X		X			X(1)
	4	X		X			X(1)

NOTE:
1. Part of Port C must be used as a control port when either Port A or Port B are used as a latched and strobed input or a latched and strobed output port or Port A is used as a bidirectional port.

Word Size

Data: 8 Bits

I/O Addressing

8255A-5 Ports	iSBX 350 Address
Port A	X0 or X4
Port B	X1 or X5
Port C	X2 or X6
Control	X3 or X7
Reserved	X8 to XF

NOTE:
The first digit of each port I/O address is listed as "X" since it will change dependent on the type of host iSBC microcomputer used. Refer to the Hardware Reference Manual for your host iSBC microcomputer to determine the first digit of the port address.

I/O Capacity

24 programmable lines (see Table 1)

Access Time

Read: 250 ns max.

Write: 300 ns max.

NOTE:
Actual transfer speed is dependent upon the cycle time of the host microcomputer.

Interrupts

Interrupt requests may originate from the programmable peripheral interface (2) or the user specified I/O (1).

Parallel Interface Connectors

Interface	No. of Pairs/ Pins	Centers (in.)	Connector Type	Vendor	Vendor Part No.
Parallel I/O Connector	25/50	0.1	Female	3M	3415-0001 with Ears
Parallel I/O Connector	25/50	0.1	Female Soldered	GTE Sylvania	6AD01251A1DD

NOTE:
Connector compatible with those listed may also be used.

Line Drivers and Teminators

I/O Drivers—The following line drivers and terminators are all compatible with the I/O driver sockets on the iSBX 350.

Driver	Characteristic	Sink Current (mA)
7438	I, OC	48
7437	I	48
7432	NI	16
7426	I, OC	16
7409	NI, OC	16
7408	NI	16
7403	I, OC	16
7400	I	16

NOTE:
I = Inverting, NI = Non-Inverting, OC = Open Collector

Port 1 has 25 mA totem pole drivers and 1 kΩ terminators.

I/O Terminators—220Ω/330Ω divider or 1 kΩ pull up - user supplied.

Physical Characteristics

Width: 7.24 cm (2.85 in.)

Length: 9.40 cm (3.70 in.)

Height*: 2.04 cm (0.80 in.) iSBX 350 Board

2.86 cm (1.13 in.) iSBX 350 Board + Host Board

Weight: 51 gm (1.79 oz)

*See Figure 2

Electrical Characteristics

DC Power Requirements

Power Requirements	Configuration
+5 @ 320 mA	Sockets XU3, XU4, XU5, and XU6 empty (as shipped).
+5V @ 500 mA	Sockets XU3, XU4, XU5, and XU6 contain 7438 buffers.
+5V @ 620 mA	Sockets XU3, XU4, XU5, and XU6 contain iSBC 901 termination devices.

Environmental

Operating Temperature: 0°C to +55°C

Reference Manual

9803191—iSBX 350 Parallel I/O MULTIMODULE Manual (NOT SUPPLIED)

Reference Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Ave., Santa Clara, California 95051.

ORDERING INFORMATION

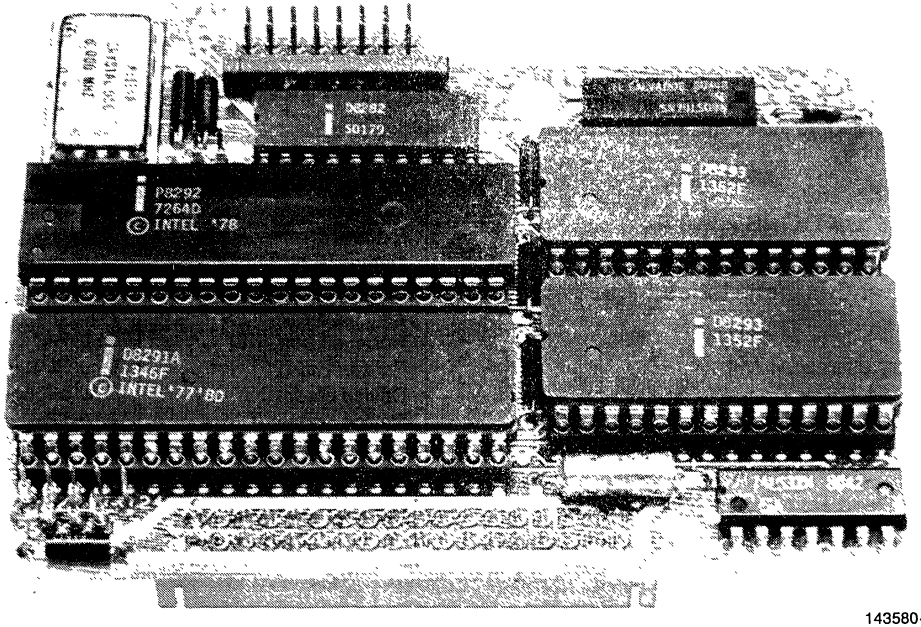
Order Code	Description
SBX350	Parallel I/O MULTIMODULE Board

iSBX™ 488*
GPIB MULTIMODULE™ BOARD

- **Complete IEEE 488-1978 Talker/ Listener Functions Including:**
  - **— Addressing, Handshake Protocol, Service Request, Serial and Parallel Polling Schemes**
- **Complete IEEE 488-1978 Controller Functions Including:**
  - **— Transfer Control, Service Requests and Remote Enable**
- **Simple Read/Write Programming**
- **Software Functions Built into VLSI Hardware for High Performance, Low Cost and Small Size**
- **Standard iSBX Bus Interface for Easy Connection to Intel iSBC™ Boards**
- **IEEE 488-1978 Standard Electrical Interface Transceivers**
- **Five Volt Only Operation**

The Intel iSBX 488 GPIB Talker/Listener/Controller MULTIMODULE board provides a standard interface from any Intel iSBC board equipped with an IEEE 959 iSBX I/O Expansion Bus connector to over 600 instruments and computer peripherals that use the IEEE 488-1978 General Purpose Interface Bus. By taking full advantage of Intel's VLSI technology the single-wide iSBX 488 MULTIMODULE board implements the complete IEEE 488-1978 Standard Digital Interface for Programmable Instrumentation on a single low cost board. The iSBX 488 MULTIMODULE board includes the 8291A GPIB Talker/Listener, 8292 GPIB Controller and two 8293 GPIB Transceiver devices. This board represents a significant step forward in joining microcomputers and instrumentation using industry standards such as the MULTIBUS® system bus, iSBX bus and IEEE 488-1978. The high performance iSBX 488 MULTIMODULE board mounts easily on Intel iSBX bus compatible single board computers.

A simple user programming interface for easy reading, writing and monitoring of all GPIB functions is provided. This intelligent interface minimizes the impact on host processor bandwidth.

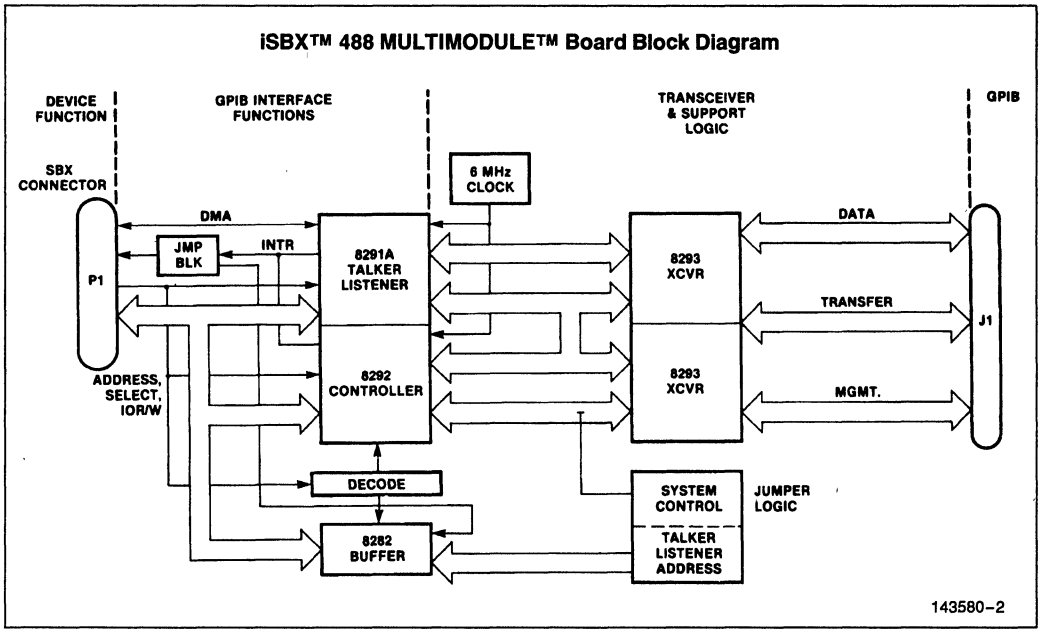


143580-1

*The iSBX™ 488 is also manufactured under product code piSBX™ 488 by Intel Puerto Rico, Inc.

September 1989
Order Number: 143580-002

BLOCK DIAGRAM



Installation

The iSBX 488 MULTIMODULE board plugs directly onto the female iSBX connector available on many Intel iSBC boards. The MULTIMODULE board is then secured at one additional point with nylon hardware (supplied) to insure the mechanical security of the assembly.

SPECIFICATIONS

Physical Characteristics

Width: 3.70 in (0.94 cm)

Length: 2.85 in (7.24 cm)

Height: 0.8 in (2.04 cm)

Weight: 3.1 oz (87.8 gm)

GPIB Data Rate*

300K bytes/sec transfer rate with DMA host iSBC board

GPIB Functions Supported

IEEE 488-1978 Functions(1)

Function	iSBX™ 488 Supported IEEE Subsets
Source Handshake (SH)	SH0, SH1
Acceptor Handshake (AH)	AH0, AH1
Talker (T)	T0 through T8
Extended Talker (TE)	TE0 through TE8
Listener (L)	L0 through L4
Extended Listener (LE)	LE0 through LE9
Service Request (SR)	SR0, SR1
Remote Local (RL)	RL0, RL1
Parallel Poll (PP)	PP0, PP1, PP2
Device Clear (DC)	DC0 through DC2
Device Trigger (DT)	DT0, DT1
Controller (C)	C0 through C28

NOTE:
1. For detailed information refer to IEEE Standard Digital Interface for Programmable Instrumentation published by The Institute of Electrical and Electronics Engineers, Inc. 1978.

50K bytes/s transfer rate using programmed I/O
730 ns Data Accept Time

*Data rates are iSBX board maximum. Data rates will vary and can be slower depending on host iSBC board and user software driver.

Electrical Characteristics

DC Power Requirements: V_{CC} = +5 VDC ±5%

I_{CC} = 600 milliamps maximum

GPIB Electrical and Mechanical Specifications

Conforms to IEEE 488-1978 standard electrical levels and mechanical connector standard when purchased with the iSBC 988 GPIB cable.

Environmental Characteristics

Operating Temperature: 0° to 60°C (32° to 140°F)

Relative Humidity: Up to 90% R.H. without condensation.

Reference Manual

143154— iSBX 488 GPIB MULTIMODULE Board Hardware Reference Manual (not supplied).

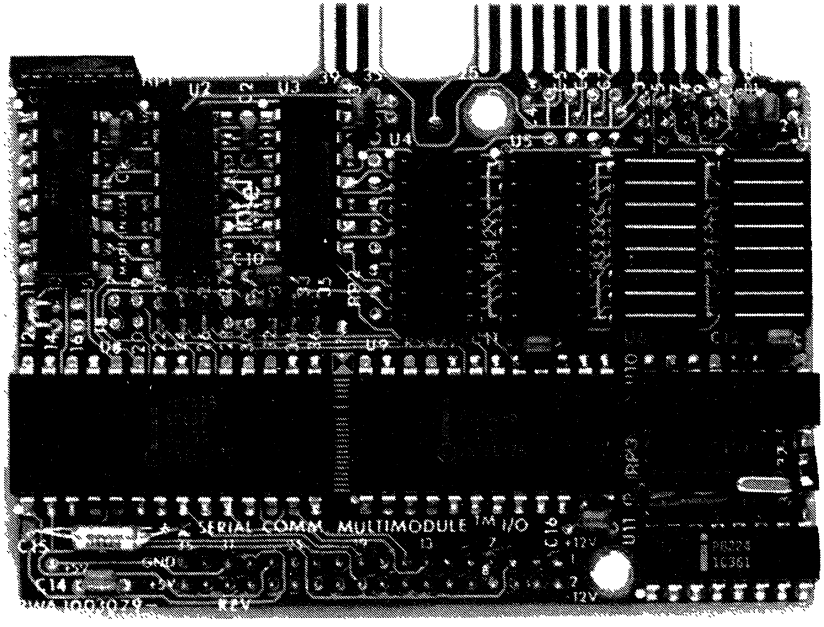
ORDERING INFORMATION

Order Code	Description
SBX488	GPIB MULTIMODULE

iSBX™ 351*
SERIAL I/O MULTIMODULE™ BOARD

- **IEEE959 iSBX™ Bus Compatible I/O Expansion**
- **Programmable Synchronous/ Asynchronous Communications Channel with RS232C or RS449/422 Interface**
- **Software Programmable Baud Rate Generator**
- **Two Programmable 16-Bit BCD or Binary Timer/Event Counters**
- **Four Jumper Selectable Interrupt Request Sources**
- **Accessed as I/O Port Locations**
- **Low Power Requirements**
- **Single +5V when Configured for RS449/422 Interface**
- **iSBX Bus On-Board Expansion Eliminates MULTIBUS® System Bus Latency and Increases System Throughput**

The Intel iSBX 351 Serial I/O MULTIMODULE board is a member of Intel's line of IEEE 959 iSBX I/O Expansion Bus compatible products. The iSBX 351 module provides one RS232C or RS449/422 programmable synchronous/asynchronous communications channel with software selectable baud rates. Two general purpose programmable 16-bit BCD or binary timers/event counters are available to the host board to generate accurate time intervals under software control. The iSBX board is closely coupled to the host board through the SBX bus, and as such, offers maximum on-board performance and frees MULTIBUS system traffic for other system resources. In addition, incremental power dissipation is minimal requiring only 3.0 watts (assumes RS232C interface).



280236-1

*The iSBX™ 351 is also manufactured under product code piSBX™ 351 by Intel Puerto Rico, Inc.

October 1989
Order Number: 280236-002

SPECIFICATIONS

I/O Addressing

I/O Address for an 8-Bit Host	I/O Address for a 16-Bit Host	Chip Select	Function
X0, X2, X4 or X6	Y0, Y4, Y8 or YC	8251A USART	Write: Data Read: Data
X1, X3, X5 or X7	Y2, Y6, YA or YE	MCS0/ Activated (True)	Write: Mode or Command Read: Status
X8 or XC	Z0 or Z8	8253 PIT	Write: Counter 0 Load: Count (N) Read: Counter 0
X9 or XD	Z2 or ZA	MSC1/Activated (True)	Write: Counter 1 Load: Count N Read: Counter 1
XA or XE	Z4 or ZC		Write: Counter 2 Load: Count (N) Read: Counter 2
XB or XF	Z6 or ZE		Write: Control Read: None

NOTE:
X = The iSBX base address that activates MCS0 & MSC1 for an 8-bit host.
Y = The iSBX base address that activates MCS0 for a 16-bit host.
Z = The iSBX base address that activates MCS1 for a 16-bit host.
The first digit, X, Y or Z, is always a variable, since it will depend on the type of host microcomputer used. Refer to the Hardware Reference Manual for your host microcomputer to determine the first digit of the I/O base address.
The first digit of each port I/O address is listed as "X" since it will change depending on the type of host iSBC microcomputer used. Refer to the Hardware Reference Manual for your host iSBC microcomputer to determine the first digit of the I/O address.

Word Size

Data—8 bits

Access Time

Read—250 ns max
Write—300 ns max

NOTE:
Actual transfer speed is dependent upon the cycle time of the host microcomputer.

Serial Communications

Synchronous—5–8-bit characters; internal character synchronization; automatic sync insertion; even, odd or no parity generation/detection.

Asynchronous—5–8-bit characters; break character generation and detection; 1, 1½, or 2 stop bits; false start bit detection; even, odd or no parity generation/detection.

Interval Timer and Baud Rate Generator

Input Frequency (selectable):

1.23 MHz ±0.1% (0.813 μs period nominal)
153.6 kHz ±0.1% (6.5 μs period nominal)

Sample Baud Rate

8253 PIT(1) Frequency (kHZ, Software Selectable)	8251 USART Baud Rate (Hz)(2)		
	Synchronous	Asynchronous	
		÷16	÷64
307.2	—	19200	4800
153.6	—	9600	2400
76.8	—	4800	1200
38.4	38400	2400	600
19.2	19200	1200	300
9.6	9600	600	150
4.8	4800	300	75
2.4	2400	150	—
1.76	1760	110	—

NOTES:
1. Frequency selected by I/O writes of appropriate 16-bit frequency factor to Baud Rate Register.
2. Baud rates shown here are only a sample subset of possible software-programmable rates available. Any frequency from 18.75 Hz to 614.4 kHz may be generated utilizing on-board crystal oscillator and 16-bit Programmable Interval Timer (used here as frequency divider).

Output Frequency

	Rate Generator (Frequency)		Real-Time Interrupt (Interval)	
	Min	Max	Min	Max
Single Timer(1)	18.75 Hz	614.4 kHz	1.63 μs	53.3 ms
Single Timer(2)	2.34 Hz	76.8 kHz	13.0 μs	426.7 ms
Dual Timer(3) (Counters 0 and 1 in Series)	0.000286 Hz	307.2 kHz	3.26 μs	58.25 min
Dual Timer(4) (Counters 0 and 1 in Series)	0.0000358 Hz	38.4 kHz	26.0 μs	7.77 hrs

NOTES:
1. Assuming 1.23 MHz clock input.
2. Assuming 153.6 kHz clock input.
3. Assuming Counter 0 has 1.23 MHz clock input.
4. Assuming Counter 0 has 153.6 kHz clock input.

Interrupts

Interrupt requests may originate from the USART (2) or the programmable timer (2).

Interfaces

IEEE959 iSBX I/O Expansion Bus

Serial—configurable of EIA Standards RS232C or RS449/422

EIA Standard RS232C signals provided and supported.

Clear to Send (CTS)
Data Set Ready (DSR)
Data Terminal Ready (DTR)
Request to Send (RTS)
Receive Clock (RXC)
Receive Data (RXD)
Transmit Clock (DTE TXC)
Transmit Data (TXD)

EIA Standard RS449/422 signals provided and supported.

Clear to Send (CS)
Data Mode (DM)
Terminal Ready (TR)
Request to Send (RS)
Receive Timing (RT)
Receive Data (RD)
Terminal Timing (TT)
Send Data (SD)

Physical Characteristics

Width: 7.24 cm (2.85 inches)

Length: 9.40 cm (3.70 inches)

Height*: 2.04 cm (0.80 inches)
iSBX 351 Board
2.86 cm (1.13 inches)
iSBX 351 Board and Host Board

Weight: 51 grams (1.79 ounces)
*(See Figure 2)

Electrical Characteristics

DC Power Requirements

Mode	Voltage	Amps (Max)
RS232C	+5V ±0.25V	460 mA
	+12V ±0.6V	30 mA
	−12V ±0.6V	30 mA
RS449/422	+5V ±0.25V	530 mA

Environmental Characteristics

Temperature: 0°C–55°C, free moving air across the base board and MULTIMODULE board.

Reference Manual

9803190— iSBX 351 Serial I/O MULTIMODULE Manual (NOT SUPPLIED)

Reference Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Ave., Santa Clara, California, 95051.

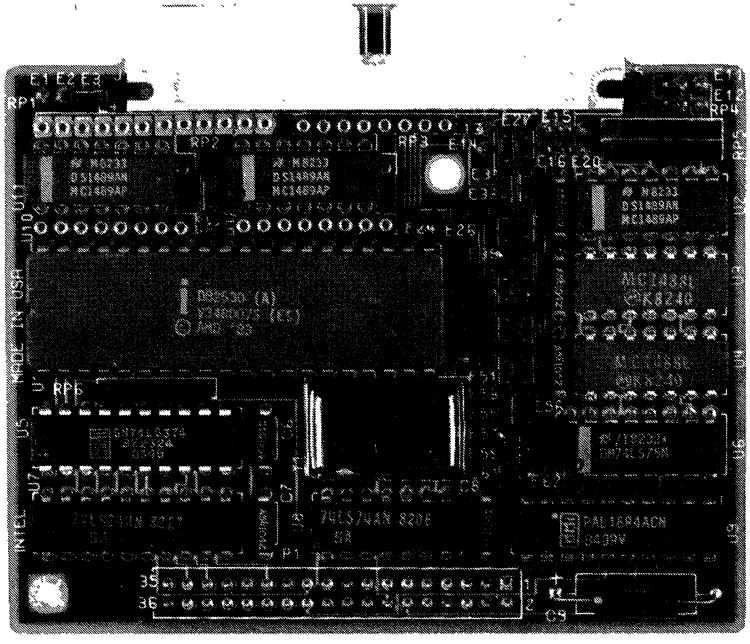
ORDERING INFORMATION

Order Code	Description
SBX351	Serial I/O MULTIMODULE Board

iSBX™ 354* DUAL CHANNEL SERIAL I/O MULTIMODULE™ BOARD

- **Two RS232C or RS422A/449 Programmable Synchronous/ Asynchronous Communications Channels**
- **Programmable Baud Rate Generation for Each Channel**
- **Full Duplex Operation**
- **IEEE 959 SBX Bus Compatible I/O Expansion**
- **Supports HDLC/SDLC, NRZ, NRZI or FM Encoding/Decoding**
- **Three Interrupt Options for Each Channel**
- **Low Power Requirements**

The Intel iSBX™ 354 Serial I/O MULTIMODULE board is a member of Intel's line of IEEE 959 iSBX I/O Expansion Bus products. Utilizing Intel's 82530 Serial Communications Controller component, the iSBX 354 module provides two RS232C or RS422A/449 programmable synchronous/asynchronous communications channels. The 82530 component provides two independent full duplex serial channels, on chip crystal oscillator, baud-rate generator and digital phase locked loop capability for each channel. The iSBX board connects to the host board through the iSBX bus. This offers maximum on-board performance and frees the MULTIBUS® System bus for use by other system resources.



280045-1

*The iSBX™ 354 is also manufactured under product code piSBX™ 354 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280045-004

SPECIFICATIONS

Word Size

Data—8 bits

Clock Frequency

4.9152 MHz

Serial Communications

Synchronous—Internal or external character synchronization on one or two synchronous characters

Asynchronous—5–8 bits and 1, 1½ or 2 stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection

Sample Baud Rate:

Synchronous X1 Clock	
Baud Rate	**82530 Count Value (Decimal)**
64000	36
48000	49
19200	126
9600	254
4800	510
2400	1022
1800	1363
1200	2046
300	8190
Asynchronous X16 Clock	
Baud Rate	**82530 Count Value (Decimal)**
19200	6
9600	14
4800	30
2400	62
1800	83
1200	126
300	510
110	1394

INTERFACES

SBX Bus: Meets the IEEE 959 Specification, Compliance Level: D8 F

Serial: Meets the EIA RS232C standard on Channels A and B. Meets the EIA RS422A/449 standard on Channels A and B, Multi-drop capability on Channel A only.

Signals Provided

RS232C DTE

-Transmit Data
-Receive Data
-Request to Send
-Clear to Send
-Data Set Ready
-Signal Ground
-Carrier Detect
-Transmit Clock (2)
-Receive Clock
-Data Terminal Ready
-Ring Indicator

RS232C DCE

-Transmit Data
-Receive Data
-Clear to Send
-Data Set Ready
-Signal Ground
-Carrier Detect
-Transmit Clock (2)
-Receive Clock
-Ring Indicator

RS422A/449

-Send Data
-Receive Timing
-Receive Data
-Terminal Timing
-Receive Common

I/O Port Addresses

Port Address		Function
8-Bit	16-Bit	
X0		Read Status Channel B Write Command Channel B
X2		Read Data Channel B Write Data Channel B
X4		Read Status Channel A Write Command Channel A
X6		Read Data Channel A Write Data Channel A
Y0		Read Disable RS422A/449 Buffer Write Enable RS422A/449 Buffer

NOTES:
1. The "X" and "Y" values depend on the address of the iSBX interface as viewed by the base board.
2. "X" corresponds with Activation of the MCS0/interface signal; "Y" corresponds with Activation of the MCS1/interface signal.

Power Requirements

+5V at 0.5A
+12V at 50 mA
−12V at 50 mA

Physical Characteristics

Width: 2.85 inches
Length: 3.70 inches
Height: 0.8 inches
Weight: 85 grams

ENVIRONMENTAL CHARACTERISTICS

Temperature: 0°C to 55°C operating at 200 linear feet per minute across baseboard and MULTIMODULE board

Humidity: To 90%, without condensation

ORDERING INFORMATION

Order Code	Description
SBX354	Dual Channel I/O MULTIMODULE

REFERENCE MANUAL

146531—iSBX 354 Channel Serial I/O Board Hardware Reference Manual

Reference manuals may be ordered from any Intel sales representative, distributor office, or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, CA 95051.

iSBX™ I/O EXPANSION BUS

- **IEEE 959-88 Industry Standard I/O Expansion Bus**
- **Provides Cost/Performance Effective On-Board Expansion of System Resources**
- **Supports Compatible 8- and 16-Bit Data Transfer Operations**
- **Low-Cost "Vehicle" to Incorporate the Latest VLSI I/O Technology into Microcomputer Systems**
- **Supported by a Broad Range of Host Single Board Computer Boards**

The iSBX™ I/O Expansion Bus is one of a family of standard bus structures resident within Intel's total system architecture. The iSBX bus is a modular, I/O expansion bus capable of increasing a single board computer's functional capability and overall performance by providing a structure to attach small iSBX MULTIMODULE™ boards to iSBC® base boards. It provides for rapid incorporation of new VLSI into iSBC MULTIBUS® systems, reducing the threat of system obsolescence. The iSBX bus offers users new economics in design by allowing both system size and system cost to be kept at minimum. As a result, the system design achieves maximum on-board performance while allowing the system bus interface to be used for other system activities.

The iSBX bus enables users to add-on capability to a system as the application demands it by providing off-the-shelf standard MULTIMODULE boards in the areas of graphics controllers, advanced mathematics functions, parallel and serial I/O, and disk and tape peripheral controllers. A full line of MULTIBUS boards and iSBX MULTIMODULE boards are available from Intel and other third party sources in the industry.

Its success as an industry standard has been reinforced by adoption of the SBX specification by the Institute of Electrical & Electronic Engineers — IEEE 959-88.

October 1989
Order Number: 280255-002

FUNCTIONAL DESCRIPTION

Bus Elements

The iSBX™ MULTIMODULE™ system is made up of two basic elements: base boards and iSBX MULTIMODULE boards. In an iSBX system, the role of the base board is simple. It decodes I/O addresses and generates the chip selects for the iSBX MULTIMODULE boards.

The iSBX bus supports two classes of base boards, those with direct memory access (DMA) support and those without. Base boards with DMA support have DMA controllers that work in conjunction with an iSBX MULTIMODULE board (with DMA capability) to perform direct I/O to memory or memory to I/O operations. Base boards without DMA support use a subset of the iSBX bus and simply do not use the DMA feature of the iSBX MULTIMODULE board.

The iSBX MULTIMODULE boards are small, specialized, I/O mapped boards which plug into base boards. The iSBX boards connect to the iSBX bus connector and convert iSBX bus signals to a defined I/O interface.

Bus Interface/Signal Line Descriptions

The iSBX bus interface can be grouped into six functional classes: control lines, address and chip select lines, data lines, interrupt lines, option lines, and power lines. The iSBX bus provides nine control lines that define the communications protocol between base board and iSBX MULTIMODULE boards. These control lines are used to manage the general operation of the bus by specifying the type of transfer, the coordination of the transfer, and the overall state of the transfer between devices. The five address and chip select signal lines are used in conjunction with the command lines to establish the I/O port address being accessed, effectively selecting the proper iSBX MULTIMODULE. The data lines on the iSBX bus can number 8 or 16, and are used to transmit or receive information to or from the iSBX MULTIMODULE ports. Two interrupt lines are provided to make interrupt requests possible from the iSBX board to the base board. Two option lines are reserved on the bus for unique user requirements, while several power lines provide +5 and ±12 volts to the iSBX boards.

Bus Pin Assignments

The iSBX bus uses widely available, reliable connectors that are available in 18/36 pin for 8-bit devices and 22/44 pin for 16-bit devices. The male iSBX connector is attached to the iSBX MULTIMODULE board and the female iSBX connector is attached to the base board. Figure 2 shows the dimensions and pin numbering of the 18/36 pin iSBX connector, while Figure 3 does the same for the 22/44 pin iSBX connector. A unique scheme allows the 16-bit female connector to support 8 or 16-bit male MULTIMODULE boards. Table 1 lists the signal/pin assignments for the bus.

Table 1. iSBX™ Signal/Pin Assignments

Pin(1)	Mnemonic	Description	Pin(1)	Mnemonic	Description
43	MD8	MDATA Bit 8	44	MD9	MDATA Bit 9
41	MDA	MDATA Bit A	42	MDB	MDATA Bit F
39	MDC	MDATA Bit C	40	MDD	MDATA Bit D
37	MDE	MDATA Bit E	38	MDF	MDATA Bit F
35	GND	Signal Gnd	36	+5V	+5V
33	MD0	MDATA Bit 0	34	MDRQT	M DMA Request
31	MD1	MDATA Bit 1	32	MDACK/	M DMA Acknowledge
29	MD2	MDATA Bit 2	30	OPT0	Option 0
27	MD3	MDATA Bit 3	28	OPT 1	Option 1
25	MD4	MDATA Bit 4	26	TDMA	Terminate DMA
23	MD5	MDATA Bit5	24		Reserved
21	MD6	MDATA Bit 6	22	MCS0/	M Chip Select 0
19	MD7	MDATA Bit 7	20	MCS/1	M Chip Select 1
17	GND	Signal Gnd	18	+5V	+5V
15	IORD/	I/O Read Cmd	16	MWAIT/	M Wait
13	IOWRT/	I/O Write Cmd	14	MINTRO	M Interrupt 0
11	MA0	M Address 0	12	MINTR1	M Interrupt 1
9	MA1	M Address 1	10		Reserved
7	MA2	M Address 2	8	MPST/	iSBX Multimodule Board Present
5	RESET	Reset	6	MCLK	M Clock
3	GND	Signal Gnd	4	+5V	+5V
1	+12V	+12V	2	−12V	−12V

NOTES:
1. Pins 37–44 are used only on 8/16-bit systems.
2. All undefined pins are reserved for future use.

Bus Operation Protocol

COMMAND OPERATION

The iSBX bus supports two types of transfer operations between iSBX elements: I/O Read and I/O Write. An iSBX board can respond to these I/O transfers using either full speed mode or extended mode.

For a full speed I/O Read (Figure 4) the base board generates a valid I/O address and a valid chip select for the iSBX MULTIMODULE board. After setup, the base board activates the I/O Read line causing the iSBX board to generate valid data from the addressed I/O port. The base board then reads the data and removes the read command, address, and chip select. The full speed I/O Write (Figure 5) operation is similar to the I/O Read except that the base board generates valid data on the lines and keeps the write command line active for the specified hold time.

The extended Read operation (Figure 6) is used by iSBX MULTIMODULE boards that aren't configured to meet full speed specifications. It's operation is similar to full speed mode, but must use a wait signal to ensure proper data transfer. The base board begins the operation by generating a valid I/O address and chip select. After setup, the base board activates the Read line causing the iSBX board to generate a Wait signal. This causes the CPU on the base board to go into a wait state. When the iSBX board has placed valid Read data on the data lines, the MULTIMODULE board will remove the Wait signal and release the base board CPU to read the data

and deactivate the command, address, and chip select. The extended Write operation (Figure 7) is similar to the extended Read except that the Wait signal is generated after the base board places valid Write data on the data lines. The iSBX board removes the Wait signal when the write pulse width requirements are satisfied, and the base board can then remove the write command after the hold time is met.

DMA OPERATION

An iSBX MULTIMODULE system can support DMA when the base board has a DMA controller and the iSBX MULTIMODULE board can support DMA mode. Burst mode DMA is fully supported, but for clarity and simplicity, only a single DMA transfer for an 8-bit base board is discussed.

A DMA cycle (Figure 8) is initiated by the iSBX board when it activates the DMA request line going to the DMA controller on the base board. When the DMA controller gains control of the base board bus, it acknowledges back to the iSBX board and activates an I/O or Memory Read. The DMA controller then activates an I/O or Memory Write respectively. The iSBX board removes the DMA request during the cycle to allow completion of the DMA cycle. Once the write operation is complete, the DMA controller is free to deactivate the write and read command lines after a data hold time.

INTERRUPT OPERATION

The iSBX MULTIMODULE board on the iSBX bus can support interrupt operations over its interrupt lines. The iSBX board initiates an interrupt by activating one of its two interrupt lines which connect to the base board. The CPU processes the interrupt and executes the interrupt service routine. The interrupt service routine signals the iSBX MULTIMODULE board to remove the interrupt, and then returns control to the main line program when the service routine is completed.

Please refer to the Intel iSBX Bus Specification for more detailed information on its operation and implementation.

SPECIFICATIONS

Word Size

Data: 8, 16-bit

Power Supply Specifications

Table 3.

Minimum (volts)	Nominal (volts)	Maximum (volts)	Maximum (current)*
+4.75	+5.0	+5.25	3.0A
+11.4	+12	+12.6	1.0A
−12.6	−12	−11.4	1.0A
—	GND	—	3.0A

NOTE:
*Per iSBX MULTIMODULE board mounted on base board.

Port Assignments

Table 2. iSBX™ MULTIMODULE™ Base Board Port Assignments

iSBX™ Connector Number	Chip Select	8-Bit Base Board Address	16-Bit Base Board Address (8-bit mode)	16-Bit Base Board Address (16-bit mode)
iSBX1	MCS0/ MCS1/	F0–F7 F8–FF	0A0–0AF 0B0–0BF	0A0,2,4,6,8, A,C,E 0A1,3,5,7,9, B,D,F
iSBX2	MCS0/ MCS1/	C0–C7 C8–CF	080–08F 090–09F	080,2,4,6,8 A,C,E 081,3,5,7,9, B,D,F
iSBX3	MCS0/ MCS1/	B0–B7 B8–BF	060–06F 060–06F	060,2,4,6,8 A,C,E 061,3,5,7,9, B,D,F

DC Specifications

Table 4. iSBX™ MULTIMODULE™ Board I/O DC Specifications

Output[1]

Bus Signal Name	Type[2] Drive	I_{OL} Max −Min (mA)	@Volts (V_{OL} Max)	I_{OH} Max −Min (μA)	@ Volts (V_{OH} Min)	C_O (Min) (pf)
MD0–MDF	TRI	1.6	0.5	−200	2.4	130
MINTRO-1	TTL	2.0	0.5	−100	2.4	40
MDRQT	TTL	1.6	0.5	−50	2.4	40
MWAIT/	TTL	1.6	0.5	−50	2.4	40
OPT1–2	TTL	1.6	0.5	−50	2.4	40
MPST/	TTL	Note 3				

Input[1]

Bus Signal Name	Type[2] Receiver	I_{IL} Max (mA)	@V_{IN} MAX (volts) Test Cond.	I_{IH} Max (μA)	@V_{IN}MAX (volts) Test Cond.	C_IMax (pf)
MD0–MDF	TRI	−0.5	0.4	70	2.4	40
MA0–MA2	TTL	−0.5	0.4	70	2.4	40
MCS0/–MCS1/	TTL	−4.0	0.4	100	2.4	40
MRESET	TTL	−2.1	0.4	100	2.4	40
MDACK/	TTL	−1.0	0.4	100	2.4	40
IORD/ IOWRT/	TTL	−1.0	0.4	100	2.4	40
MCLK	TTL	−2.0	0.4	100	2.4	40
OPT1–OPT2	TTL	−2.0	0.4	100	2.4	40

NOTES:
1. Per iSBX MULTIMODULE I/O board.
2. TTL = standard totem pole output. TRI = Three-state.
3. iSBX MULTIMODULE board must connect this signal to ground.

All Inputs: Max V_{IL} = 0.8V
Min V_{IH} = 2.0V

Connectors

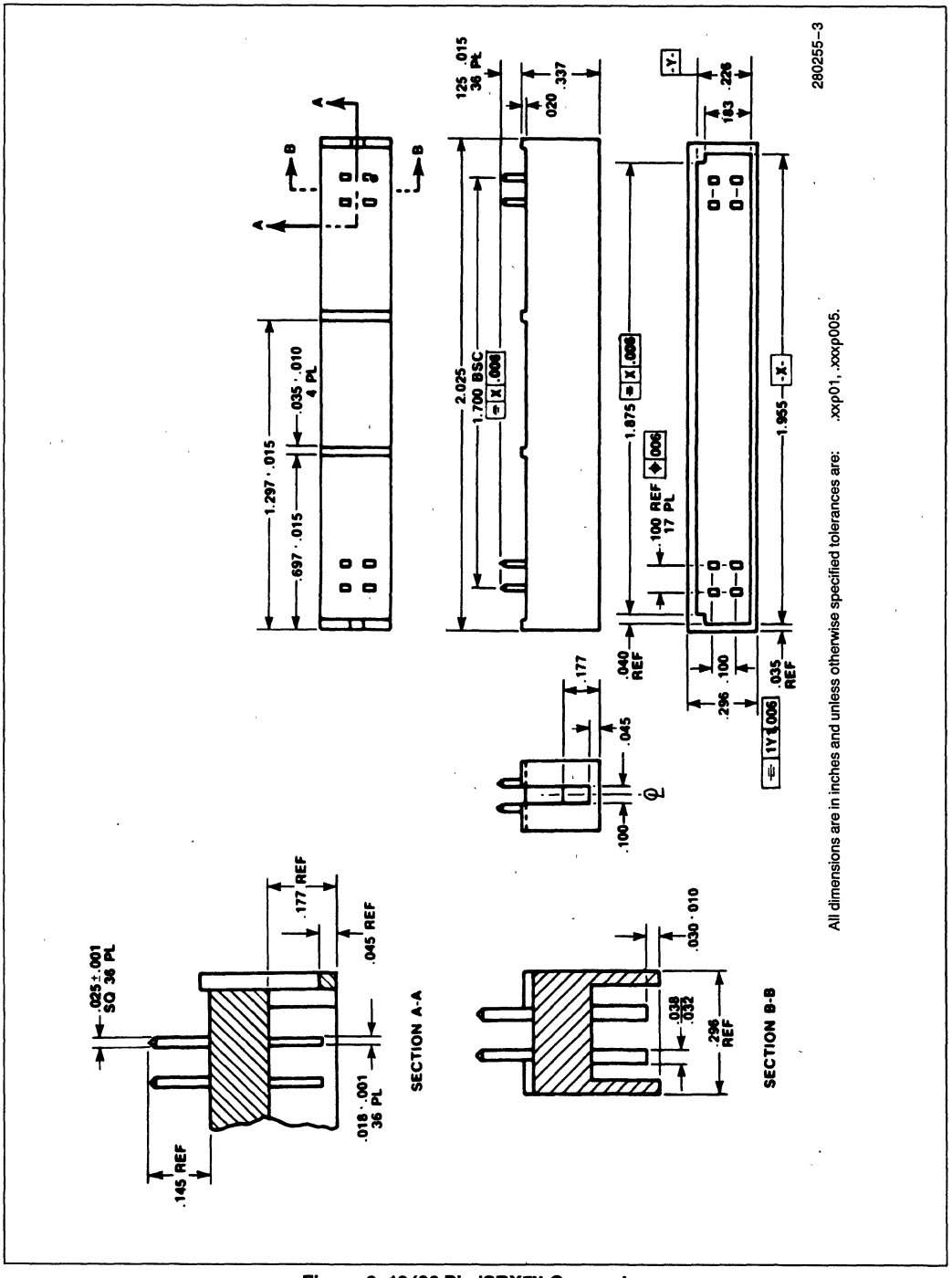


Figure 2. 18/36 Pin iSBX™ Connector

Connectors (Continued)

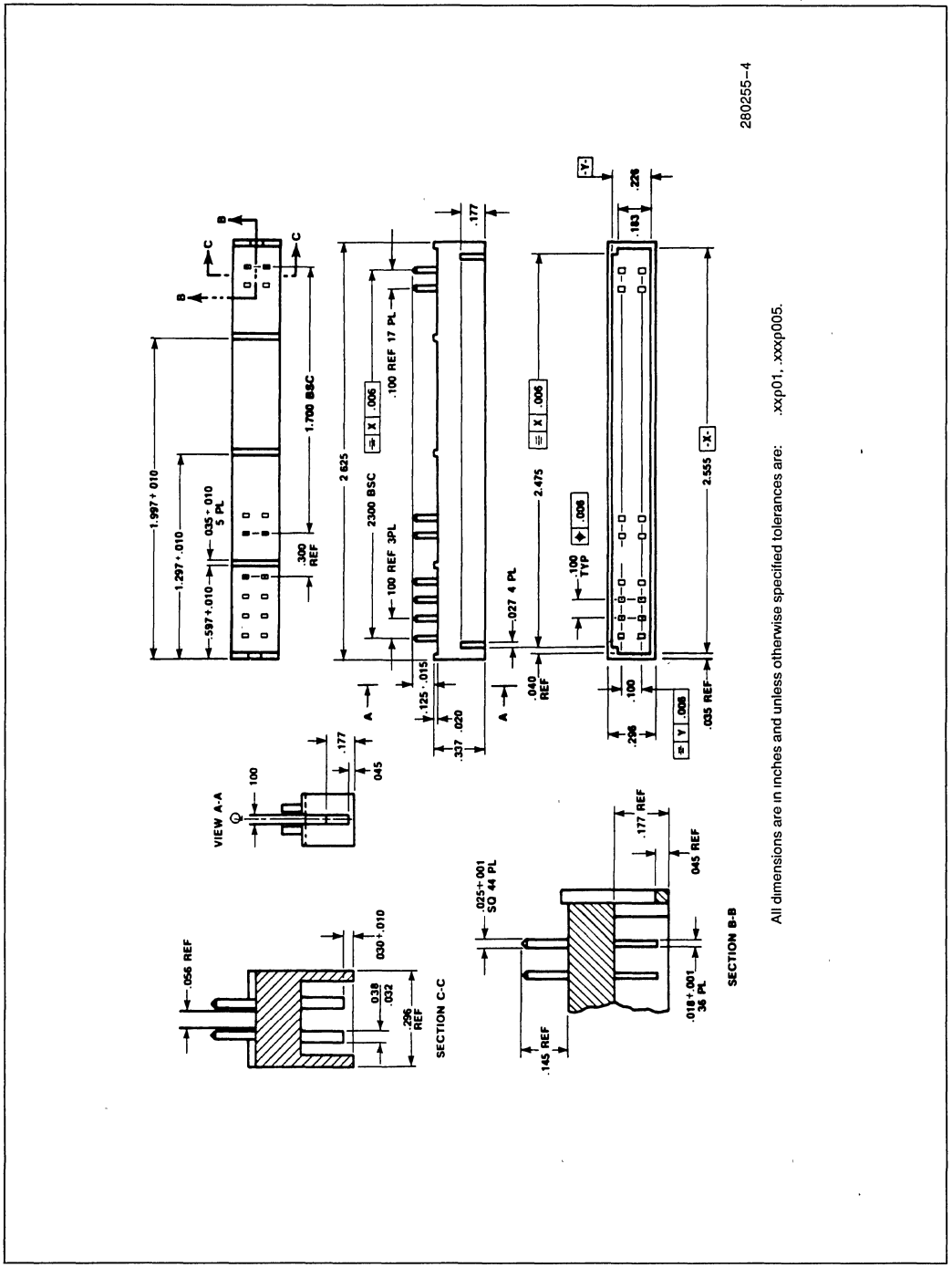


Figure 3. 22/44 Pin iSBX™ Connector

Bus Timing Diagrams

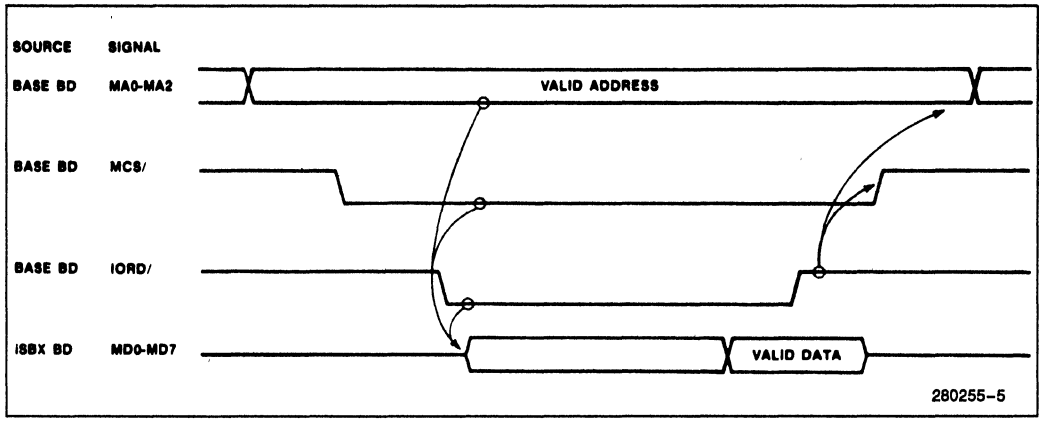


Figure 4. ISBX™ MULTIMODULE™ Read, Full Speed

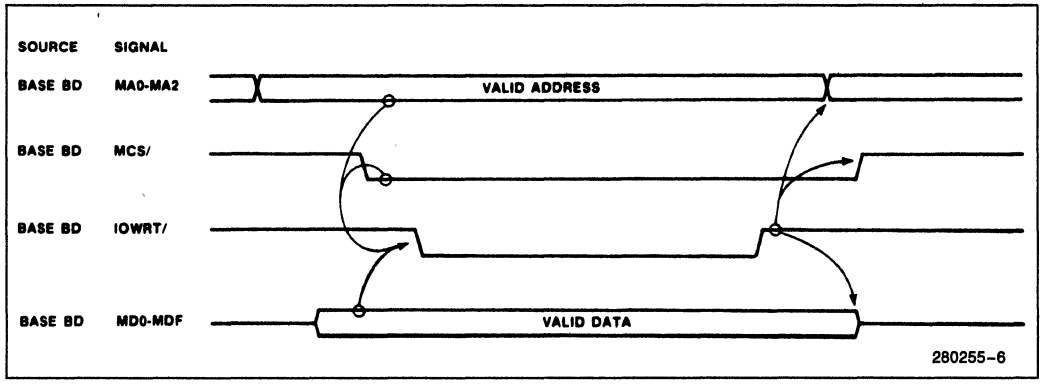


Figure 5. ISBX™ MULTIMODULE™ Board Write, Full Speed

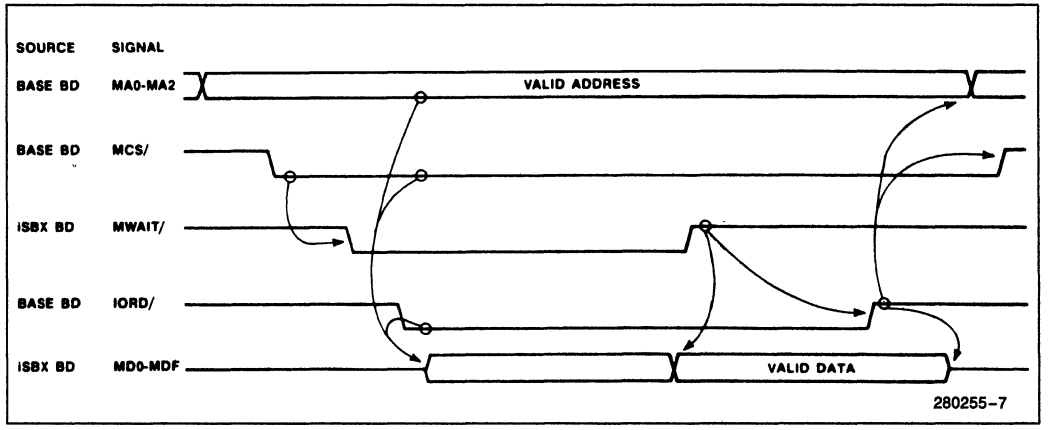


Figure 6. ISBX™ MULTIMODULE™ Board Extended Read

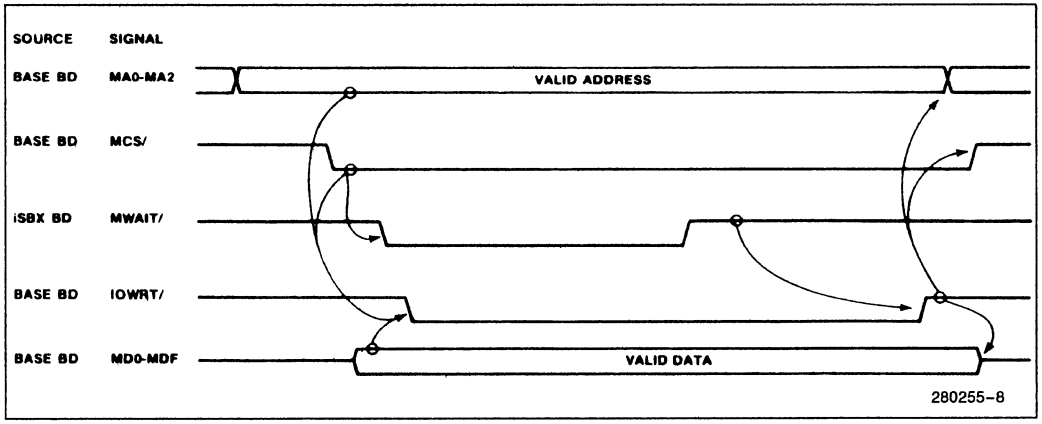


Figure 7. iSBC® MULTIMODULE™ Board Extended Write

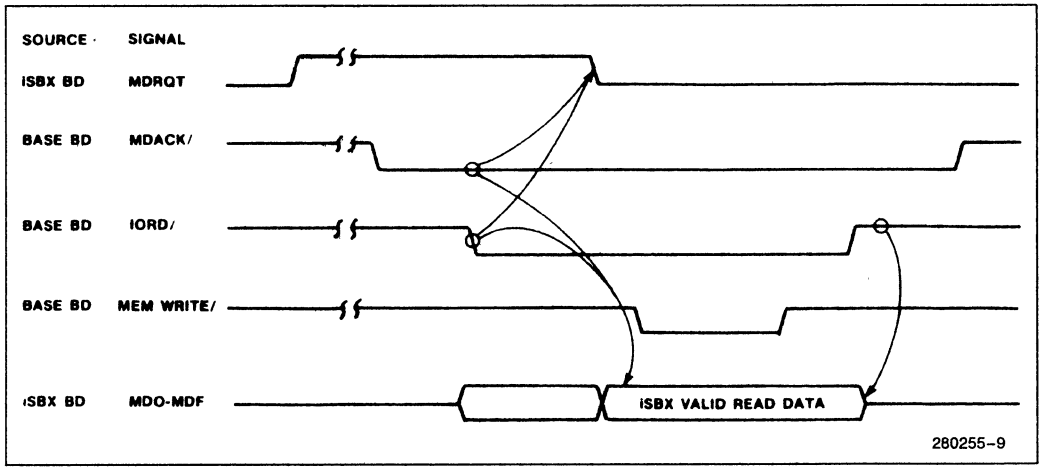


Figure 8. iSBX™ MULTIMODULE™ Board DMA Cycle (iSBX™ MULTIMODULE™ to Base Board Memory)

Board Outlines

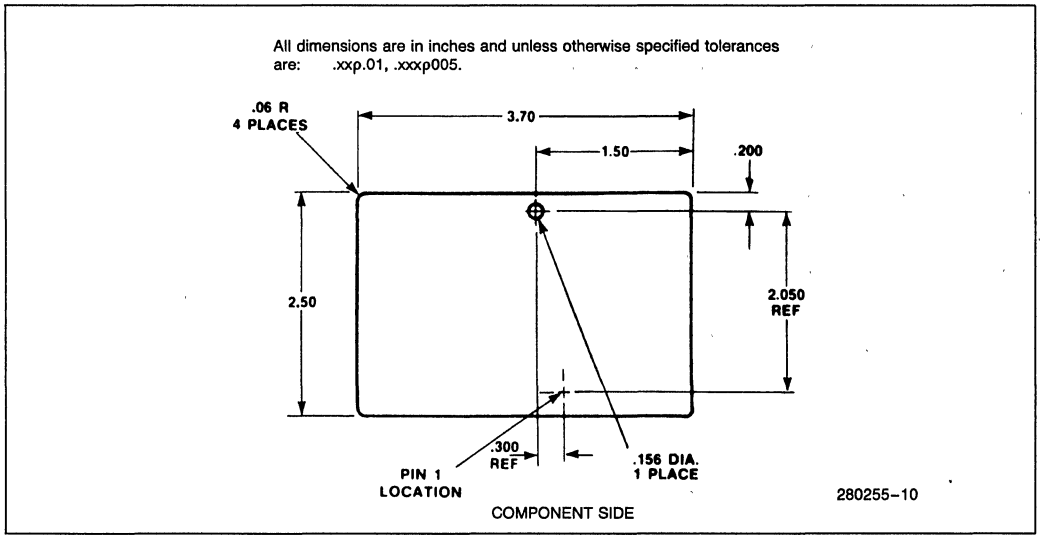


Figure 9. iSBX™ Board Outline

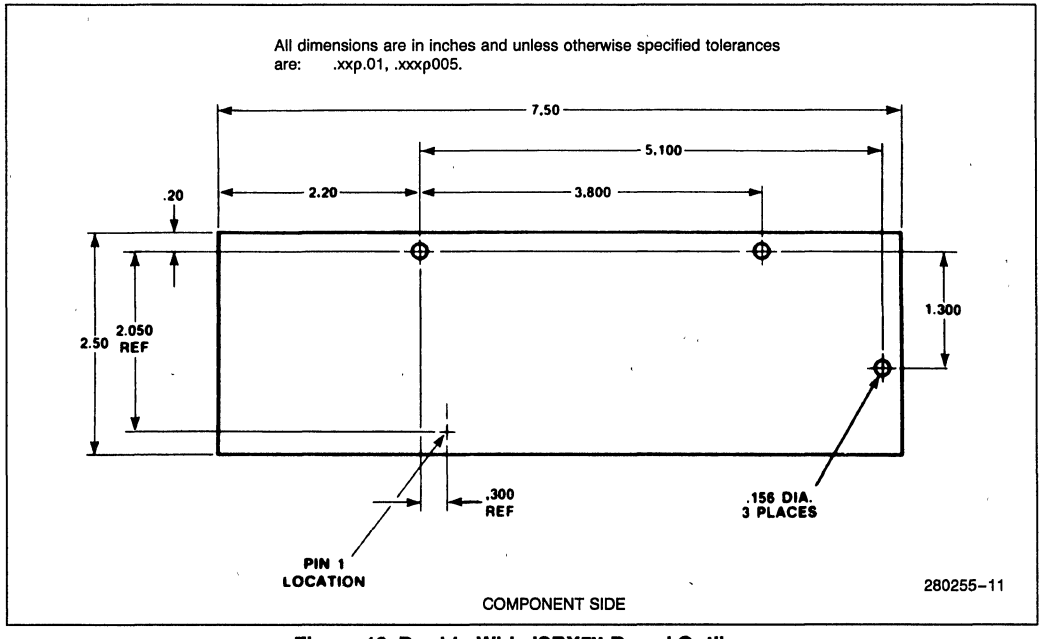


Figure 10. Double Wide iSBX™ Board Outline

Environmental Characteristics

Operating Temperature: 0°C to 55°C

Humidity: 90% maximum relative; non-condensing

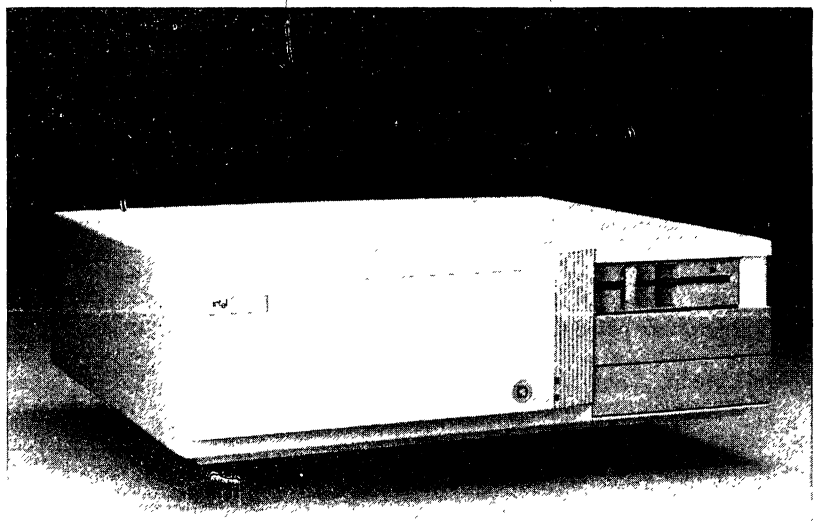
Reference Manuals

210883-002—MULTIBUS Architecture Reference Book

Real-Time Systems and Software

7

iRMX® SYSTEM 120



LOW-COST REAL-TIME 386™ SYSTEMS FROM INTEL

The Intel System 120 delivers real-time capability to users demanding a low-cost system for running time-critical applications, such as high-speed switching, process control, and data acquisition. The System 120 combines the rich functionality of the world's most popular Real-Time Operating System with the power and speed of the 386™ 32-bit microprocessor.

The System 120 makes available the ability to host, on a standard computing platform, real-time applications that have previously been impractical with other AT-Bus systems.

Applications developed for the System 120 can be moved easily to Intel's complete line of MULTIBUS® I and MULTIBUS II products, giving the user a broad spectrum of price, performance, and functions from which to choose.

FEATURES

- iRMX II: a complete real-time operating system; more than a kernel
- Intel 386™ 16 MHz and 25 MHz AT-Bus systems
- Development platform for iRMX® applications
- Easy migration of applications to and from MULTIBUS systems
- I/O expansion for PC-AT* and PC boards
- iRMX II to DOS file exchange
- OpenNET™ networking support
- 387™ numeric coprocessor support

The 25-MHz System 120 is shown at the top of the page
The 16-MHz System 120 is shown here as an iRMX® development platform with the addition of a monitor, video adapter, and keyboard.

September, 1989
Order Number 280649-003

iRMX II: A COMPLETE REAL-TIME OPERATING SYSTEM

iRMX system software is used in more real-time designs than any other operating system. There are over 500,000 CPUs worldwide running the iRMX Operating System, making it the most widely accepted standard real-time operating system for microprocessor-based designs.

The iRMX II Operating System provides a rich set of real-time programming facilities not found in general-purpose operating systems such as DOS, OS/2** or UNIX**. These include:

- Pre-emptive, dynamic priority-based scheduling of application tasks
- Bounded interrupt latency
- Multitasking support for real-time applications
- Inter-task communications through priority-based mailboxes, semaphores, and regions
- Interrupt management with exception handling
- Cross or on-target development

The iRMX II Operating System also offers high performance and code integrity. iRMX typically responds 100 times faster than general-purpose operating systems, enabling real-time applications to keep up with the rapid data and control flow of machine and communications interfaces. Code integrity is ensured through sophisticated memory protection schemes.

Finally, the iRMX II Operating System is highly configurable. Its modular design allows you to select only those functions and device drivers that are required. This keeps memory requirements to a minimum. Guided by the many examples in the System 120 Development Toolkit documentation, you can add custom device drivers and applications to the iRMX Operating System.

LOW COST PC-AT BASED CONFIGURATIONS

The System 120 target models are available with a number of processor speed, memory, and mass storage options to fit a range of applications. These include a basic system with 8 open slots, and a 40M-byte hard disk system with a 387 math coprocessor and floppy disk (see Table 1).

The System 120 is also available as a board-level product. (Contact your local Intel office for special ordering instructions.)

Intel offers PC-AT add-in boards for the System 120 that include: 2M-byte and 8M-byte 32-bit memory boards, the OpenNet PCLINK2 networking board and the iPCX 344A BITBUS™ board. A standard keyboard is also available.

Table 1: System 120 Target Configurations

PRODUCT CODE	CPU		RAM*	FLOPPY DISK	HARD DISK
	386	387	M-bytes	1-2 M-bytes	40 M-bytes
SYP12016Z0	16 MHz		2		
SYP12016Z40	16 MHz	16 MHz	2	✓	✓
SYP120Z5X0	25 MHz		4		
SYP120Z5M40	25 MHz	25 MHz	4	✓	✓

*Subtract 384 K-bytes for RAM available to iRMX® Operating System.

The iRMX software supports a range of popular display adapters, disk and tape controllers, and a four/eight-channel I/O controller. Additional drivers are available from third parties and from Intel's field systems engineers.

EASY APPLICATION DEVELOPMENT

You can develop applications for AT bus, MULTIBUS I and MULTIBUS II directly on the system using the System 120 Development Toolkit. In addition to the iRMX II Operating System for the System 120, the toolkit contains: PL/M 286 Compiler, ASM assembler, AEDIT and a source level debugger, Soft-Scope* II and an interface manager (JAM) that can be used as a front end to applications. Intel also offers a number of compilers (C, Fortran, Pascal), performance and debug tools for iRMX II.

Table 2: System 120 Development Toolkits

PRODUCT CODE	SOFTWARE		CPU	RAM	DISK	
	iRMX®II	PL/M 286 ASM AEDIT SOFTSCOPE JAM	386 & 387	Mb	FLOPPY	HARD
					1.2Mb	40Mb
SYS120KITZ40	✓	✓	16 MHz	2	✓	✓
SYS120KITM40	✓	✓	25 MHz	4	✓	✓
SYS120RMXZ40	✓		16 MHz	2	✓	✓
SYS120RMXM40	✓		25 MHz	4	✓	✓

APPLICATION MIGRATION TO HIGHER PERFORMANCE SYSTEMS

Applications written for the System 120 can be easily moved to the higher performance and functionality of MULTIBUS I and MULTIBUS II designs. That's because the System 120 iRMX II Operating System is binary compatible with the MULTIBUS implementation of iRMX II.

The iRMX II Operating System spans the entire Intel systems product line, from the low-cost System 120 through the MULTIBUS I System 320, to the high-end, multi-processing MULTIBUS II System 520. Applications can easily be re-hosted on different bus architectures, allowing you to create a group of products satisfying a wide range of customer performance requirements.

DOS APPLICATION COMPATIBILITY

The System 120 supports the DOS 3.X operating system as well as iRMX II, enabling you to use popular DOS applications to process data collected in real time. Some common applications are already available from software vendors. The applications include data bases, menu systems, and device drivers. The System 120 hard disk can be divided into iRMX and DOS partitions, allowing users to boot from either partition. A System 120 utility allows transfer of iRMX files into a DOS environment. DOS execution requires a customer-supplied version of DOS, a video adapter, a monitor, and a keyboard.

WORLDWIDE SERVICE AND SUPPORT

The System 120 is fully supported by Intel's worldwide staff of trained hardware and software support engineers. Intel also provides field application assistance, extensive iRMX Operating System classes, maintenance services, and a help hotline.

Among many services, Intel's systems engineers can implement special user needs, such as new device drivers.

The System 120 Development Toolkit comes with a 90-day software warranty and a one-year hardware warranty. System 120 target units come with a one year hardware warranty. Other support packages are optionally available; for more information please contact your local Intel Sales Office.

INTEL QUALITY AND RELIABILITY

The System 120 is designed to meet the high standards of quality and reliability that users have come to expect from Intel products. The iRMX II Operating System software has undergone thousands of hours of testing and evaluation and is one of the most stable operating systems in the industry today.

SPECIFICATIONS

SYSTEM 120 16-MHZ BASE SYSTEM

Central Processor	Intel 386, 16 MHz
Floating-Point Coprocessor	Intel 387, 16 MHz**
Main Memory	2M Bytes on CPU Board
Maximum RAM	16M Bytes
Cycle Time	120 ns
Data Bus Width	32-Bits
Error Detection	Byte Parity
I/O	1 serial port (asynchronous, RS232C, 9-pin connector) 1 parallel port (Centronics compatible, 25-pin connector)
8 expansion slots:	2 32- or 8-bit slots 4 16-bit slots 2 8-bit slots

ELECTRICAL

AC Voltage/Frequency	Switching power supply, 115 V/60Hz or 230 V/50 Hz
DC Power	220 Watts
+5V	23.0 A maximum continuous
+12V	8.0 A maximum continuous; 12.0 A maximum for 12 seconds
−12V	0.5 A maximum continuous
−5V	0.5 A maximum continuous

DIMENSIONS

Length	439 millimeters (17.3 inches)
Width	541 millimeters (21.3 inches)
Height	165 millimeters (6.5 inches)

WEIGHT

Base System:	20 kilograms (44 lbs)

SYSTEM 120 25-MHZ BASE SYSTEM

Central Processor	Intel 386, 25 MHz
Floating-Point Coprocessor	Intel 387, 25 MHz**
Main Memory	4M Bytes on CPU Board
Maximum RAM	24M Bytes
Cycle Time	80 ns
Data Bus Width	32-Bits
Error Detection	Byte Parity
I/O	2 serial ports (asynchronous, RS232C, 9-pin connector) 1 parallel port (Centronics compatible, 25-pin connector)
8 expansion slots	2 32-, 16- or 8-bit slots 5 16- or 8-bit slots 1 8-bit slot

ELECTRICAL

AC Voltage/Frequency	Switching power supply, 115 V/60Hz or 230 V/50 Hz
DC Power	220 Watts
+5V	23.0 A maximum continuous
+12V	8.0 A maximum continuous; 12.0 A maximum for 12 seconds
−12V	0.5 A maximum continuous
−5V	0.5 A maximum continuous

DIMENSIONS

Length	475 millimeters (18.7 inches)
Width	541 millimeters (21.3 inches)
Height	165 millimeters (6.5 inches)

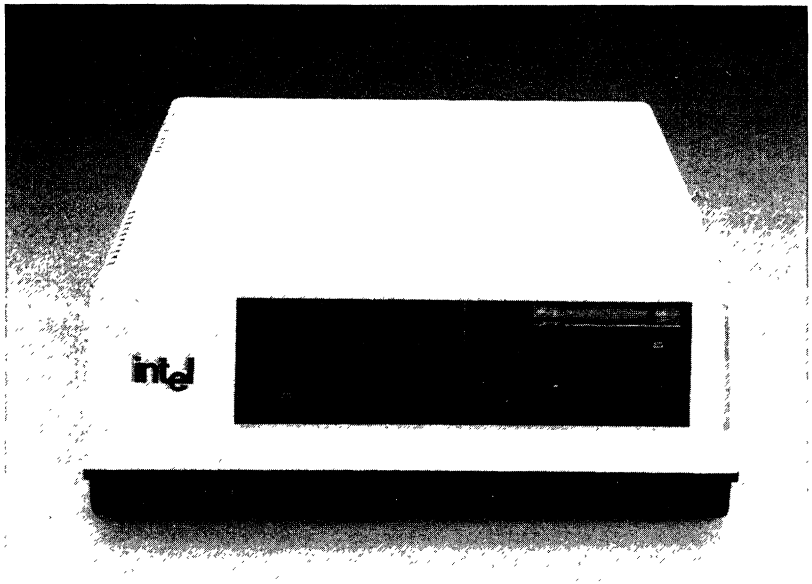
WEIGHT

Base System:	20 kilograms (44 lbs)

**See tables 1 and 2 for configurations that apply

ORDERING INFORMATION

For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada).



SYSTEM 310 AP

The System 310 AP is faster than many minicomputers. Powerful dedicated processors for communications and mass storage input/output control allow the 8 MHz 80286 CPU to concentrate on application software. The System 310 AP is open, which means you can upgrade performance and/or functionality in the future without purchasing a new system. The open system design protects your investment from becoming obsolete. Open systems design also means easy system customization with Intel and third-party add-in MULTIBUS® boards.

FEATURES

- 80286 Based System
- Open System MULTIBUS architecture for upgradeability and growth
- iRMX® Operating System
- OpenNET™ Local Area Networking
- Total hardware and software support from Intel's worldwide customer support organization

* The System 310 AP is also manufactured under product code pSYS310 by Intel Puerto Rico, Inc

September, 1989
Order Number 280129-005

FEATURES

SYSTEM 310 AP—AN OPEN SYSTEM

The Intel System 310 AP is based on the MULTIBUS architecture, (IEEE 796) industry standard system bus supported by over 200 vendors providing over 2000 compatible products.

The System 310 AP is an 80286 based open system designed with expansion in mind. The system can be expanded to accommodate up to 9MB of parity-checked RAM, all accessible with no wait states across MULTIBUS's Local Bus Extension (LBX™). For terminal communications, the systems can be expanded to a total of 18 RS232 serial ports.

The System 310 AP supports 40MB-140MB of Winchester disk storage. The 310 AP also supports a 320KB 5″ floppy drive and a 60MB streaming tape cartridge drive.

iRMX® OPERATING SYSTEM

The iRMX operating system delivers real-time performance. Designed to manage and extend the resources of the System 310 AP, this multitasking operating system provides configurable resources ranging from interrupt management and standard device drivers to data file maintenance commands for human interface and program development.

A wide range of popular industry standard high-level languages are supported for application development. The iRMX facilities also include powerful utilities for easy, interactive configuration and debugging.

OpenNET™—NETWORKING CAPABILITY

Intel's OpenNET product family provides a complete set of networking software and hardware that follows the International Standards Organization (ISO) Open Systems Interconnect (OSI) model.

OpenNET Network File Access Protocol adheres to the IBM/Microsoft/Intel Core File Sharing Protocol specification, providing transparent local/remote file access and file transfer capability between Intel's complete line of systems products, as well as with MSNET* and VAX/VMS*# based systems.

The System 310 AP distributes the transport protocol processing to intelligent Ethernet controllers that host Intel's OSI-compliant iNA 960 Class 4 Transport software, thereby unburdening the system CPU for greater performance.

INTEL SERVICE AND SUPPORT

The System 310 AP is backed by Intel's worldwide service and support organization. Total hardware and software support is available, including a hotline number for when you need help fast.

SPECIFICATIONS

SYSTEM/MODELS	310 AP-40B	310 AP-41B	310 AP-42	310 AP-82B	310 AP-142
Microprocessor	80286 8 MHz	80286 8 MHz	80286 8 MHz	80286 8 MHz	80286 8 MHz
Numeric Coprocessor	80287	80287	80287	80287	80287
RAM Memory	1MB	1MB	1MB	2MB	2MB
Floppy	360KB	360KB	360KB	360KB	360KB
Mass Storage	40MB	40MB	40MB	85MB	140MB
Tape Backup	NA	NA	60MB	60MB	60MB
Serial I/O Ports	2	10	10	14	10
Parallel Ports	1	1	1	1	1
OpenNET					

ENVIRONMENT

Operating Temperature	10°C to 35°C
Wet Bulb Temperature	26°C maximum
Relative Humidity	20% to 70% noncondensing
Altitude	Sea level to 8,000 feet

REGULATIONS

Meets or exceeds the following requirements:

Safety	
US	UL114
Canada	CSA C22.2
Europe	TUV ICE 435
EMI/RFI	
US and Canada	FCC Docket 20780—Class A
Europe	VDE 0871 Class A

ELECTRICAL

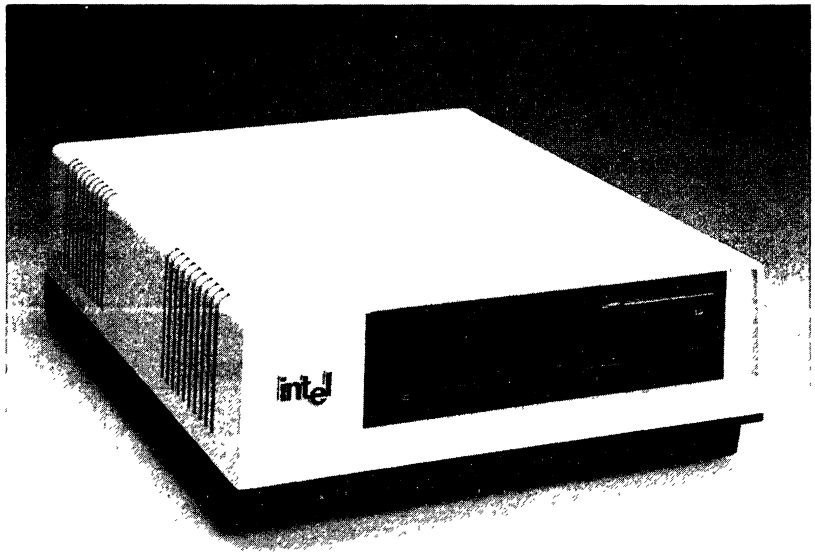
DC Power Output	360 watt maximum
AC Power Input	88-132 VAC or 180-264 VAC, 47-63 Hz (user selectable)

DIMENSIONS

Height	6½″
Width	17″
Depth	22″
Weight	Approx. 55 lbs

* MSNET is a trademark of Microsoft
* VAX/VMS is a trademark of Digital Equipment Corporation

SYSTEM 310* 386 UPGRADE



Intel's System 310 386 Upgrade offers the user an upgrade path to the performance of the 386™ microprocessor without sacrificing existing software and hardware investments. This Customer Service Installed upgrade is designed for the System 310 and System 310AP series of microcomputer systems using iRMX II operating system.

STANDARD FEATURES

- 16 MHz 386™ Microprocessor
- 16 Bit 80287 Numeric Data Processor
- Memory Options:
  1, 2, 4 and 8 MB
  0 wait state RAM
- Systems Confidence Test
  (SCT) and boot firmware
- Installed by Intel Customer Service at
  your location

* The iSXM386 kit is also manufactured under product code piSXM386 kit by Intel Puerto Rico, Inc

September, 1989
Order Number 280705-001

SPECIFICATIONS

The iSXM™ 386 Kit is designed to meet certain UL, FCC, CSA, IEC and VDE requirements when it is installed into an INTEL System 310 and System 310AP. It is the responsibility of the customer to reconfirm that the specific systems they have created from MULTIBUS elements continue to meet the required safety and environmental specifications in the customer environment. Intel is not responsible for any changes made after the product is accepted by Intel's customer.

SAFETY REQUIREMENT/EMI LIMITS

The iSXM 386 Kit is designed to meet:

Safety:

- UL 478 5th edition
- CSA C22.2 no. 154
- TUV IEC435 and VDE 0806

RMI/EMI:

- FCC 47 CFR Part 15 Subpart J Class A
- VDE 0871 Level A

Actual compliance will depend on the modules, peripherals and cable connectors which you install in the system.

ELECTRICAL

Voltage and Maximum Current:

iSXM 386 KIT-1, 1 MB Memory

± 5 VDC	± 5%	12.5 amps
+ 12 VDC	± 5%	0.025 amps
− 12 VDC	± 5%	0.025 amps

2 MB Memory add .3 amps @5 VDC
4 MB Memory add .0 amps @5 VDC
8 MB Memory add .3 amps @5 VDC

BASE REQUIREMENTS

You must have a current copy of iRMX II Release 2.0 or later installed on your system before the system can be upgraded. The −4 and −8 kits are recommended for use on 80 MB or 140 MB Winchester based systems only.

ORDERING INFORMATION

Your memory requirements will determine the product order code:

Memory Requirement	Order Code
1 MB RAM	iSXM386KIT-1
2 MB RAM	iSXM386KIT-2
4 MB RAM	iSXM386KIT-4
8 MB RAM	iSXM386KIT-8

SYSTEM SOFTWARE

iRMX II Languages:
FORTRAN 286, C286
Assembler 286, PL/M 286

Intel believes that the information in this document is accurate as of its publication date. Such information is subject to change without notice. Intel is not responsible for any inadvertent errors.

iRMX® SYSTEM 320

Intel combines the power of its high performance 386™ microprocessor-based System 320, the widely used iRMX II real-time software, complete network service software and comprehensive customer support capabilities to deliver, install and maintain a complete system. The result is the iRMX System 320 gives you the performance and capabilities of a minicomputer at less than half the cost. The system is especially suited for applications requiring real-time response and resource control typically found in financial transaction, industrial automation, medical and communications markets. The iRMX System 320 is also appropriate as the development environment for module-based design.

iRMX® SYSTEM 320 FEATURES

- 80386 Based System
- iRMX Real-time Multitasking Operating System
- Open System Architecture
- OpenNET Local Area Networking
- Complete Installation, Service and Support
- Worldwide User Group Support
- Range of Configurations

intel

* The System 320 is also manufactured under product code pSYS320 by Intel Puerto Rico, Inc

September, 1989
Order Number 280502-002

FEATURES

iRMX® II—REAL-TIME SOFTWARE

The iRMX II operating system delivers real-time performance. Designed to manage and extend the resources of the System 320, this multitasking operating system provides configurable resources ranging from interrupt management and standard device drivers to data file maintenance commands for human interface and program development. The iRMX II facilities also include powerful utilities for easy, interactive configuration and debugging.

SYSTEM 320—AN OPEN SYSTEM

The iRMX System 320 is based on MULTIBUS architecture, (IEEE 796) industry standard system bus supported by over 200 vendors providing over 2000 compatible products, and on the iRMX II operating system composed of modular layers, highly configurable for tailoring to target applications. A wide range of popular industry standard high-level languages are supported for application development. Special configurations can be tailored by the user, by Intel's Custom System Integration group or by Intel's authorized Value Added Distribution Centers.

OpenNET™ NETWORKING CAPABILITY

Intel's OpenNET product family provides a complete set of networking software and hardware that follows the International Standards Organization (ISO) Open Systems Interconnect (OSI) model.

OpenNET Network File Access Protocol adheres to the IBM/Microsoft/Intel Core File Sharing Protocol specification, providing transparent local/remote file access and file transfer capability between Intel's complete line of systems products, as well as with MSNET* and VAX/VMS*# based systems.

The System 320 distributes the transport protocol processing to intelligent Ethernet controllers that host Intel's OSI-compliant iNA 960 Class 4 Transport software, thereby unburdening the system CPU for greater performance.

INSTALLATION SERVICE & SUPPORT

The Intel iRMX System 320 is backed by Intel's worldwide service and support organization. Installation is available to quickly get the system up and running. Total hardware and software support is available, including a hotline number for when the user needs help fast. Intel also provides hands-on training workshops to give the user a thorough understanding of the iRMX System 320. These workshops are conducted at Intel training centers or customer sites worldwide.

WORLDWIDE USER GROUP SUPPORT

iRUG (iRMX User Group), provides members a user's library of iRMX software tools and utilities, access to the group bulletin board, receipt of regularly published newsletters and invitations to User Group Conferences. iRUG numbers over 42 local chapters in 20 countries worldwide.

RANGE OF CONFIGURATIONS

Intel offers a wide range of configurations for the iRMX System 320. Contact your local Intel representative for further information.

* MSNET is a trademark of Microsoft
** VAX/VMS is a trademark of Digital Equipment Corporation

SPECIFICATIONS

ENVIRONMENT

Operating Temperature	10°C to 40°C
Wet Bulb Temperature	26°C maximum
Relative Humidity	85% at 40°C
Altitude	Sea level to 10,000 feet

REGULATIONS

Meets or exceeds the following requirements:

Safety	
US	UL 478
Canada	CSA C22.2
Europe	IEC 435
EMI/RFI	
US and Canada	FCC Class B Computing Device
Europe	VDE Limit Class B

ELECTRICAL

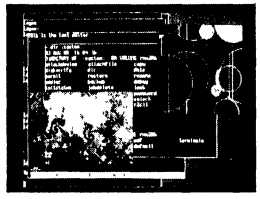
DC Power Output	435 watt maximum
AC Power Input	88-132 VAC or 176-264 VAC, 47-63 Hz, single phase

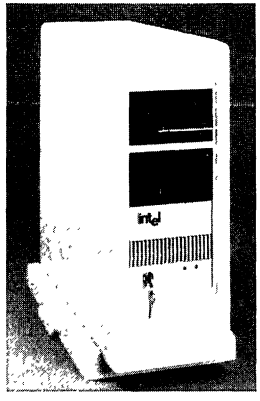
DIMENSIONS

Height	8″
Width	17.5″
Depth	22.25″
Weight	Approx. 55 lbs

ORDERING INFORMATION

For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada).





UNLOCK THE POWER OF MULTIBUS®II WITH AN INTEL® OEM SYSTEM

The Intel System 520 and MULTIBUS II System Architecture (MSA) make it easy to unlock the power of MULTIBUS II. The System 520 is the first in a family of high performance, real-time OEM systems to combine Intel's open MSA architecture, the powerful 386™ microprocessor, and UNIX* System V or the industry-leading iRMX®II real-time multitasking operating system. Together, they provide an easily scalable, recomposable open bus system.

As an open OEM system, the System 520 allows users to add to the basic system, or purchase the system's contents separately and repackage them into another enclosure. Intel's MSA provides this capability via a structured set of open, standard interfaces and protocols that build on and are fully compatible with the MULTIBUS II (IEEE 1296) bus standard. As a result, the System 520 provides new standards of ease of integration, ease of use, and board compatibility for the OEM.

FEATURES

- 386 CPU-based performance
- Supports System V/386 UNIX or iRMX II real-time operating system
- Easy 386 application processor expansion (1 to 4)
- High performance SCSI I/O subsystem
- OpenNET™ transparent remote file sharing & virtual terminal between 386 processors and IEEE 802.3 networked systems
- Hardware windowed graphics and virtual terminal support
- iRMX II and System V/386 UNIX* development systems available

October, 1989
Order Number 280672-002

THE MULTIBUS®II SYSTEM ARCHITECTURE (MSA)

The System 520 is built around Intel's MULTIBUS II System Architecture (MSA) to ease the development and integration work of MULTIBUS II OEM system designers. Intel's MSA delivers open system interface and protocol standards that build on and extend the basic MULTIBUS II (IEEE/ANSI 1296) bus standard. The MSA specifications define diagnostics, built-in self test, system initialization and boot loading, board configuration, and message passing. The user benefits from MSA because the level of vendor and board compatibility has been raised above basic electrical bus specifications to a set of powerful programmatic interfaces that handle the bus specifics with software. This provides OEMs quicker time to market with faster system integration and shorter design cycles.

EASY USER EXPANSION AND RE-COMPOSABLE SYSTEMS

The MULTIBUS II System Architecture is used as the foundation for integrating Intel's full line of Single Board Computer (iSBC®) modules. OEMs have the option to buy the contents of the system (i.e. the boards, the firmware, the software, etc.) separately, and re-compose all or part of the system's pieces into a different configuration or enclosure.

The system is available in a series of hardware configurations. As hardware-only platforms, they may be purchased without an operating system by OEMs requiring a MULTIBUS II System 520 to host their own operating system software.

Easy user expansion and re-composability of the System 520 is supported by a line of System Integration Toolkits (SIT kits) that contain all the firmware necessary to allow standard, off-the-shelf MULTIBUS II boards to integrate cleanly into the System 520. With these toolkit products, the OEM can purchase the pieces needed and profit from greater ease of use and ease of integration, and higher levels of open standards.

WINDOWED GRAPHICS AND VIRTUAL TERMINAL SUBSYSTEM

The System 520 hosts the iRMX® Graphics Interface driven by Intel's iSBX™ 279 Graphics Board. The graphics subsystem and its companion software provide a windowed virtual terminal console with graphics capabilities. With its onboard processing power and large graphics memory buffer, the iSBX 279 off-loads the application processors of the display processing tasks. The user interfaces to the System 520's subsystem are an RGB color monitor (640 × 480), a mouse, and an AT-style keyboard (purchased separately).

A NETWORK IN THE SYSTEM

Using the MULTIBUS II backplane as an ultra-fast network (40MBytes/sec), multiple peer-to-peer 386™ CPU-based application processors operate as independent "networked" iRMX or UNIX systems over the MULTIBUS II Parallel System Bus (PSB). Each application processor running Intel's OpenNET™ network software will provide transparent distributed file sharing, file transfer, and virtual terminal capability among all application processors on the backplane—and among IEEE 802.3-based (1.25MBytes/sec) OpenNET networked system nodes. Using the PSB as a network makes the System 520 a high performance "minicomputer" cluster condensed into one multiprocessor system.

The major advantage to the OEM is total network extensibility inside and outside the system, using the same OpenNET software. The key benefits are: reduced cost compared to multiple uni-processor networked systems and servers, drastically reduced physical space requirements, increased overall network throughput and performance, and preserved software investment.

HIGH PERFORMANCE 386™ CPU-BASED I/O SUBSYSTEM

The System 520 uses the Intel iSBC 386/258 SCSI Peripheral Controller to provide minicomputer level I/O performance to SCSI (Small Computer System Interface) mass storage devices. I/O critical applications are accelerated by the combination of a 386 processor, a large (1-4MBytes) data buffer for cacheing, and a 4 MByte per second SCSI synchronous transfer rate. This flexible I/O subsystem can be used as an intelligent disk controller, or a hybrid iRMX II application processor and file server, through soft-loaded firmware.

UNIX* OPERATING SYSTEM SUPPORT FOR THE SYSTEM 520

The System V/386 UNIX operating system provides portability of applications and systems programs from one hardware architecture to another. Integrated with System 520, the UNIX operating system provides a powerful open system platform. The System V/386 product family is ideal as a base for developing custom multi-user systems applications, and multi-purpose network servers requiring open system configurability and flexible packaging.

The System V/386 UNIX operating system delivers full support for Intel's MULTIBUS II System Architecture. Intel has built a complete System V/386 product family providing OEMs, system integrators and computing manufacturers with industry standard UNIX, OpenNET™ networking, system hardware, and Ada development tools for the MULTIBUS II System Architecture. Together, these elements deliver a rich, complete UNIX development environment.

FEATURES

iRMX® II: A FULL FEATURED REAL-TIME OPERATING SYSTEM FOR MULTIBUS®II

Intel's iRMX II Real-Time Multitasking Operating System is a full featured, stand alone operating environment, designed to address the complete range of real-time applications, from embedded control designs to reprogrammable MULTIBUS II multiprocessor systems. It provides complete MULTIBUS II facilities supporting MULTIBUS II Transport message passing and Interconnect space access. Using iRMX II software, engineers can assemble a powerful, cluster of application processors into a single, integrated multiprocessor system. The iRMX II system software manages all message transmission and reception, making the construction of real-time multiprocessor systems easier.

The iRMX II Operating System provides a rich set of real-time programming facilities not found in general-purpose operating systems. Some of its key features include pre-emptive, dynamic priority-based scheduling of application tasks; bounded interrupt latency; multitasking support; inter-task communications and synchronization through priority-based mailboxes, semaphores, and regions, and interrupt management with exception handling. By combining these features with a modular design; quick response; and sophisticated memory protection schemes, the OEM receives a highly configurable, customizable operating system with the high performance and code integrity that real-time applications require.

COMPLETE MULTIBUS®II DEVELOPMENT ENVIRONMENT

The System 520 is also available as a bundled, complete, networked development system for MULTIBUS II modules development, software development and testing of real-time applications. The System 520 is unique as a MULTIBUS II development system, because its multiprocessor cluster capability can support both on-target or cross-hosted system and software development in one chassis.

OpenNET™ NETWORKING CAPABILITY

Intel's OpenNET product family provides a complete set of networking software and hardware that follows the International Standards Organization (ISO) Open Systems Interconnect (OSI) seven layer model. The System distributes the ISO/OSI transport protocol processing to intelligent Ethernet controllers hosting Intel's OSI compliant iNA 960 Class 4 Transport software. Intel's OpenNET Network File Access (NFA) protocol provides the upper layer functionality of transparent local or remote file access and file transfer between Intel's complete line of system products, as well as MS-DOS Operating System-based personal computers and VAX/VMS* minicomputers. The OpenNet NFA protocol adheres to the standard IBM*/Microsoft*/Intel Core File Sharing protocol specification.

WORLDWIDE SERVICE AND SUPPORT

The System 520 is fully supported by Intel's worldwide staff of trained hardware and software support engineers. Intel also provides field application assistance, extensive operating system classes, maintenance services, and a help hotline.

The System 520 OEM System products come with a standard 90-day hardware warranty. The System 520 MULTIBUS II Development System products come bundled with a one (1) year service warranty. This one year warranty includes: hardware installation and one year of on-site maintenance, software installation of the operating system and 48 hours of phone support.

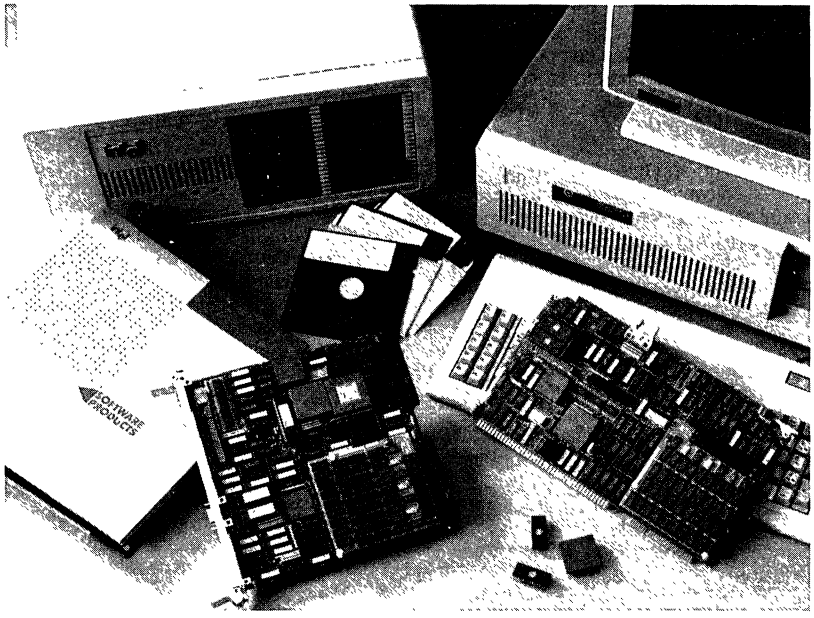
INTEL QUALITY AND RELIABILITY

The System 520 is designed to meet the high standards and reliability that users have come to expect from Intel products.

SPECIFICATIONS

SYSTEM 520 CONFIGURATION/OPTIONS— TABLE 1

Product	iSBC 386/120 4 MB	iSBC 386/258, 4 MB, CSM 002	iSBX 279 Graphics	iSBC 186/410 Terminal Controller	iSBC 186/530 Ethernet Controller	380 MB Hard Disk (SCSI)	Hard Disk (SCSI) Add-in Options	125/150 MB ¼" Tape (SCSI)	1.2 MB Floppy and Adapter (SCSI)	Floor Stand Chassis Sta	Table Top Chassis (option)	iRMX II Development Software	System V/386 Software
System 520 OEM Base Plus I/O	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
System 520 OEM Base without Tape		✓					✓		✓	✓	✓		
System 520 OEM Base with Tape		✓						✓	✓	✓			
iRMX System 520 Development System	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
System V/386 520 Development System	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓



A 32-BIT REAL-TIME KERNEL

The iRMK™ Version I.2 Real-time Kernel is the 32-bit real-time executive developed, sold, and supported by Intel, the Intel386™ architecture experts. It reduces the cost and risk of designing and maintaining software for numerous real-time applications such as embedded control systems and dedicated real-time subsystems in multiprocessor systems.

FEATURES

- 32-bit real-time multitasking kernel
- Rich set of real-time services
- Designed and optimized for the Intel386™ and Intel376™ families
- Extremely fast execution with predictable response times for time critical applications
- Compact design, as small as 8K bytes
- Multiprocessor support
- Requires only the 80386 or 80376; Provides optional support for 80387 and 80387SX Numeric Coprocessors and other peripheral devices
- Works with any bus including the MULTIBUS® I and MULTIBUS II architectures
- Optional MULTIBUS II message passing support provided
- Designed for easy customization and enhancement
- Easily programmed into PROMs or EPROMs
- Comprehensive development tool support
- Supported by Intel

September, 1989
Order Number 280613-002

REAL-TIME SOFTWARE FROM THE INDUSTRY LEADER

Intel has been the industry leader in microprocessor-based real-time computing since it invented the microprocessor. No other company supplies the range of real-time solutions that we do. Since 1977, thousands of customers have used our iRMX® real-time operating systems.

Now Intel has put its real-time expertise into a 32-bit kernel that supports the 80386 microprocessor and the 80376 embedded controller. The iRMK Version I.2 Kernel saves you the cost of designing, debugging, and maintaining your own executive for real-time systems. You can concentrate on writing your application rather than on writing a kernel.

THE QUICKEST PATH FOR A WIDE RANGE OF REAL-TIME APPLICATIONS

The iRMK Kernel's high performance and rich set of real-time services make it ideal for a wide range of real-time applications, including:

Data acquisition and analysis
Continuous process control
Discrete process control
Simulation
Medical instruments
Test instrumentation
Image processing
Automated test
Avionics and navigation
Field command control
Energy and environmental control
Radio control
Satellite communications
Terminals
Graphics work stations
Robotics
Signal processing
Laser printers
Front-end concentrators
Host communications

A RICH SET OF REAL-TIME SERVICES

The iRMK Version I.2 Kernel provides a rich set of services for real-time applications, including:

- Task management with system calls to create, manage, and schedule tasks in a multitasking environment. The Kernel offers pre-emptive priority scheduling combined with optional time-slice (round robin) scheduling.

  The scheduling algorithm used by the iRMK Kernel allows tasks to be rescheduled in a fixed amount of time regardless of the number of tasks. Applications may contain any number of tasks.

  An application can provide optional task handlers to customize task management. These handlers can execute on task creation, task switch, task deletion, and task priority change. Task handlers can be used for a wide range of functions including saving and restoring the state of coprocessor registers on task switch, masking interrupts based on task priority, or implementing statistical and diagnostic monitors.

- Interrupt management by immediately switching control to user-written interrupt handlers when an interrupt occurs. Response to interrupts is both fast and predictable. Most of the Kernel's system calls can be executed directly from interrupt handlers.

- Time management providing single-shot alarms, repetitive alarms, and a real-time clock. Alarms can be reset.

  These time management facilities can solve a wide range of real-time programming problems. Single-shot alarms, for example, can be used to handle timeouts. If the timeout occurs, the alarm invokes a user-written handler; if the event occurs before the timeout, the application simply deletes the alarm. Other uses for the Kernel's time management facilities include polling devices with repetitive alarms, putting tasks to sleep for specified periods of time, or implementing a time-of-day clock.

- Semaphores, regions, and mailboxes for intertask synchronization and communication. Semaphores are used for intertask signalling and synchronization. Regions are special binary semaphores used to ensure mutual exclusion and prevent deadlock when tasks contend for control of system resources. A task holding a region's unit runs at the priority of the highest priority task waiting for the region's unit.

  Mailboxes are queues that can hold any number of messages and are used to exchange data between tasks. Either data or pointers can be sent using mailboxes. The Kernel allows mailbox messages to be of any length. High priority messages can be placed (jammed) at the front of the message queue to ensure that they are received before other messages queued at the mailbox

  To ensure that high priority tasks are not blocked by lower priority tasks, the Kernel allows tasks to queue at semaphores and mailboxes in priority order. The Kernel also supports first-in, first-out task queuing.

- Memory pool manager that provides fixed and variable block allocation. Memory can be divided into any number of pools. Multiple memory pools might be created for different speed memories or for allocating different size blocks. Access to a memory pool for fixed-sized allocation is always deterministic.

  The Kernel-supplied memory manager works with flat, segmented, and paged addressing. Users can write their own memory manager to provide different memory management policies or to support virtual memory.

SUPPORT FOR MULTIPROCESSING VIA MULTIBUS® II ARCHITECTURE

The MULTIBUS II architecture is designed to optimize multiprocessor designs. This bus:

- Implements a loosely coupled architecture in which interprocessor interrupts and data are exchanged via messages transmitted as packets over the bus;
- Provides fast bus access;
- Allows interprocessor signalling at interrupt speeds from as many as 255 sources;
- Provides data transfer rates of up to 32 megabytes per second;
- Allows multiple communication sessions to occur simultaneously between processors;
- Supports up to 21 CPU boards per chassis with each board providing the processor, memory, and I/O needed for its portion of the application; and
- Provides registers—called Interconnect Space—on each board that can be used for dynamic system configuration.

Two optional modules allow iRMK Kernel applications to make full use of the MULTIBUS II architecture. The first module implements message passing allowing the application to have direct access to the message passing hardware or to use Intel's MULTIBUS II transport protocol. The second module implements interconnect space access to support dynamic system configuration.

These modules can be used to implement high performance multiprocessor designs that:

- Break a highly complex real-time application into multiple lower complexity applications distributed across multiple processors
- Distribute an application that's too CPU intensive for a single processor between several processors
- Provide redundancy
- Dedicate processors to specific tasks
- Provide interoperation with any operating system or controller board that uses Intel's MULTIBUS II transport protocol, including the iRMX™ II.3, iRMK I.2, and Intel System V/386 operating systems.

HARDWARE REQUIREMENTS AND SUPPORT

The iRMK Kernel requires only an 80386 microprocessor or an 80376 embedded controller and sufficient memory for itself and its application. Its design, however, recognizes that many systems use additional programmable peripheral devices and coprocessors. The Kernel provides optional device managers for:

- The 80387 and 80387SX Numeric Coprocessors
- The 82380 and 82370 Integrated System Peripherals
- The 8254 Programmable Interval Timer
- The 8259A Programmable Interrupt Controller

An application can supply managers for other devices and coprocessors in addition to or in replacement of the devices listed above.

The iRMK Kernel was designed to be programmed into PROM or EPROM, making it easy to use in embedded designs.

The iRMK Kernel can be used with any system bus including the MULTIBUS I and MULTIBUS II busses. The optional MULTIBUS II message passing and Interconnect Space access modules use the Message Passing Coprocessor (MPC). The Kernel provides managers to use the 82380/82370 Integrated System Peripherals or the 82258 Advanced DMA controller with the MPC for message passing.

SUPPORT FOR THE INTEL386™ AND INTEL376™ ARCHITECTURES

The iRMK Kernel provides 32-bit, protected mode 80386 and 80376 operation. By default, the Kernel and its application execute in a flat memory space of up to 4 gigabytes and in a single privilege level. Applications can add support for any mixture of additional protected mode features including:

- Any model of segmentation
- Memory paging
- Virtual memory
- Multiple privilege levels
- Call and trap gates

These protected mode features can be used to increase the reliability of the application by using the processor's hardware to:

- Protect against attempts to write beyond segment bounds (to catch, for example, situations like stack overflow or underflow)
- Allow only privileged or trusted code to access key routines and data
- Isolate bugs to single modules so that the rest of the application and the Kernel are not corrupted
- Assign access rights to code and data
- Isolate address spaces

To use these features, the application manipulates the processor's descriptor tables. Since the Kernel was designed specifically to support 80386 and 80376 applications, it provides an optional Descriptor Table manager that simplifies protected mode programming. This manager provides system calls to read and write descriptor table entries, to convert addresses from linear to physical and vice versa, and to get a segment's selector.

A MODULAR ARCHITECTURE FOR EASY CUSTOMIZATION

The iRMK Kernel was designed for maximum flexibility so it can be customized for each application. Each major function—mailboxes, for example—was implemented as a separate module. The Kernel's modules have not been linked together and are supplied individually. You link the modules you need for your application. Any module not used does not need to be linked in, and does not increase the size of the Kernel in your application. You can also replace any optional Kernel module with one that implements specific features required by your application. For example, you might want to replace the Kernel's memory manager with one that supports virtual memory.

Table 1 lists the Kernel's modules.

Table 1: iRMK™ Version I.2 Kernel Modules

Core Functions	Optional Modules	Optional Device Managers
• Task manager • Time manager • Interrupt manager	• Mailbox manager • Semaphore manager • Memory Pool manager • Descriptor Table manager • MULTIBUS II Message Passing • MULTIBUS II Interconnect Space Access	• 80387 & 80387SX • 82380 & 82370 • 8254 • 8259A

DEVELOPING WITH THE iRMK™ REAL-TIME KERNEL

iRMK Kernel applications can be written using any language or compiler that produces code that executes in the 80386's protected mode or on the 80376. This independence is achieved by using interface libraries. These libraries work with the idiosyncrasies of each language—for example, the ordering of parameters. The interface libraries translate the call provided by the language into a standard format expected by the Kernel. Intel provides interface libraries for our iC 386 and PL/M 386 languages. The source code for these libraries is provided so you can modify them to support other compilers.

Intel's 80386 Utilities are used to link the Kernel's modules and to locate the Kernel in memory. Applications written with a compiler that produces OMF386 object module format can be linked directly to the Kernel for the highest possible performance. Alternately, applications written in OMF386 or another object module format can access the Kernel through a call gate mechanism included with the Kernel.

Because the Kernel is supplied as unlinked object modules, applications can be developed on any system that hosts the development tools that you will use.

COMPREHENSIVE DEVELOPMENT TOOL SUPPORT

Intel provides a complete line of 80386 and 80376 development tools for writing and debugging iRMK Kernel applications. These tools include:

Software:	PL/M 386 Compiler iC 386 Compiler ASM 386 Assembler RLL 386 Utilities
Debuggers:	ICE™ 386 and ICE 376 P-MON 386 D-MON 386

These tools run on IBM* PC AT systems and compatibles running PC- or MS-DOS* 3.X. The languages and utilities also run on VAX/VMS and MicroVAX/VMS* systems. The iRMK Version I.2 Kernel software is available on IBM PC format 5¼ inch, 360K byte diskettes.

INTEL SUPPORT, CONSULTING, AND TRAINING

With the iRMK Kernel you get the Intel386 architecture and real-time expertise of Intel's customer support engineers. We provide phone support, on- or off-site consulting, troubleshooting guides, and updates. The Kernel includes 90 days of Intel's Technical Information Phone Service (TIPS). Extended support and consulting are also available.

CONTENTS OF THE iRMK™ KERNEL DEVELOPMENT PACKAGE

The iRMK Kernel comes in a comprehensive package that includes:

- Kernel object modules
- Source for the Kernel-supplied 82380 and 82370 Integrated System Peripherals; 8259A PIC; 8254 PIT; and 80387 and 80387SX Numeric Coprocessor device managers
- Source for PL/M 386 and iC 386 interface libraries
- Source for the call gate interface
- Source for sample applications showing:
  - —Structure of Kernel applications
  - —Use of the Kernel with application written in both PL/M 386 and iC 386
  - —Compile, bind, and build sequences
  - —Sample initialization code for the 80386 microprocessor
  - —MULTIBUS II message passing
  - —Applications written to execute in a flat memory space and in a segmented memory space
- User Reference Guide
- 90 days of Customer Support

* IBM is a registered trademark of the International Business Machines Corporation.
* MS-DOS is a trademark of Microsoft Corporation.
* VAX is a registered trademark of Digital Equipment Corporation. VMS is a trademark of Digital Equipment Corporation.

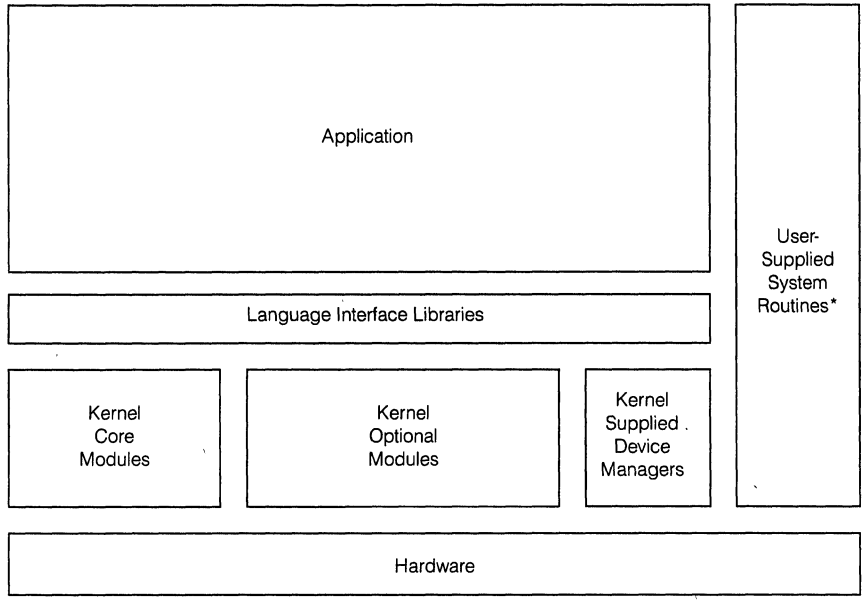


Figure 1: iRMK™ Version I.2 Real-Time Kernel Architecture
*User-supplied system routines would include interrupt handlers, user-written device managers, and similar routines.

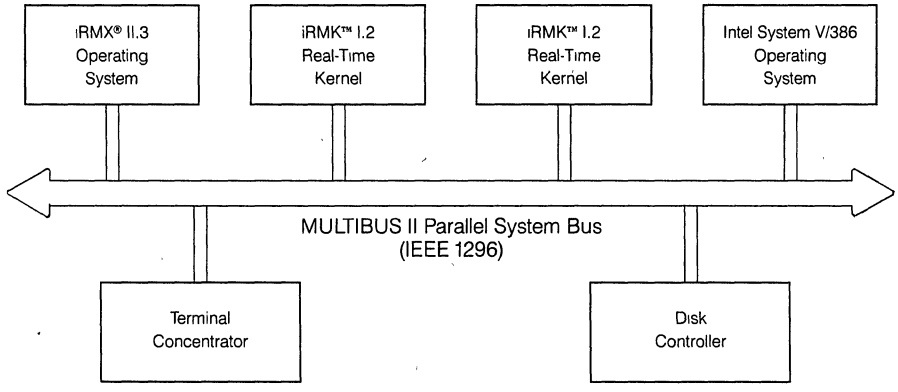


Figure 2: The optional MULTIBUS II message passing modules give the iRMK™ I.2 Version Kernel full multiprocessing capabilities for distributing applications among processors and interoperating with other operating systems.

iRMK™ VERSION I.2 KERNEL SYSTEM CALLS[1]

Kernel Initialization

KN_initialize	Initialize Kernel

Object Management

KN_token_to_ptr	Returns a pointer to area holding object
KN_current_task	Returns a token for the current task

Task Management

KN_create_task	Create a task
KN_delete_task	Delete a task
KN_suspend_task	Suspend a task
KN_resume_task	Resume a task
KN_set_priority	Change priority of a task
KN_get_priority	Return priority of a task

Interrupt Management

KN_set_interrupt	Specify interrupt handler
KN_stop_scheduling	Suspend task switching
KN_start_scheduling	Resume task switching

Time Management

KN_sleep	Put calling task to sleep
KN_create_alarm	Create and start virtual alarm clock
KN_reset_alarm	Reset an existing alarm
KN_delete_alarm	Delete alarm
KN_get_time	Get time
KN_set_time	Set time
KN_tick	Notify kernel that clock tick has occurred

Intertask Communication and Synchronization

KN_create_semaphore	Create a semaphore
KN_delete_semaphore	Delete a semaphore
KN_send_unit	Add a unit to a semaphore
KN_receive_unit	Receive a unit from a semaphore
KN_create_mailbox	Create a mailbox
KN_delete_mailbox	Delete a mailbox
KN_send_data	Send data to a mailbox
KN_send_priority_data	Place (jam) priority message at head of message queue
KN_receive_data	Request a message from a mailbox

Memory Management

KN_create_pool	Create a memory pool
KN_delete_pool	Delete a memory pool
KN_create_area	Create a memory area from a pool
KN_delete_area	Return a memory area to a memory pool
KN_get_pool_attributes	Get a memory pool's attributes

Descriptor Table Management

KN_get_descriptor_attributes	Get a descriptor's attributes
KN_set_descriptor_attributes	Set a descriptor's attributes
KN_initialize_LDT	Initialize local descriptor table (LDT)
KN_null_descriptor	Overwrite a descriptor with the null descriptor

[1] System calls Copyright© 1987, 1988 Intel Corporation.

KN_initialize_subsystem	Allows application to be divided into multiple subsystems when application interfaces to Kernel through a call gate
KN_linear_to_ptr	Convert a linear address to a pointer
KN_ptr_to_linear	Convert a pointer to a linear address
KN_get_data_selector	Get the selector for the data segment
KN_get_code_selector	Get the selector for the code segment
KN_translate_ptr	Converts a pointer that will be based on a user-specified selector

82380, 82370, and 8259A PIC Management

KN_initialize_PICs	Initialize the PICs
KN_mask_slot	Mask out interrupts on a specified slot
KN_unmask_slot	Unmask interrupts on a specified slot
KN_send_EOI	Signal the PIC that the interrupt on a specified slot has been serviced
KN_new_masks	Change interrupt masks
KN_get_slot	Return the most important active interrupt slot

82380, 82370, and 8254 PIT Management

KN_initialize_PIT	Initialize a PIT
KN_start_PIT	Start PIT counting
KN_get_PIT_interval	Return PIT interval

80387 and 80387SX Numeric Coprocessor Management

KN_initialize_NDP	Initialize an 80387 or 80387SX Numeric Coprocessor

MULTIBUS® II Message Passing Management

KN_initialize_message_passing	Initialize the message passing module
KN_mp_working_storage_size	Compute size of work space needed for message passing
KN_send_tp	Send a transport message
KN_attach_receive_mailbox	Attach a receive mailbox
KN_cancel_tp	Cancel a solicited message or request-response transaction
KN_send_dl	Send a data link message
KN_attach_protocol_handler	Attach a protocol handler
KN_cancel_dl	Cancel a data link buffer request

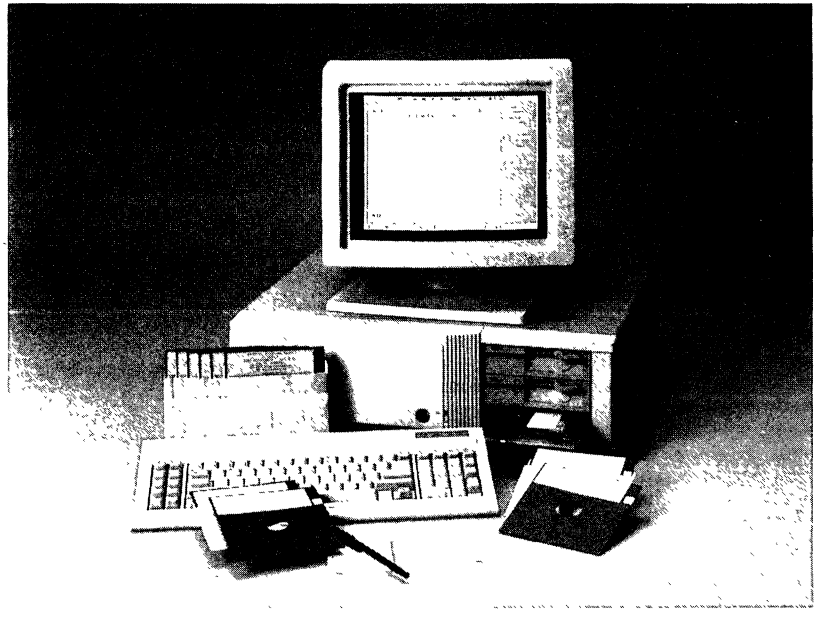
MULTIBUS® II Interconnect Space Management

KN_initialize_interconnect	Initialize the interconnect module
KN_get_interconnect	Get the value of an interconnect register
KN_set_interconnect	Set the value of an interconnect register
KN_local_host_ID	Get the host ID of the local host

ORDERING INFORMATION

Order Code	*Product*	*Contents*
RMK	iRMK Version I.2 Development Software	iRMK Version I.2 Kernel
RMKDEVP RMKDEVC	iRMK Version I.2 Developer's Kit	iRMK Version I.2 Kernel PL/M 386 or iC 386 Compilers ASM 386 Assembler RLL 386 Utilities
SSC-430	Technical Information Phone Support	Phone support *;Comments* Magazine, Troubleshooting Guides
CONSULT/DAILY CONSULT/LT	On- or off-site consulting on iRMK I.2 Kernel or other Intel products by Intel systems engineer. Available on a daily or long term basis.	
	iRMK Real-time Kernel	Customer Training Workshop
	80386 Programming Using ASM 386	Customer Training Workshop
	80386 System Software	Customer Training Workshop
	80386 System Hardware Design	Customer Training Workshop

iRMX® OPERATING SYSTEM FAMILY



The iRMX® family of operating systems provides designers with the world's most advanced real-time software for designs based on the Intel 8086/88, 80186/88, 80286, 386™, and i486™ family microprocessors. The product of twelve years of real-time expertise by Intel, iRMX software provides high-performance response to external events, excellent support of special-purpose hardware, and sophisticated real-time programming facilities.

A COMPLETE FAMILY OF REAL-TIME OPERATING SYSTEMS

- Multiprocessing support
- Multiple tasks and multiple jobs
- Multiple users
- On-target development
- Priority based and/or round robin scheduling
- User-extendable object oriented architecture

SOFTWARE WITH A FUTURE

- Leading real-time microprocessor software with over 6000 licenses sold
- Active iRMX Users Group (iRUG) with worldwide chapters, a regular newsletter, and an annual technical convention
- Applications easily migrated from iRMX I to iRMX II to iRMX III
- Application software from third-party sources
- Future 8086, 80186, 80286, 386, and i486 family processor support by iRMX operating systems.
- Optional networking to systems running the MS-DOS, VAX/VMS, UNIX, XENIX, iNDX, iRMX I, iRMX II, and iRMX III operating systems

October, 1989
Order Number 280109-001

SUPPORT FOR THE FULL RANGE OF REAL-TIME APPLICATIONS

The iRMX operating systems support the full range of real-time applications, from embedded control designs to reprogrammable systems which require dynamic creation, deletion, and priority arbitration of tasks. This flexibility makes it possible to save substantial staff retraining and software maintenance costs by using a single operating system for many different real-time systems and subsystems. Users have shown that the iRMX operating systems are ideal for such applications as:

avionics
communications
data acquisition and analysis
energy management
factory automation
financial trader workstations
image processing
machine control
manufacturing test
medical instruments
process control
railroad control
missile controls
satellite communications
simulation
transaction processing

REAL-TIME SOFTWARE FOR REAL-TIME APPLICATIONS

Real-time applications are easier to develop with real-time software. Operating systems designed for general business use typically lack essential real-time features, so real-time application development is often expensive, difficult, or even impossible. In contrast, iRMX software is real-time software designed to make the development of real-time applications easy and successful. Attributes offered by the iRMX software include:

High performance

For real-time applications, iRMX software is many times faster than general purpose operating systems. This high performance enables applications based on the iRMX operating systems to keep up with the rapid data and control flow of machine and communication interfaces.

A rich set of real-time programming facilities

The iRMX software includes a rich set of real-time programming facilities that are usually missing in whole or in part from non-real-time operating systems. These facilities include:

- preemptive, priority-based scheduling with round robin (time slice) scheduling within a priority level
- interrupt management with standard or user-defined exception handlers
- support for multiple tasks
- inter-task communication through mailboxes and semaphores
- deterministic program execution
- control of critical resources through regions

Support for designs based on Intel systems, single board computers, and components

Excellent support for special purpose hardware

Most real-time applications involve some special purpose hardware, and general purpose operating systems are often relatively monolithic and difficult to interface to this hardware. The iRMX operating systems are highly configurable, modular software systems which easily support custom hardware. Support for special purpose hardware includes:

- the ability to configure the operating system by layer
- hooks for user-written handlers at key points
- the ability to add operating system extensions
- standard device driver interfaces

More reliable code through iRMX® II and iRMX® III memory protection

When the iRMX II or iRMX III software allocates memory to a task, it assigns a combination of read, write, and execute-only status to the allocated code and data segments. If the code attempts to execute outside of this range (e.g., stack overflow) or write to a data segment marked read-only, the operating system will issue a "protection" error. This flag can be used to notify an operator of the exact location in the code where the problem occurred. Bounds and access rights checking, which is enforced by the hardware, can catch up to 90% of common coding errors.

COMPLETE REAL-TIME OPERATING SYSTEMS, NOT JUST A KERNEL

With comparable performance, the iRMX operating systems provide many features that are extra-cost items, or simply unavailable, in real-time kernels. These features make the development of real-time applications much easier and faster, but do not add unnecessary overhead. In fact, all functional layers above the nucleus are optional in the iRMX operating systems. This flexibility allows you to include only those features that your application requires.

The following is a brief description of the major functional groups within the iRMX operating systems.

Nucleus

The Nucleus is the heart of the operating system and controls all resources available to the system. The nucleus provides key real-time features including:

- Support of multiple tasks
- Priority based and time slice scheduling
- Dynamic priority adjustment
- Memory management
- Inter-task communication and synchronization using mailboxes and semaphores
- Interrupt management with custom exception handlers
- Time management
- Object management
- Addition of custom operating system extensions
- Inter-processor communication for multi-processor systems

Basic I/O System (BIOS)

The Basic I/O System (BIOS) provides primitives to read from and write to peripherals. The BIOS also sets up the file structures used by the system and provides access to all required peripherals through a standard device driver interface. Both synchronous and asynchronous system calls are supported. Many device drivers are provided with the iRMX operating systems, and custom device drivers and file drivers may be added by the user.

Extended I/O System (EIOS)

The Extended I/O System (EIOS) provides similar services to the BIOS, with simplified calls that give less explicit control of device behavior and performance. The software supports synchronous system calls and provides automatic buffering of I/O operations. The EIOS also provides a logical-to-physical device connection, and allows a program to specify a logical address for output.

Application Loader

The Application Loader is used to load programs from mass storage into memory, where they execute. Programs may be loaded under program or operator control.

Bootstrap Loader

The Bootstrap Loader is used to load the operating system or an iRMX application from mass storage into memory, and then to begin the system's execution.

Universal Development Interface (UDI)

The Universal Development Interface provides an easy-to-use interface with a standard set of system calls to allow programs and languages to be easily transported to or from the iRMX operating systems to other operating systems which support the UDI standard. For example, UDI lets iRMX host MS DOS-based tools.

System Debugger

The System Debugger is used to debug applications and give a view into the system itself. A static debugger provides a view of system objects. Source-level debuggers are separately available.

Human Interface

The Human Interface allows multiple users to effectively develop applications, maintain files, run programs, and communicate with the operating system. It consists of a set of system calls, a set of commands, and a Command Line Interpreter. Commands are available for file management, device management, and system status. Features include dynamic log-on, full line editing, user extensions, and support for background jobs. In addition, the Command Line Interpreter may be replaced for special applications. For example, a Computer Aided Tomography (CAT) scanner controlled by an iRMX operating system could use a custom Command Line Interpreter to allow the operator to direct the movement of the scanner.

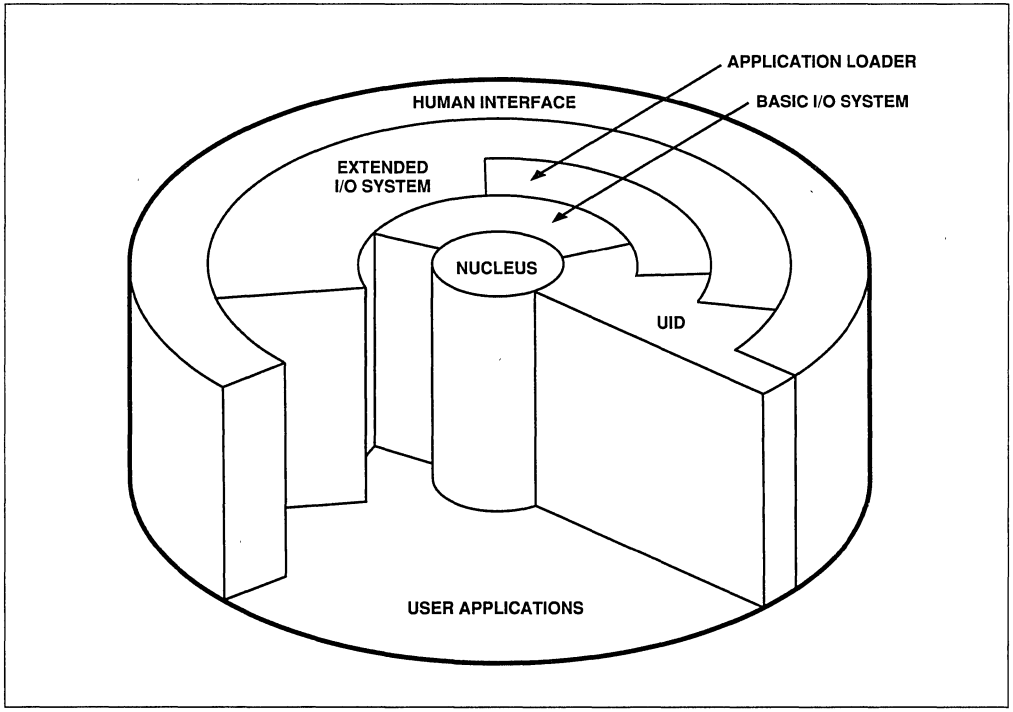


Figure 1: iRMX® Operating System Functional Elements

iRMX® I OPERATING SYSTEM

The iRMX I Operating System is the system first developed for real-time operating system support of the 8088 and 8086 microprocessors. It has become today's most widely accepted real-time operating system for microcomputers. Its features include:

- 16-bit operating system; uses 8086 instruction set and 8086 compilers
- Microprocessors: 8088, 8086, 80186, 80188; 80286, 386, i486 (Real address mode)
- Math co-processors: 8087, 80287, 387™
- Memory management: up to 1 MB, Real address mode
- Applications can be written using C, FORTRAN, Pascal, and PL/M compilers and assembler available from Intel.
- Ideal for embedded, nucleus-only applications, optimized for speed and compactness
- Applications can be easily upgraded to iRMX II- and iRMX III-based designs
- For MULTIBUS I and custom designs
- Development on iRMX or MS DOS hosts
- iRMX-hosted development

iRMX® II OPERATING SYSTEM

The iRMX II operating system features include:

- 16-bit operating system; uses 80286 instruction set and 80286 compilers
- Microprocessors: 80286, 386, i486
- Math co-processors: 80287, 387
- Dynamic memory management: up to 16 MB, 16-bit Protected address mode
- Applications can be written using C, FORTRAN, Pascal, and PL/M compilers and assembler available from Intel.
- Applications can be easily upgraded to iRMX III-based designs or back ported to iRMX I.
- For MULTIBUS I, MULTIBUS II, AT-Bus, and custom designs

iRMX® III OPERATING SYSTEM

The iRMX III operating system is a compatible derivative of the iRMX II operating system. The 32-bit functions let users gain the 32-bit power of the Intel 386 and i486 microprocessors. The iRMX III operating system handles 32-bit math and segments up to 4 gigabytes, retaining protection.

Binary compatibility with iRMX II lets iRMX II users easily move their applications to 32-bits on iRMX III with the 386 and 387 or i486 processors: Most 16-bit applications run without change; selected parts of an application can be moved to 32-bits while others remain unchanged. Its features include:

- 32-bit operating system; uses full 386/387 instruction set
- Microprocessors: 386, i486
- Math co-processor: 387, 2-5 times faster than 286 CPU-based math co-processor
- Dynamic memory management: up to 4 GB, segmented or flat, 32-bit Protected mode
- 32- and 16-bit tasks can run concurrently with full inter-task communication
- Applications can be written using C, FORTRAN, and PL/M compilers and assembler available from Intel.
- Custom device drivers and custom interrupt and exception handlers need to be 32 bits.
- For MULTIBUS I; MULTIBUS II; 386 CPU-based, AT-bus personal computers; and custom designs.
- PC support includes all Intel 300-series 386-based AT platforms.

iRMX® FAMILY COMPATIBILITY

iRMX® I	iRMX® II	iRMX® III

INTEL SYSTEMS

System 310 Family System 320 Family	System 120 Family System 320 Family System 520 Family	System 120 Family System 320 Family System 520 Family Systems 301, 302, 300SX

MULTIBUS®I SINGLE BOARD COMPUTERS

iSBC 86/C38 iSBC 86/05A iSBC 86/12A iSBC 86/30 iSBC 86/35 iSBC 186/03A iSBC 186/51 iSBC 188/56 iSBC 286/10A* iSBC 286/12* iSBC 286/14* iSBC 286/16* iSBC 386/12* iSBC 386/12S* iSBC 386/2X* iSBC 386/3X*	iSBC 286/10A iSBC 286/12 iSBC 286/14 iSBC 286/16 iSBC 386/12 iSBC 386/12S iSBC 386/2X iSBC 386/3X	iSBC 386/12 iSBC 386/12S iSBC 386/2X iSBC 386/3X

MULTIBUS®II SINGLE BOARD COMPUTERS

	iSBC 286/100A iSBC 386/116 iSBC 386/120 iSBC 386/258 iSBC 386/133 iSBC 486/125	iSBC 386/116 iSBC 386/120 iSBC 386/258 iSBC 386/133 iSBC 486/125

INTEL MICROPROCESSOR DESIGNS

8086, 8088, 80186, 80188, 80286*, 386*, i486*	80286, 386, i486	386, i486
8259A Programmable Interrupt Controller 8254 or 8253 Programmable Interval Timer 8274, 8251A, and 82530 serial controllers 8255 parallel interface Necessary memory		

*Real address mode with 8086 instruction set

MULTIBUS®II HARDWARE AND iRMX® SOFTWARE BUILD HIGH-PERFORMANCE SYSTEMS

MULTIBUS®II systems, which pass data over the bus using high-speed messages, enable engineers to easily assemble high-performance multiprocessor systems. Bus arbitration problems are virtually eliminated and slower speed I/O boards cannot slow down data transfer across the bus since all data is passed at the full bus bandwidth of 40 MBytes/second.

Nucleus Communications Service

The Nucleus Communications Service provides the software interface between application code and the MULTIBUS II message-passing coprocessor. This software simplifies the job of sending messages between tasks on different boards and provides a standard software interface to any other MULTIBUS II board in the system.

Multiprocessor iRMX® Systems

With iRMX II and iRMX III software and other software from Intel's family of real-time software products, engineers can design complex, high-capability systems with a minimum of custom code. An example is the system shown in figure 2. This system has a single iSBC 386/258 peripheral controller board that functions as both a boot server and file server to multiple CPU boards in the system. File transfers are handled via the iSBC 186/530 Network Interface Adapter, which also provides an Ethernet network connection. The iSBC 186/410 terminal controller board uses communication software that is downloaded from the system disk. The iSBX™ 279 Display Subsystem, together with iRMX Virtual Terminal software, provides access to any processor in the system via a single console displaying multiple windows.

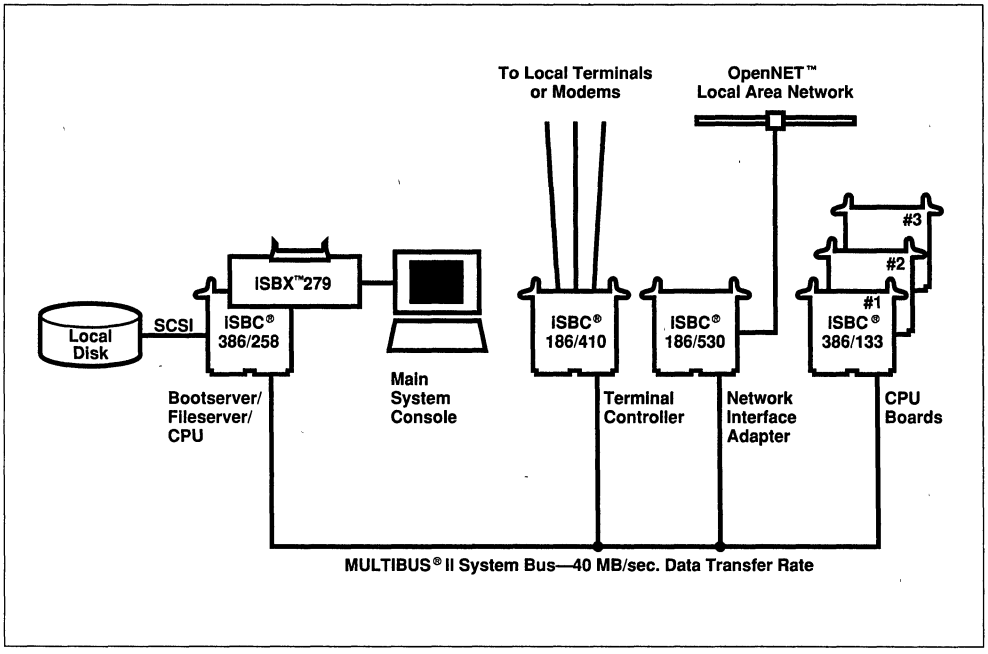


Figure 2: MULTIBUS®II iRMX® II Multiprocessor System

ON-TARGET DEVELOPMENT

A CONVENIENT WAY TO DEVELOP REAL-TIME APPLICATIONS

Designers familiar with both cross development and on-target development agree that on-target development is an easy, reliable method for developing applications. Testing is greatly simplified, and you need to become comfortable with only one operating system.

The iRMX operating systems provide solid on-target development capability—a capability entirely missing from other real-time software for microprocessors.

Developers can use the full, rich feature set of the iRMX operating systems for development, and then include only a minimum set of iRMX functions in their final applications. As a result, your final application receives the benefits of on-target development without the overhead that general purpose operating systems incur.

Included with the iRMX operating systems

- Interactive Configuration Utility (ICU)—a tool that can be used to generate a custom version of the operating system to match exact system requirements. The ICU automates the otherwise time consuming and error prone configuration of the system. The ICU accepts the user's system parameters and requirements, then builds a command file to compile, assemble, link, and locate necessary files.
- A Human Interface supporting multiple users
- Over 50 Human Interface commands for system status, device management, and file management
- A Command Line Interpreter supporting background jobs and full line editing
- Hardware traps to catch up to 90% of typical programming errors for iRMX II on iRMX applications
- System Debugger
- Bootstrap loader with debug option
- Parameter and data validation of operating system calls
- Universal Development Interface
- Numerous device drivers for Intel boards

Other development facilities are available separately for use with the iRMX® operating systems

- Re-entrant languages
- Assembly- and source-level debuggers
- Development utilities
- Graphics software for the iSBX™ 279 subsystem graphics controller (iRMX II and iRMX III)
- iPAT Performance Analysis Tool
- In Circuit Emulators, hosted on an IBM PC-AT or equivalent, to aid in hardware debugging and software tracing
- A variety of user-supplied utilities and special software available from the iRMX Users Group (iRUG)

iRMX® LANGUAGES

FULL LANGUAGE SUPPORT FOR iRMX® SYSTEMS

Intel has the languages you will need to develop a wide range of high-performance applications. iRMX languages include C, PL/M, FORTRAN, Pascal, and Assembler. iRMX language compilers run on an Intel 300 or 500 series microcomputer or System 120, and can be used for MULTIBUS®I or II target systems or embedded applications.

In addition to the wealth of languages available. iRMX-based systems are complemented by utilities with which to create and manage object modules. For the iRMX II and iRMX III systems, utilities are provided that allow system programmers to initialize and manage the memory protection features of the 80286, 386, and i486 transparently to the applications programmer. This latitude in configurability allows programmers to team their efforts in order to achieve a shorter development time than would otherwise be possible.

Because the high-level languages are actually resident on the iRMX-based system and can be licensed on a development license or pass-through license, OEMs can pass application software directly on to end users. End users may then tailor the OEM's system to better meet application needs by writing programs using the same languages.

Language-Independent Application Development

Intel's Object Module Format (OMF) enables several users to write different modules of an application in different languages, then link them together. Users can choose exactly the right language tools for specific pieces of the application, rather than compromising specialized tasks for the sake of one project-wide language.

Fast, Lean Programs for Rapid Processing

iRMX operating system calls are made directly from C, FORTRAN, Pascal (iRMX I and II), and PL/M. This means that application developers can take full advantage of the iRMX multi-tasking capability, whereby multiple applications execute concurrently on the operating system.

Application code can be easily transported across processor architectures to yield increased performance. For example, 8086 object code will run on the 80286, 386, and i486 processors.

Standardized Math Support

All iRMX languages support floating point operations. This ensures universal consistency in numeric computation results and enables the user to take advantage of the Intel 8087, 80287, and 387 Numeric Data Processors.

C LANGUAGE

The C programming language is known for its flexibility and portability. It is a block-structured, high-level language that is ideal for developing multi-user, multitasking, virtual memory operating systems to run in protected mode of the 80286, 386, and i486 processors.

Intel C compilers provide many substantial benefits to software developers, including:

- **Built-in functions.** Allow highly optimized code and eliminate the need for in-line assembly. With built-ins you can enable interrupts or directly control hardware I/O from the high level language.
- **Symbolic debug information.** Intel C compilers provide extensive symbolic debug information to speed development with an ICE or Soft-Scope debugger.

In addition, runtime libraries include the STDIO library, conversion routines, string manipulation routines, routines for performing 32-bit arithmetic and floating-point operations, and routines that provide an interface to the operating system.

The iC 86 compiler is a new generation C compiler providing high performance for embedded microprocessor designs. In addition to the features above, this compiler has the ability to mix memory models with "near" and "far" pointers. iC 86 is compatible with other ANSI C standard compilers and PL/M providing both standard C and PL/M calling conventions and has four optimization levels. iC 286 supports the 16 MB physical address space of the 80286 and enables programs to use 80286 features such as protection and virtual memory. It is upward compatible with iC 86.

iC 386 supports the full 4 Gigabyte physical address space of the 386 and enables programs to use new 80386 features, such as memory paging. It manipulates bit fields, pointers, addresses, and registers, enabling programs to take full advantage of the fundamental concepts of the 386 and i486 microprocessors. It is upward compatible from iC 86 and iC 286.

FORTRAN LANGUAGE

FORTRAN has long been the industry-standard programming language for numerical processing applications. FORTRAN 86 meets the ANSI FORTRAN 77 Language Subset Specification and includes many features of the full standard. It supports single-precision (32-bit), double-precision (64-bit), double-extended-precision (80-bit), complex (two 32-bit), and double-complex (two 64-bit) floating-point data types. Floating-point operations can be performed with software or with numeric coprocessors, such as the 8087, 80287, and 387. In addition, FORTRAN 86 has microprocessor extensions for performing direct byte- or word-oriented port I/O, developing reentrant procedures, and creating interrupt procedures.

- Features high-level support for floating-point calculations, transcendentals, interrupt procedures, and run-time exception handling
- Meets ANSI FORTRAN 77 subset language specifications
- Produces standard Intel 8086 object modules that can be compiled separately, linked to programs written in any Intel 8086 language.
- Supports the IEEE floating-point math standard with 8087, 80287, and 387 coprocessors
- Supports arrays larger than 64K bytes

PASCAL LANGUAGE

The Pascal compilers provide a complete implementation of the ISO proposed standard for Pascal for 8086, 80186, 80286, 386, and i486 microprocessors. In addition, the Intel Pascal compilers contain extensions to standard Pascal that tailor the resulting code to fit microcomputer applications. There are extensions for interrupt handling and port I/O. Predefined type extensions also allow you to specify the precision of real, integer, and unsigned calculations; check errors on 8087 or 80287 operations; and circumvent the type checking on calls to non-Pascal routines.

The following characteristics are common to Intel Pascal packages:

- Offers strict implementation of ISO standard Pascal
- Contains extensions to the ISO standard that are essential for microcomputer applications
- Allows separate compilation with type-checking enforced between modules
- Has compiler option to support full run-time range-checking
- Supports large array operation

PL/M LANGUAGE

The PL/M language is a structured language created specifically as a system development language for Intel microcomputers. It provides the advantages of a high-level language with the power of assembly language. PL/M is an excellent alternative to C in, for example, I/O-intensive applications. PL/M does not require a run-time environment and thus can produce highly optimized code. In 80286, 386, and i486 systems, PL/M is ideal for developing multi-user, multitasking, virtual-memory operation systems to run in protected mode. It is easy to learn and use, yet it allows complete access to the processor and it produces code whose efficiency rivals that of assembly language.

The following characteristics are common to Intel PL/M packages:

- Produces code whose efficiency rivals that of assembly language
- Has a block-structured syntax that encourages program modularity
- Requires fewer source statements than any other high-level language
- Has built-in syntax checker
- Allows foreign character sets in comments and strings
- Object code across multiple hosts is identical

Three PL/M compilers are available: PL/M 86 for 16-bit real address mode applications; PL/M 286 for 16-bit protected mode applications; and PL/M 386 for 32-bit applications. The compilers are upwardly compatible with each other. As a result, applications can be easily upgraded from PL/M 86 to PL/M 286 to PL/M 386 with only minor changes to the source code.

ASSEMBLERS

The Assemblers and Relocation/Linkage packages provide the tools that assembly-language programmers need to maintain complete control over the 8086, 80186, 80286, 386, and i486 microprocessors. The assembly languages are strongly typed, providing extensive checks on variables and labels. This helps catch many programming errors long before the debugging cycle. Macro facilities are also available to speed and simplify your work.

The relocation and linkage packages make your programs ready to run. They link programs together, assign absolute addresses, gather modules into libraries, and perform other system functions.

Assembler 86, 286 and 386:

- Are highly mnemonic and compact, and are strongly typed to detect errors at assembly time
- Place high-level symbolic information in object modules to enable symbolic debugging
- Have powerful text macro facility with three macro listing options, including string functions, and can expand conditional assembly pseudo-ops

The Relocation/Linkage Packages:

- Resolve PUBLIC/EXTERNAL references and perform intermodule type checking
- Select required modules from libraries to satisfy symbolic references and provide fast, easy management of object module libraries
- Simplify debugging by producing detailed maps that show references between program modules

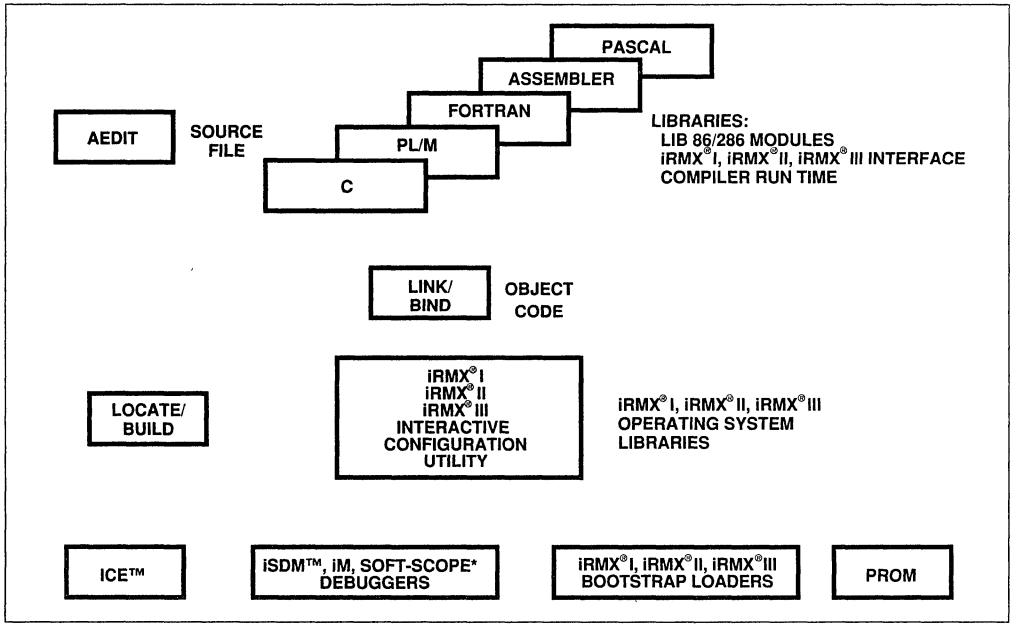


Figure 3: iRMX® Operating Systems Development Environment

COMPLETE HIGH-LEVEL DEBUGGING

For real-time applications development professionals want to focus on original source code for most debugging operations. The Soft-Scope debuggers do just that. They integrate the original source code into the debugging process directly. All breakpoint prompts and high-level stepping operations prompt with original source code rather than reporting what line number the program has reached or what assembly instruction is next.

Source Code Interface and On-Line Listings

The source code interface frees the programmer from having to divide attention between the console and program listings, eliminates the need to get a fresh program listing each time a small change is made, and reduces the time needed to make software modifications.

Automatic Expansion of Data Types

Symbols declared in the program are accessible by name for display and modification of contents. These symbols include arrays, structures, static variables, based variables, and stack-based variables (including local variables, re-entrant variables, and passed parameters). Memory can also be displayed with absolute references or with register-relative references.

Symbolic Display of All iRMX® System Objects

The VIEW command allows viewing the status of any iRMX object including tasks, jobs, mailboxes, semaphores, regions, and segments. With VIEW, the stack of a task can be examined to determine which iRMX call the task has made most recently. Any job's object directory and the list of ready and sleeping tasks can be examined.

Second Terminal Option

Because so many applications are screen-intensive, the Soft-Scope Debuggers allow the option of using a second terminal for all debugger I/O, freeing the main console for exclusive use by the application for application output.

Multi-Tasking Support

The Soft-Scope debuggers support simultaneous debugging of concurrent tasks when they are all linked together as a Human Interface command and each concurrent task is coded in a separate module. The debugger loads and then allows the user to suspend and resume execution of the tasks from the command line with the SUSPEND and RESUME commands. In this way the developer can observe the effect of dynamic changes on the software under test.

Handling of 80286, 386™, and i486™ Protection Traps and Software Exceptions

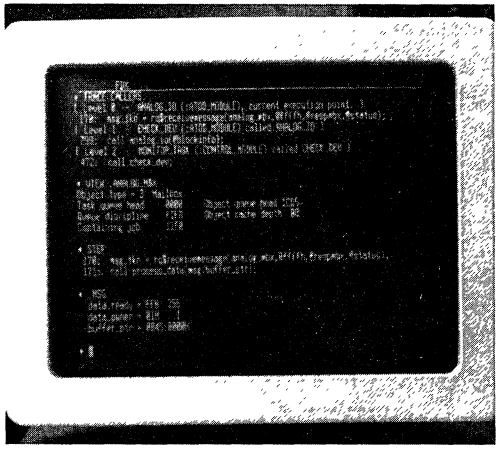
Exception Handling: The exact source line which causes an exception can easily be reached and displayed. All environmental and programmer exceptions are trapped and reported, without causing a Soft-Scope debugger exit.

Most of the 80286/386/i486 hardware traps are handled by the Soft-Scope debuggers, including Bounds Check (INT 5), Invalid Opcode (INT 6), Double Fault (INT 8), Stack Fault (INT 12), and General Protection (INT 13). Upon encountering one of these interrupts, the debugger breaks execution with a message similar to the following:

```
< General Protection fault (INT 13) >?
[ Break near line #145 in TESTPROC
  (:TESTMODULE) ]
145: ARRAYX(INDEX)=XYZ;
```

In the above example, the General Protection trap could have been caused by the variable INDEX being too large for the segment which contained ARRAYX, or by ARRAYX being based on an undefined pointer. Because the debugger handles these traps directly, other users in a multi-user system won't even be aware in most cases that there was a hardware fault.

Soft-Scope debuggers are available for iRMX I, iRMX II, and iRMX III designs. The Soft-Scope II and Soft-Scope III debuggers are available directly from Intel. The Soft-Scope I and Soft-Scope III debuggers are available from Concurrent Sciences, inc.



ASSEMBLY-LEVEL DEBUGGING

The user can use the iSDM monitor package to load programs into the target system from the development system, execute programs in an instruction-by-instruction manner, and add custom commands through the command extension interface.

Powerful Debugging Commands

The iSDM Monitor contains a powerful set of commands to support the debugging process on Intel 16- and 32-bit microprocessors. Some of the features included are: bootstrap of application software; selective execution of program modules based on breakpoints or single stepping requests; examination, modification, and movement of memory contents; examination and modification of CPU registers, including NPX registers; and disassembly of instruction set code. All results are displayed in clearly understandable formats.

Two versions of iSDM are available. iSDM II supports 16-bit code debug on 8086/186, 8088/188, 80286, and 386 processors. iSDM III supports 16- and 32-bit code debug on the 386 and i486 processors for iRMX III applications.

Numeric Data Processor Support

Arithmetic applications utilizing the 8087 or 80287 Numeric Processor Extension (NPX) are fully supported by the iSDM Monitor. In addition to executing applications with the full NPX performance, users may examine and modify the NPX's registers using decimal and real number format.

This feature allows the user to feel confident that correct and meaningful numbers are entered for the application without having to encode and decode complex real, integer, and BCD hexadecimal formats.

Command Extension Interface

The Command Extension Interface (CEI) allows the addition of custom commands to the iSDM Monitor commands. The CEI consists of various procedures that can be used to generate custom commands. Up to three custom commands (or sets of commands) can be added to the monitor without programming new EPROMs or changing the monitor's source code.

Program Load Capability

The iSDM loader allows the loading of 8086, 8088, 80186, 80188, 80286, 386, or i486 CPU-based programs into the target system. It executes on an iRMX development system and communicates with the target system through a serial link, a parallel link, or a fast parallel link.

Configuration Facility

The monitor contains a full set of configuration facilities which allows it to be carefully tailored to the requirements of the target system. Pre-configured EPROM-resident monitors are supplied by Intel for most MULTIBUS®I and MULTIBUS II CPU boards. iRMX I and iRMX II system users may use the configuration facilities to include the Bootstrap Loader (V5.0 or newer) in the monitor. iSDM can be easily configured for custom hardware.

The iSDM III monitor is preconfigured to be loaded with iRMX III. The user can configure I/O devices to be used by the monitor. iSDM III runs on any hardware iRMX III runs on.

The iSDM Monitor does not require the use of a development system. The monitor can be used by simply attaching a stand-alone terminal to the target system.
iSDM II is available as a separate product. The iSDM III monitor is included with the iRMX III operating system.

iM III MONITOR

The iM III monitor is a 32-bit debugging tool for the Intel 386 family processors: 386 and i486 microprocessors, and 387 co-processor. The tool gives the user visibility into 386-or i486 CPU-based hardware as the software executes. Although developers use iM III primarily for software debugging, they also find it useful for finding and solving hardware problems.

The monitor is highly configurable, so that users can shape it to fit the unique needs of their applications. The basic monitor provides raw debugging facilities, with a variety of ways to interact with the user's application code. The default configuration in the monitor package is ready to be programmed into EPROMs. It supports debugging on Intel's 386- and i486 CPU-based boards.

The package includes source code to let users prepare their own custom debugging routines. And configuration source code lets them configure the monitor to work on their own target systems. The custom code may be programmed into EPROMs or built into boot-loadable files. The iM III monitor also works with the Soft-Scope III high-level debugger.

Capabilities the iM III monitor gives its users include:

- Interact with the monitor from a terminal, from a host computer, or from a program running on the target system.
- Use a built-in set of monitor commands or replace or add commands during execution.
- Change the console controlling the monitor at any time.
- Redirect data from one console to another.
- Add functionality or change the user interface by intercepting calls to the monitor.
- Switch from one task to another.
- Develop a high-level debugging program that uses the monitor as a kernel.

AEDIT EDITOR

AEDIT is a full-screen editor designed specifically for software engineers and writers. It has many features that make it ideal for program editing. For example, it lets you switch between files instantly, and its split-screen windowing capability enables you to view two files at once. It has a macro facility that you can use to combine multiple functions into a single command. These macros can be created in two ways: by using AEDIT's "learning" mode to store your keystrokes or by using the powerful macro language. You can also use and modify the extensive library of macros provided with the editor.

With these and other features, such as context-sensitive command menus and shell escape to the operating system, AEDIT is the complete program-editing tool.

- Allows full screen editing of source code and documentation
- Provides a full range of editing support, from document processing to hexadecimal code entry
- Supports macros for repetitive or complex editing tasks
- Provides a powerful macro language for developing "smart" macros
- Supports dual-file editing with optional split-screen windowing
- Allows unlimited file size and line length
- Offers quick response with easy-to-use, context-sensitive command menus
- Is configurable and extensible for complete control of the editing process, yet remains easy to learn and use
- Supports documentation preparation with paragraph filling and justification options
- Provides shell escape function for access to operating system commands

iPAT™ PERFORMANCE ANALYSIS TOOLS

iPAT™, Intel's Performance Analysis Tool, helps engineers control the performance and reliability of a software-driven system by showing, via histograms and tables, the real-time execution activity of software in terms of range names or addresses.

- Provides real-time performance and code coverage analysis non-intrusively with 100% sampling
- Displays information using histograms or analysis tables
- Accepts specification of ranges with addresses, program symbolic names, or user-defined symbolic names
- Performs disarm/arm analysis on called subroutines, external interrupts, interrupt routines, operating system functions, or any execution address or range
- Hosted on PC/XT and PC/AT systems, using a serial link for target communications
- Presents an easy-to-use human interface, including function keys and color/monochrome graphics
- Available for 8086/88, 80286, and 386 microprocessors.

The iPAT products consist of DOS-hosted control and display software, plus appropriate microprocessor probes which replace the microprocessor in the target system. The iPAT 386 also can be operated in piggy-back fashion with the Intel ICE-386/25 in-circuit emulator in prototype systems at speeds up to 20 MHz. All iPAT products use an iPAT core base system, which also can be attached to ICE-186 or ICE-286 in-circuit emulators.

iRMX® TOOLBOX

ASSISTANCE FOR iRMX® PROJECT DOCUMENTS

The iRMX toolbox is a set of utilities to provide assistance to the software developer in text processing and document preparation.

Text Formatting (SCRIPT)

The SCRIPT utility is a text formatting program that streamlines document formatting and preparation. Commands include facilities to do paging, centering, left and right margins, justification, subscripts, superscripts, page headers and footers, underlines, boldface type, upper and lower case, etc.

Input text which has been prepared using the AEDIT text editor can be formatted using the SCRIPT utility.

Spelling Verification (SPELL, WSORT)

The SPELL utility finds misspelled words in a text file. The included dictionary can be expanded by the user for specialized vocabularies. This utility can be used interactively or in a batch mode.

File Comparisons (COMP)

The COMP utility performs line oriented text file comparisons showing changes between text, source, or object files.

Sort (ESORT, HSORT)

Files can be sorted on multiple keys (or fields) in ascending or descending order and the resultant sorted files stored.

iRMX® SOURCE CONTROL SYSTEM

SOFTWARE VERSION MANAGEMENT

The iRMX Source Control System (SCS) provides an integrated version control and generation management system for users in an iRMX software development cycle. This facility is useful for large and small software projects to assist in bringing more control, order and methodology to the software development process. SCS can be effectively used on a single iRMX System or across the OpenNET™ network. It can be utilized by developers using any of the popular iRMX languages—PL/M, Assembler, FORTRAN, C, Pascal or other special language requirements.

Controls Access to Source Files

With iRMX Source Control System the system manager has certain privileged commands. These commands can be useful to designate those team members who can access the source files only for object generation and those who can access the source files for updating or changing. Other such privileged commands include the ability to archive a specific version of source and combine several versions of a source file.

Tracks Changes to Source Files

The iRMX Source Control System keeps track of changes made to any source files. These changes are stored as backward deltas for disk economy and fast access to the latest version. The project team can now better interact and synchronize using the latest updated version for integration and testing. The specific versions of tools used to produce the source code is also tracked.

Approachable and Efficient

The iRMX Source Control System has a tutorial, menu interface, and on-line help facility that help make it very approachable by the user.

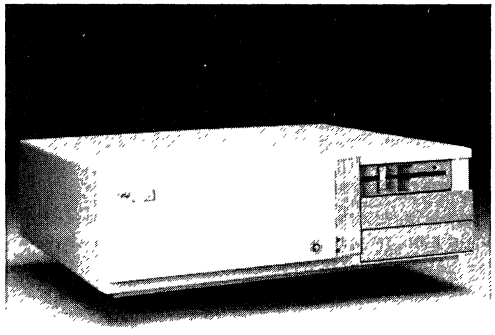
ONE OPERATING SYSTEM; A CHOICE OF BUS ARCHITECTURES

Intel has integrated iRMX development software into systems based on the PC AT bus, MULTIBUS I, and MULTIBUS II, all using the power of the Intel 386 32-bit microprocessor. These systems are available in configurations suited for software development and target systems.

Compatibility of iRMX-based software across buses makes it easy to move applications among System 120, System 320, System 520, and user-built systems based on AT-Bus, MULTIBUS I, and MULTIBUS II. This flexibility lets the user select from a spectrum that ranges from AT-bus price to minicomputer performance and functions.

SYSTEM 120

The Intel System 120 is a 386-based, PC/AT platform that delivers real-time capability to users demanding a low-cost system for running time-critical applications. The System 120 combines a PC/AT bus configuration of Intel's iRMX real-time operating system and an Intel 386-based PC/AT platform.



Low Cost PC/AT Based Configurations

The System 120 target models are available with a number of processor speed, memory, and mass storage options to fit a range of applications. These include a basic system with 8 open slots, and a 40 MB hard disk system with a 387 math coprocessor and floppy disk.

Intel offers PC/AT add-in boards for the System 120 that include: 2 MB and 8 MB 32-bit memory boards, the OpenNET PCLINK2 networking board and the iPCX 344A BITBUS™ board. A standard keyboard is also available.

Easy Application Development

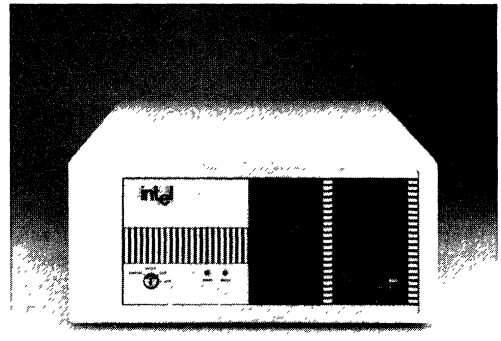
You can develop applications for AT bus, MULTIBUS I and MULTIBUS II directly on the system using the System 120 Development Toolkit. In addition to the iRMX II and iRMX III Operating Systems for the System 120, the toolkit contains: PL/M Compiler, Assembler, AEDIT, and a source level debugger, Soft-Scope, and an interface manager that can be used as a front end to applications.

DOS Application Compatibility

The System 120 supports the DOS 3.X and later operating systems as well as iRMX II and III, enabling you to use popular DOS applications to process data collected in real time. Many common applications are already available from software vendors. The applications include data bases, menu systems, and device drivers. The System 120 hard disk can be divided into iRMX and DOS partitions, allowing users to boot from either partition. A System 120 utility allows transfer of iRMX files to a DOS disk. DOS execution requires a customer-supplied version of DOS, a video adapter, a monitor, and a keyboard.

SYSTEM 320

The System 320 is based on the MULTIBUS®I architecture (IEEE 796) industry standard system bus supported by over 200 vendors providing 2000 compatible products and the iRMX® operating systems, composed of modular layers, highly configurable for tailoring to target applications. A wide range of popular industry standard high-level languages are supported for application development. Special configurations can be tailored by the user, by Intel's Custom System Integration group or by Intel's authorized Value Added Distribution Centers.



The System 320 consists of a system package which contains a seven slot MULTIBUS cardcage, a power supply, and three 5¼" full-height peripheral bays. Available options include CPU, memory, peripheral controller, operating systems, storage devices, channel communications, host communications, networks, productivity software and accessories.

The System 320 is available in a wide range of configurations based on the 386 microprocessor. All 386 based models include the 387 numeric processor. The 386 based systems can be expected to perform two to three times faster than the 80286 based System 310 models.

SYSTEM 520

The System 520 is built around Intel's MULTIBUS®II System Architecture (MSA) to ease the development and integration work of MULTIBUS II OEM system designers. MSA delivers open system interface and protocol standards that build on and extend the basic MULTIBUS (IEEE/ANSI 1296) bus standard. The MSA specifications define diagnostics, built-in self test, system initialization and boot loading, board configuration, and message passing. The user benefits from MSA because the level of vendor and board compatibility has been raised above basic electrical bus specifications to a set of powerful programmatic interfaces that handle all of the details of bus specifics with software. This provides OEMs quicker time to market with faster system integration and shorter design cycles.

Easy User Expansion and Re-Composable Systems

The MULTIBUS II System Architecture is used as the foundation for integrating Intel's full line of Single Board Computer (iSBC®) modules and iRMX II system software into the System 520. OEMs have the option to buy the contents of the system (i.e., the boards, the firmware, the software, etc.) separately, and re-compose all or part of the system's pieces into a different configuration or enclosure.

The System is available with or without the iRMX Operating System.

Easy user expansion and recomposability of the System 520 is supported by a line of System Integration Toolkits (SIT kits) that contain all the firmware necessary to allow standard, off-the-shelf MULTIBUS II boards to integrate cleanly into the System 520. With these toolkit products, the OEM can purchase the pieces needed and profit from greater ease of use, ease of integration, and higher levels of open standards.

A Network IN the System

Using the MULTIBUS II backplane as an ultra-fast network (40 MB/sec), multiple peer-to-peer 386™- and i486™-based iRMX application processors operate as independent "networked" iRMX systems over the MULTIBUS II Parallel System Bus (PSB). Each iRMX application processor running Intel's OpenNET™ network software will provide transparent distributed file sharing, file transfer, and virtual terminal capability among all application processors on the backplane, and IEEE 802.3-based (1.25 MB/sec) OpenNET networked system nodes (See Figure 4). Using the PSB as a network makes the System 520 a high performance "minicomputer" cluster condensed into one multiprocessor system.

The major advantage to the OEM is total network extensibility inside and outside the system, using the same OpenNET software The key benefits are: reduced cost through the "replacement" of multiple uni-processor networked systems and servers, drastically reduced physical space requirements, increased overall network throughput and performance, and preserved software investment.

The customer can choose the combination of packaging, CPU/system performance, and communications bandwidth suited for the application.

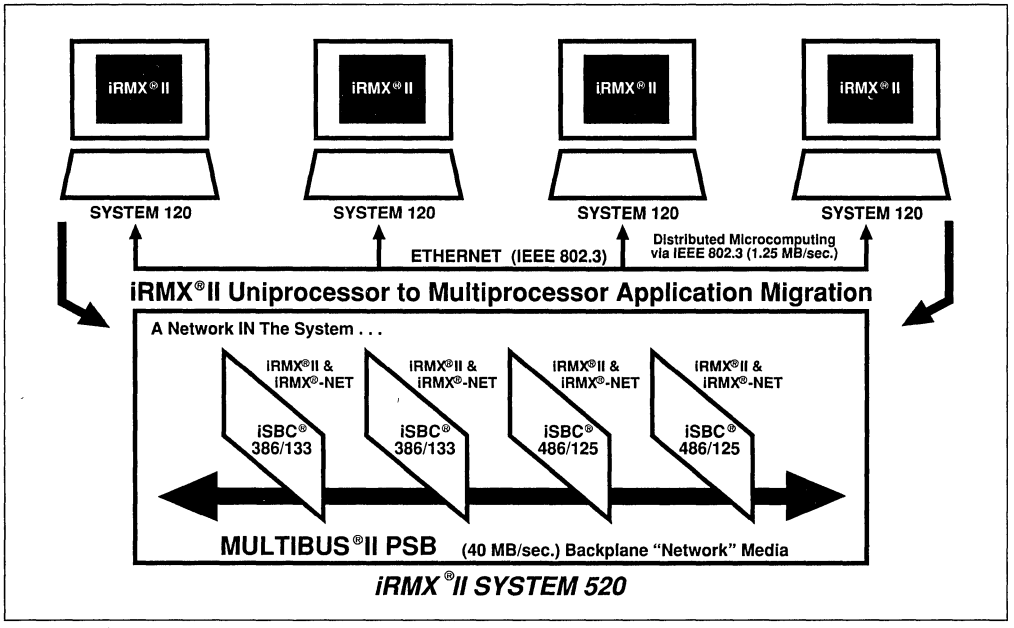


Figure 4: A Network IN the System with an iSBC 386 and i486 board cluster; also shows iRMX II application migration from networked System 120s to an iRMX II System 520 running OpenNET IN the System

COMPLETE OpenNET™ SOLUTION FOR REAL-TIME SYSTEMS

Many real-time applications require network communication. Intel's iRMX-NET Release 3.0 delivers a rich set of networking capabilities and a full range of iRMX platform support for iRMX System 120 (AT-bus), 320 (MULTIBUS I) and 520 (MULTIBUS II)

Transparent Network File Access

iRMX-NET implements the NFA protocol to provide transparent file access capabilities among iRMX, DOS, VAX/VMS, UNIX, XENIX and iNDX systems on the OpenNET network. Remote files are accessed as if they resided on the local iRMX system. iRMX-NET can be configured as a network file consumer, file server, or both, depending on the application's requirements.

With the addition of iRMX-NET, the iRMX Human Interface commands and system calls are transparently extended to remote access as well. Transparency means that applications using the iRMX Human Interface commands or BIOS system calls do not need to know whether the files they access reside locally or on some remote system.

OSI Transport and Distributed Name Server with Programmatic Interface

The iRMX-NET R3.0 product includes iNA 960 R3 OSI Transparent and Network software preconfigured for a variety of Intel Network Interface Adapters.

iRMX-NET R3.0 also includes the iRMX-NET Distributed Name Server software. The Distributed Name Server software maintains and provides access to a network directory database. The database is distributed across the network with each system maintaining its own logical piece of the directory. The Distributed Name Server software provides a full set of network directory services and is used to perform such tasks as logical name to network address mapping for establishing network connections between systems.

The combination of transparent network file access with iRMX commands and system calls, plus direct programmatic access to the iNA 960 Transport and iRMX-NET Distributed Name Server software gives the programmer a powerful set of capabilities for developing real-time network applications.

Remote Boot for Diskless Systems

iRMX-NET R3.0 supports networked diskless systems by providing network Boot Consumer, Boot Server and File Server capabilities.

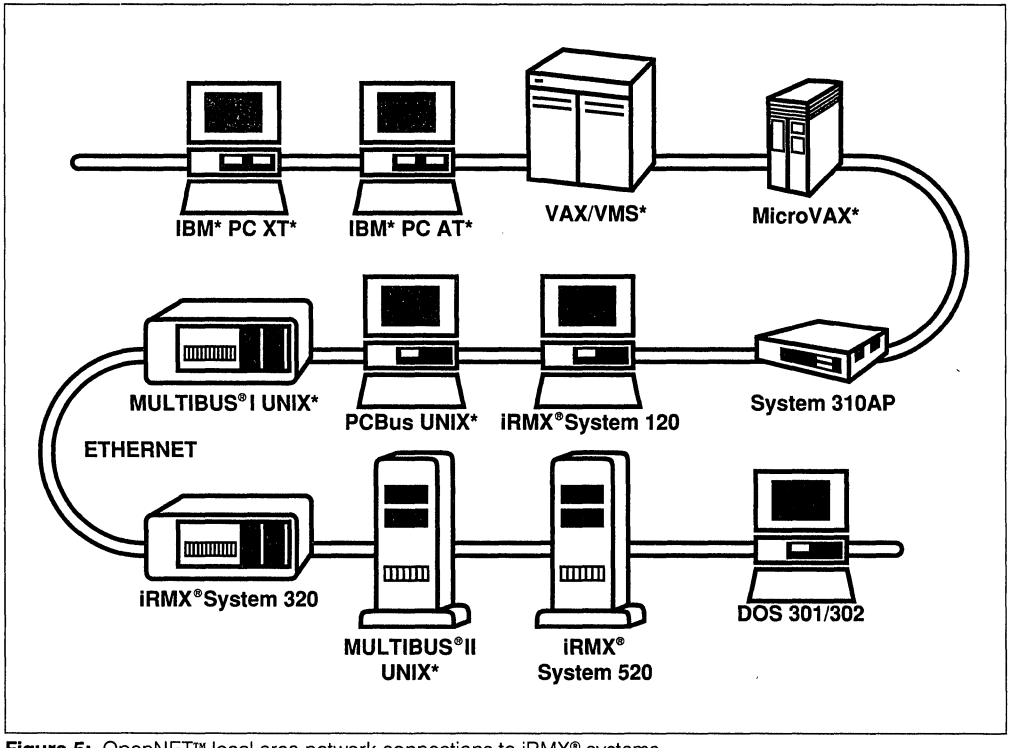


Figure 5: OpenNET™ local area network connections to iRMX® systems.

iRMX® VIRTUAL TERMINAL

Virtual Terminal allows local iRMX users to "Logon" to a remote Intel iRMX node within an OpenNET network or across the MULTIBUS II system bus. This capability enables users to access all the available resources on the remote system. In addition the iRMX Virtual Terminal is fully interoperable with other OpenNET Virtual Terminal products. Now a PC, VAX, or UNIX user can "connect" to a remote iRMX system without the need to use a locally connected iRMX terminal.

The iRMX Virtual Terminal server can be configured to support from 1 to 32 virtual terminal connections per system.

The administration utility allows the system manager to disable, terminate or start the iRMX Virtual Terminal server. The capability to report on the status of all the virtual terminal connections to the local server is also supported.

SOFTWARE SERVICES

A FULL RANGE OF TECHNICAL SUPPORT

With the iRMX operating systems you're not alone when you're developing a real-time application. Intel has the best technical sales support in the real-time business. If you need help, training, consulting, and design advice are readily available.

Standard Software Support

All Intel software products include Intel's Software Support for a 90 day period immediately following the licensing and receipt of the product. Standard Support includes:

- Product updates
- Subscription Service and technical product information distributed via:
  - —Monthly issue of *;Comments* newsletter
  - —Quarterly Troubleshooting Guides
  - —Software Problem Report (SPR) Service
- Technical Information Phone Service (iTIPS™) toll-free hot line
- Membership in Insite™ User Program Library

Additional Services Available

- Consulting services on a long or short-term basis (Systems Engineering Support)
- Worldwide training workshops on a wide variety of Intel products
- A full range of hardware maintenance services for end users or OEM/VAR customers

Ordering Codes	Product

iRMX® OPERATING SYSTEMS

RMXIJKIT	iRMX I operating system
RMXIJKITS	iRMX I operating system with AEDIT, ASM 86, and PL/M 86
SYRIJKIT	iRMX I operating system with one-year software support
SYRIJKITS	iRMX I operating system with AEDIT, ASM 86, and PL/M 86 and one-year software support
RMXIDC	iRMX I manual set
RMXIIKIT	iRMX II operating system
RMXIIKITS	iRMX II operating system with AEDIT, ASM 86, ASM 286, PL/M 286
SYRIIKIT	iRMX II operating system with one-year software support
SYRIIKITS	iRMX II operating system with AEDIT, ASM 86, ASM 286, PL/M 286 and one-year software support
RMXIIDC7	iRMX II manual set
SYRIIIKIT	iRMX III operating system, assembler, utilities for AT bus
SYRIIIMBKIT	iRMX III operating system, assembler, utilities for AT bus and MULTIBUS
RMXIIIMNL	iRMX III manual set

DEVELOPMENT TOOLS

iRMX864	AEDIT text editor for iRMX I operating system
R286EDI286	AEDIT text editor for iRMX II and iRMX III operating systems
R86ASM86	ASM/R&L 86 package for iRMX I operating system
R286ASM86	ASM/R&L 86 package for iRMX II and iRMX III operating systems
R286ASM286	ASM/R&L 286 package for iRMX II and iRMX III operating systems
D386ASM386	ASM/R&L 386 package for iRMX III operating system
R86C86	iC 86 package for iRMX I operating system
R286C286	iC 286 package for iRMX II and iRMX III operating systems
(Early 1990)	iC 386 package for iRMX III operating system
R86FOR86	FORTRAN 86 package for iRMX I operating system
R286FOR286	FORTRAN 286 package for iRMX II and iRMX III operating systems
(Early 1990)	FORTRAN 386 package for iRMX III operating systems
R86PLM86	PL/M 86 package for iRMX I operating system
R286PLM86	PL/M 86 package for iRMX II and iRMX III operating systems
R286PLM286	PL/M 286 package for iRMX II and iRMX III operating systems
D386PLM386	PL/M 386 package for iRMX III operating system
R86PAS86	Pascal 86 package for iRMX I operating system
R286PAS286	Pascal 286 package for iRMX II and iRMX III operating systems
RMXIISFTSCP	Soft-Scope II debugger for iRMX II
(Early 1990)	Soft-Scope III debugger for iRMX III
SDMSC	System Debug Monitor
SYRIIPATKIT	System Debug Monitor and iPAT 286 performance analysis tool
RMX286TBL	iRMX Toolbox
RMXSCSKIT	Source Control

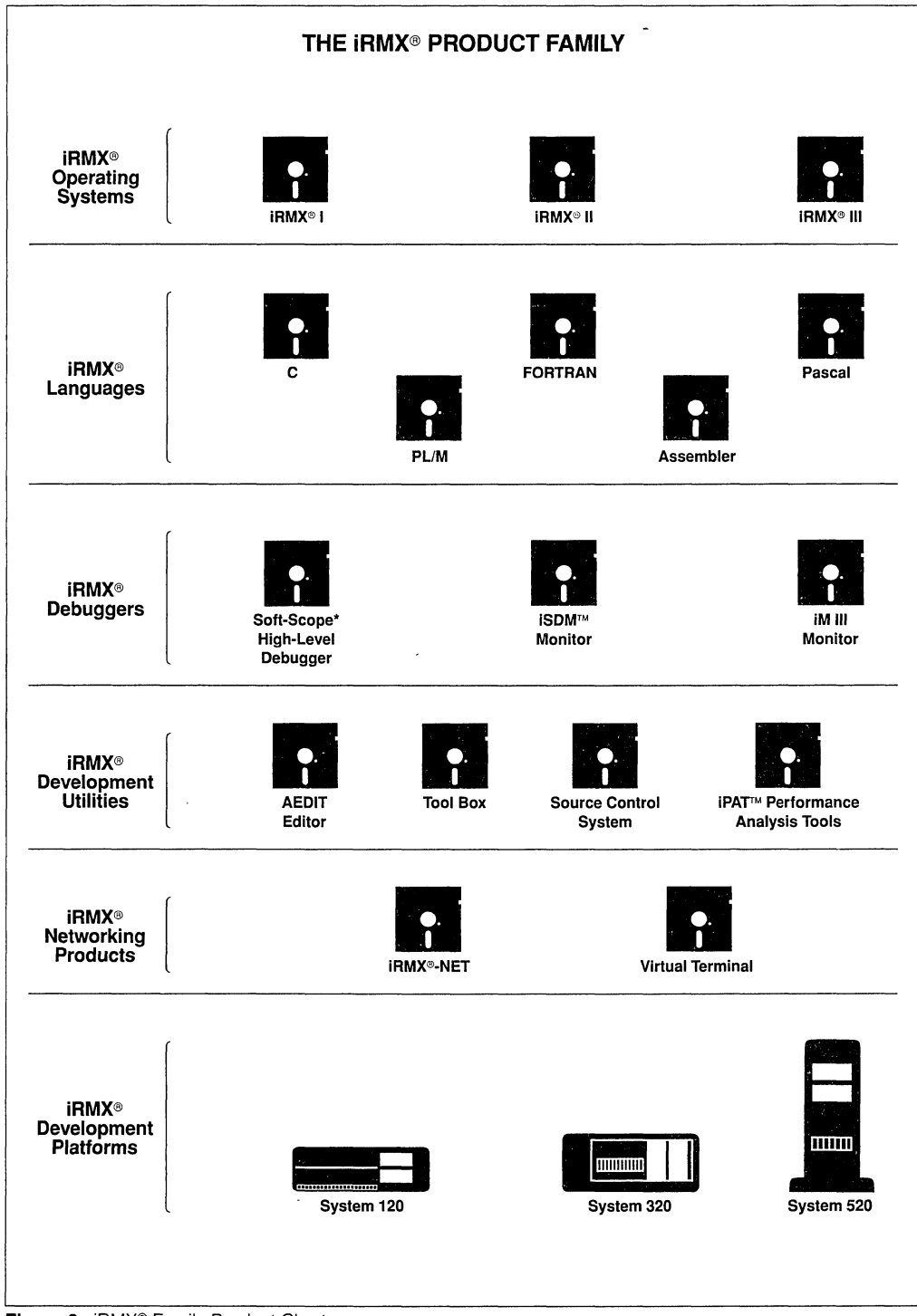
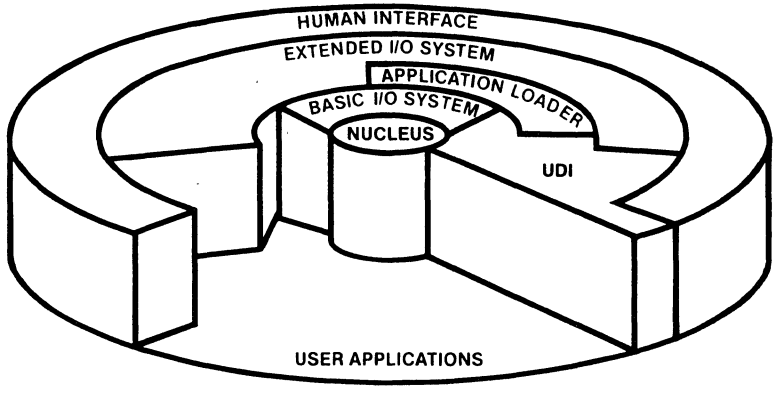


Figure 6: iRMX® Family Product Chart

iRMX® I OPERATING SYSTEM

- **Real-Time Processor Management for Time-Critical 8086, 8088, 80186, 80188, and 80286/386™ (Real Address Mode) Applications**
- **On-Target System Development with Universal Development Interface (UDI)**
- **Configurable System Size and Function for Diverse Application Requirements**
- **All iRMX® I Code Can Be (P)ROM'ed to Support Totally Solid State Designs**
- **Configured Systems for the 8086, 80286, and 386 Processors in the Intel System 300 Series Microcomputers**
- **Multi-Terminal Support with Multi-User Human Interface**
- **Broad Range of Device Drivers Included for Industry Standard MULTIBUS® Peripheral Controllers**
- **Support of 8087, 80287, and 80387 Processor Extension**
- **Powerful Utilities for Interactive Configuration and Real-Time Debugging**

The iRMX I Operating System is an easy-to-use, real-time, multi-tasking and multi-programming software system designed to manage and extend the resources of iSBC® 86, iSBC 186, iSBC 188, iSBC 286, and iSBC 386 Single Board Computers, as well as other 8086, 8088, 80186, 80188, and 80286/386™ (Real Address Mode) based microcomputers. The Operating System provides a number of standard interfaces that allow iRMX I applications to take advantage of industry standard device controllers, hardware components, and a number of software packages developed by Independent Software Vendors (ISVs). Many high-performance features extend the utility of iRMX I Systems into applications such as data collection, transaction processing, and process control where immediate access to advances in VLSI technology is paramount. These systems may deliver real-time performance and explicit control over resources; yet also support applications with multiple users needing to simultaneously access terminals. The configurable layers of the System provide services ranging from interrupt management and standard device drivers for many sophisticated controllers, to data file maintenance commands provided by a comprehensive multi-user human interface. By providing access to the standard Universal Development Interface (UDI) for each user terminal, Original Equipment Manufacturers (OEMs) can pass program development and target application customization capabilities to their users.



210885-1

iRMX® VLSI Operating System

November 1988
Order Number: 210885-004

The iRMX I Operating System is a complete set of system software modules that provide the resource management functions needed by computer systems. These management functions allow Original Equipment Manufacturers (OEMs) to best use resources available in microcomputer systems while getting their products to market quickly, saving time and money. Engineers are relieved of writing complex system software and can concentrate instead on their application software.

This data sheet describes the major features of the iRMX I Operating System. The benefits provided to engineers who write application software and to users who want to take advantage of improving microcomputer price and performance are explained. The first section outlines the system resource management functions of the Operating System and describes several system calls. The second section gives a detailed overview of iRMX I features aimed at serving both the iRMX I system designer and programmer, as well as the end users of the product into which the Operating System is incorporated.

FUNCTIONAL DESCRIPTION

To take best advantage of 8086, 8088, 80186, 80188, and 80286/386 (Real Address Mode) microprocessors in applications where the computer is required to perform many functions simultaneously, the iRMX I Operating System provides a multiprogramming environment in which many independent, multi-tasking application programs may run. The flexibility of independent environments allows application programmers to separately manage each application's resources during both the development and test phases.

The resource management functions of the iRMX I System are supported by a number of configurable software layers. While many of the functions supplied by the innermost layer, the Nucleus, are required by all systems, all other functions are optional. The I/O systems, for example, may be omitted in systems having no secondary storage requirement. Each layer provides functions that encourage application programmers to use modular design techniques for quick development of easily maintainable programs.

The components of the iRMX I Operating System provide both implicit and explicit management of system resources. These resources include processor scheduling, up to one megabyte of system memory, up to 57 independent interrupt sources, all input and output devices, as well as directory and data files contained on mass storage devices and accessed by a number of independent users. Management of these system resources and methods for sharing resources between multiple processors and users is discussed in the following sections.

Process Management

To implement multi-tasking application systems, programmers require a method of managing the different processes of their application, and for allowing the processes to communicate with each other. The Nucleus layer of the iRMX I System provides a number of facilities to efficiently manage these processes, and to effectively communicate between them. These facilities are provided by system calls that manipulate data structures called tasks, jobs, regions, semaphores and mailboxes. The iRMX I System refers to these structures as 'objects".

Tasks are the basic elements of all applications built on the iRMX I Operating System. Each task is an entity capable of executing CPU instructions and issuing system calls in order to perform a function. Tasks are characterized by their register values (including those of an optional 8087, 80287, or 80387 Numeric Processor Extension), a priority between 0 and 255, and the resources associated with them.

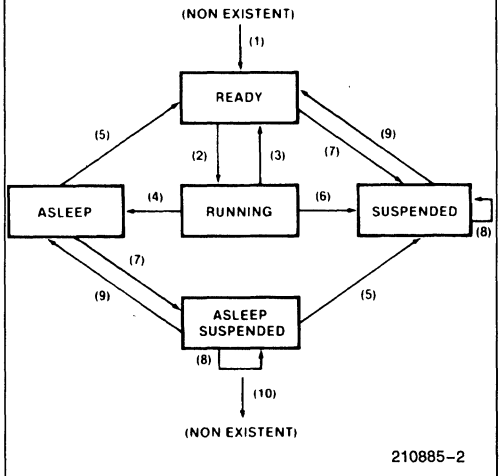
Each iRMX I task in the system is scheduled for operation by the iRMX I Nucleus. Figure 1 shows the five states in which each task may be placed, and some examples of how a task may move from one state to another. The iRMX I Nucleus ensures that each task is placed in the correct state, defined by the events in its external environment and by the task issuing system calls. Each task has a priority to indicate its relative importance and need to respond to its environment. The Nucleus guarantees that the highest priority ready-to-run task is the task that runs. The nucleus can also be configured to allow multiple tasks of the same priority to run in a round-robin, time-slice fashion.

Jobs are used to define the operating environment of a group of tasks. Jobs effectively limit the scope of an application by collecting all of its tasks and other objects into one group. Because the environment for execution of an application is defined by an iRMX I job, separate applications can be efficiently developed by separate development teams.

The iRMX I Operating System provides two primary techniques for real-time event synchronization in multi-task applications: regions and semaphores.

Regions are used to restrict access to critical sections of code and data. Once the iRMX I Operating System gives a task access to resources guarded by a region, no other tasks may make use of the resources, and the task is given protection against deletion and suspension. Regions are typically used to protect data structures from being simultaneously updated by multiple tasks.

Semaphores are used to provide mutual exclusion between tasks. They contain abstract "units" that are sent between the tasks, and can be used to implement the cooperative sharing of resources.



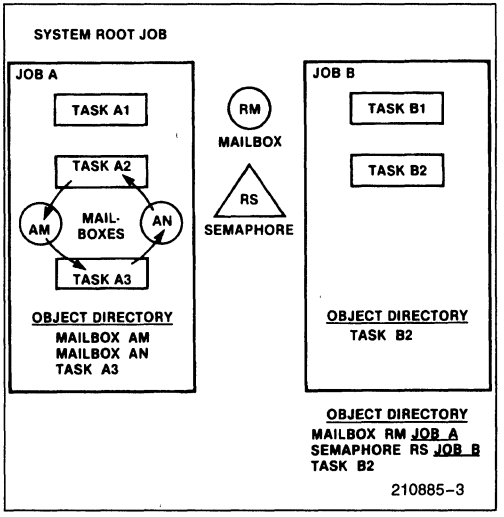
NOTES:
1. Task is created.
2. Task becomes highest priority ready task.
3. Task gets pre-empted by one with higher priority.
4. Task calls SLEEP or task waits at an exchange.
5. Task sleep period has ended, message was sent to waiting task or wait has ended.
6. Task calls SUSPEND on self.
7. Task suspended by other than self.
8. Task suspended by other than self or a resume that did not bring suspension depth to zero.
9. Task was resumed by other task.
10. Task is deleted.

Figure 1. Task State Diagram

Multi-tasking applications must communicate information and share system resources among cooperating tasks. The iRMX I Operating System assigns a unique 16-bit number, called a token, to each object created in the System. Any task in possession of this token is able to access the object. The iRMX I Nucleus allows tasks to gain access to objects, and hence system resources, at run-time with two additional mechanisms: mailboxes and object directories.

Mailboxes are used by tasks wishing to share objects with other tasks. A task may share an object by sending the object token via a mailbox. The receiving task can check to see if a token is there, or can wait at the mailbox until a token is present.

Object Directories are also used to make an object available to other tasks. An object is made public by cataloging its token and name in a directory. In this manner, any task can gain access to the object by knowing its name, and job environment that contains the directory.



Two example jobs are shown in Figure 2 to demonstrate how two tasks can share an object that was not known to the programmer at the time the tasks were developed. Both Job 'A' and Job 'B' exist within the environment of the 'Root Job' that forms the foundation of all iRMX I systems. Each job posseses a directory in which tasks may catalog the name of an object. Semaphore 'RS', for example, is accessible by all tasks in the system, because its name is cataloged in the directory of the Root Job. Mailbox "AN" can be used to transfer objects between Tasks 'A2' and 'A3' because its token is accessible in the object directory for Job 'A'.

Table 1 lists the major functions of the iRMX I Nucleus that manages system processes.

Memory Management

Each job in an iRMX I System defines the amount of the one megabyte of addressable memory to be used by its tasks. The iRMX I Operating System manages system memory and allows jobs to share this critical resource by providing another object type: segments.

Segments are contiguous pieces of memory between 16 Bytes and 64 Kbytes in length, that exist within the environment of the job in which they were created. Segments form the fundamental piece of system memory used for task stacks, data storage, system buffers, loading programs from secondary storage, passing information between tasks, etc.

The example in Figure 2 also demonstrates when information is shared between Tasks 'A2' and 'A3'; 'A2' only needs to create a segment, put the information in the memory allocated, and send it via the Mailbox 'AM' using the RQ$SEND$MESSAGE sys-

Table 1. Process Management System Calls

System Call	Function Performed
RQ$CREATE$JOB	Creates an environment for a number of tasks and other objects, as well as creating an initial task and its stack.
RQ$DELETE$JOB	Deletes a job and all the objects currently defined within its bounds. All memory used is returned to the job from which the deleted job was created.
RQ$OFFSPRING	Provides a list of all the current jobs created by the specified job.
RQ$CATALOG$OBJECT	Enters a name and token for an object into the object directory of a job.
RQ$UNCATALOG$OBJECT	Removes an object's token and its name from a job's object directory.
RQ$LOOKUP$OBJECT	Returns a token for the object with the specified name found in the object directory of the specified job.
RQGETTYPE	Returns a code for the type of object referred to by the specified token.
RQ$CREATE$MAILBOX	Creates a mailbox with queues for waiting tasks and objects with FIFO or PRIORITY discipline.
RQ$DELETE$MAILBOX	Deletes a mailbox.
RQ$SEND$MESSAGE	Sends an object to a specified mailbox. If a task is waiting, the object is passed to the appropriate task according to the queuing discipline. If no task is waiting, the object is queued at the mailbox.
RQ$RECEIVE$MESSAGE	Attempts to receive an object token from a specified mailbox. The calling task may choose to wait for a specified number of system time units if no token is available.
RQ$DISABLE$DELETION	Prevents the deletion of a specified object by increasing its disable count by one.
RQ$ENABLE$DELETION	Reduces the disable count of an object by one, and if zero, enables deletion of that object.
RQ$FORCE$DELETE	Forces the deletion of a specified object if the disable count is either 0 or 1.
RQ$CREATE$TASK	Creates a task with the specified priority and stack area.
RQ$DELETE$TASK	Deletes a task from the system, and removes it from any queues in which it may be waiting.
RQ$SUSPENDS$TASK	Suspends the operation of a task. If the task is already suspended, its suspension depth is increased by one.
RQ$RESUME$TASK	Resumes a task. If the task had been suspended multiple times, the suspension depth is reduced by one, and it remains suspended.
RQ$SLEEP	Causes a task to enter the ASLEEP state for a specified number of system time units.
RQGETTASK$TOKENS	Gets the token for the calling task or associated objects within its environment.
RQSETPRIORITY	Dynamically alters the priority of the specified task.
RQGETPRIORITY	Obtains the current priority of a specified task.
RQ$CREATE$REGION	Creates a region, with an associated queue of FIFO or PRIORITY ordering discipline.
RQ$DELETE$REGION	Deletes the specified region if it is not currently in use.
RQ$ACCEPT$CONTROL	Gains control of a region only if the region is immediately available.
RQ$RECEIVE$CONTROL	Gains control of a region. The calling task may specify the number of system time units it wishes to wait if the region is not immediately available.
RQ$SEND$CONTROL	Relinquishes control of a region.
RQ$CREATE$SEMAPHORE	Creates a semaphore.
RQ$DELETE$SEMAPHORE	Deletes a semaphore.
RQ$SEND$UNITS	Increases a semaphore counter by the specified number of units.
RQ$RECEIVE$UNITS	Attempts to gain a specified number of units from a semaphore. If the units are not immediately available, the calling task may choose to wait.

tem call (see Table 1). Task 'A3' would get the message by using the RQ$RECEIVE$MESSAGE system call. The Figure also shows how the receiving task could signal the sending task by sending an acknowledgement via the second Mailbox 'AN'.

Each job is created with both maximum and minimum limits set for its memory pool. Memory required by all objects and resources created in the job is taken from this pool. If more memory is required, a job may be allowed to borrow memory from the pool of its containing job (the job from which it was created). In this manner, initial jobs may efficiently allocate memory to jobs they subsequently create, without knowing their exact requirements.

The iRMX I Operating System supplies other memory management functions to search specific address ranges for available memory. The System performs this search at system initialization, and can be configured to ignore non-existent memory and addresses reserved for I/O devices and other application requirements.

Table 2 lists the major system calls used to manage the system memory.

Interrupt Management

Real-time systems, by their nature, must respond to asynchronous and unpredictable events quickly. The iRMX I Operating System uses interrupts and the event-driven Nucleus described earlier to give real-time response to events. Use of a pre-emptive scheduling technique ensures that the servicing of high priority events always takes precedence over other system activites.

The iRMX I Operating system gives applications the flexibility to optimize either interrupt response time or interrupt response capability by providing two tiers of Interrupt Management. These two distinct tiers are managed by Interrupt Handlers and Interrupt Tasks.

Interrupt Handlers are the first tier of interrupt service. For small simple functions, interrupt handlers are often the most efficient means of responding to an event. They provide faster response than interrupt tasks, but must be kept simple since interrupts (except the 8086, 8088, 80186, 80188, 80286, and 386™ processors non-maskable interrupts) are masked during their execution. When extended service is required, interrupt handlers "signal" a waiting interrupt task that, in turn, performs more complicated functions.

Interrupt Tasks are distinct tasks whose priority is associated with a hardware interrupt level. They are permitted to make an iRMX I system call. While an interrupt task is servicing an interrupt, interrupts of lower priority are not allowed to pre-empt the system.

Table 3 shows the iRMX I System Calls provided to manage interrupts.

Table 2. Memory Management System Calls

System Call	Function Performed
RQ$CREATE$SEGMENT	Dynamically allocates a memory segment of the specified size.
RQ$DELETE$SEGMENT	Deletes the specified segment by deallocating the memory.
RQGETPOOL$ATTRIBUTES	Returns attributes such as the minimum and maximum, as well as current size of the memory in the environment of the calling task's job.
RQGETSIZE	Returns the size (in bytes) of a segment.
RQSETPOOL$MIN	Dynamically changes the minimum memory requirements of the job environment containing the calling task.

Table. 3. Interrupt Management System Calls

System Call	Function Performed
RQSETINTERRUPT	Assigns an interrupt handler and, if desired, an interrupt task to the specified interrupt level. Usually the calling task becomes the interrupt task.
RQ$RESET$INTERRUPT	Disables an interrupt level, and cancels the assignment of the interrupt handler for that level. If an interrupt task was assigned, it is deleted.
RQGETLEVEL	Returns the number of the highest priority interrupt level currently being processed.
RQ$SIGNAL$INTERRUPT	Used by an interrupt handler to signal the associated interrupt task that an interrupt has occurred.
RQ$WAIT$INTERRUPT	Used by an interrupt task to SLEEP until the associated interrupt handler signals the occurrence of an interrupt.
RQ$EXIT$INTERRUPT	Used by an interrupt handler to relinquish control of the System.
RQ$ENABLE	Enables the hardware to accept interrupts from a specified level.
RQ$DISABLE	Disables the hardware from accepting interrupts at or below a specified level.

INTERRUPT MANAGEMENT EXAMPLE

Figure 3 illustrates how the iRMX I Interrupt System may be used to output strings of characters to a printer. In the example, a mailbox named 'PRINT' is used by all tasks in the system to queue messages to be printed. Application tasks put the characters in segments that are transmitted to the printer interrupt task via the PRINT Mailbox. Once printing is complete, the same interrupt task passes the messages on to another application via the FINISHED Mailbox so that an operator message can be displayed.

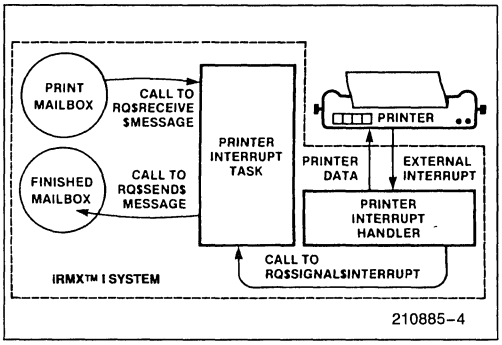


Figure 3. Interrupt Management Example

Basic I/O System

The Basic I/O System (BIOS) provides the direct access to I/O devices needed by real-time applications. The BIOS allows I/O functions to overlap other system functions. In this manner, application tasks make asynchronous calls to the iRMX I BIOS, and proceed to perform other activities. When the I/O request must be completed before an application can continue, the task waits at a mailbox for the result of the operation. Some system calls provided by the BIOS are listed in Table 4.

The Basic I/O System communicates with peripheral devices through device drivers. These device drivers provide the System with four basic functions needed to control and communicate with devices: Initialize I/O, Finish I/O, Queue I/O, and Cancel I/O. Using the device driver interface, users of non-standard devices may write custom drivers compatible with the I/O System.

The iRMX I Operating System includes a number of device drivers to allow applications to use standard USART serial communications devices, multiple CRTs and keyboards, bubble memories, diskettes, disks, a Centronics-type parallel printer, and many of Intel's iSBC and iSBX™ device controllers (see Table 8). If an application requires use of a non-standard device, users need only write a device driver to be included with the BIOS, and access it as if it were part of the standard system. For most common random-access devices, this job is further simplified by using standard routines provided with the System. Use of this technique ensures that applications can remain device independent.

Table 4. Key BIOS I/O Management System Calls

System Calls	Function Performed
RQAATTACH$FILE	Creates a Connection to an existing file.
RQACHANGE$ACCESS	Changes the types of accesses permitted to the specified user(s) for a specific file.
RQACLOSE	Closes the Connection to the specified file so that it may be used again, or so that the type of access may be changed.
RQACREATE$DIRECTORY	Creates a Named File used to store the names and locations of other Named Files.
RQACREATE$FILE	Creates a data file with the specified access rights.
RQADELETE$CONNECTION	Deletes the Connection to the specified file.
RQAGET$FILE$STATUS	Returns the current status of a specified file.
RQAOPEN	Opens a file for either read, write, or update access.
RQAREAD	Reads a number of bytes from the current position in a specified file.
RQASEEK	Moves the current data pointer of a Named or Physical file.
RQAWRITE	Writes a number of bytes at the current position in a file.
RQ$WAIT$IO	Synchronizes a task with the I/O System by causing it to wait for I/O operation results.

Multi-Terminal Support

The iRMX I Terminal Support provides line editing and terminal control capabilities. The Terminal Support communicates with devices through simple drivers that do only character I/O functions. Dynamic terminal reconfiguration is provided so that attributes such as terminal type and line speed may be changed without modifying the application or the Operating System. Dynamic configuration may be typed in, generated programmatically or stored in a file and copied to a terminal I/O connection.

The iRMX I Terminal Support provides automatic translation of control characters to specific control sequences for each terminal. This translation enables applications using standard control characters to function with non-standard terminals. The translation requirements for each terminal can be stored in terminal description files and copied to a connection, as described above.

Peripheral Device Drivers

Each device driver can be used to interface to a number of separate and, in some cases, different devices. The iSBC 215G Device Driver, supplied with the system, is capable of supporting the iSBC 215G Winchester Disk Controller, the iSBC 220 SMD Disk Controller, and the iSBX 218A Flexible Disk Controller (when mounted on an iSBC 215G board). Each device controller may, in turn, control a number of separate device units. In addition, each driver may control a number of like device controllers. This capability allows the use of large storage systems with a minimum of I/O system code to write or maintain.

Extended I/O System

The iRMX I Extended I/O System (EIOS) adds a number of I/O management capabilities to simplify access to files. Whereas the BIOS provides users with the basic system calls needed for direct management of I/O resources, many users prefer to have the system perform all the buffering and synchronization of I/O requests automatically. The EIOS allows users to access I/O devices without having to write procedures for buffering data, or to specify particular devices with constant device names.

By performing device buffering automatically, the iRMX I EIOS optimizes accesses to disks and other devices. Often, when an application task asks the System to READ a portion of a file, the System is able to respond immediately with the data it has read in advance of the request. Similarly, the EIOS will not delay a task for writing data to a device unless it is specifically told to, or if its output buffers are filled.

Logical file and device names are provided by the EIOS to give applications complete file and device independence. Applications may send data to the 'line printer' (:LP:) without needing to know which specific device will be used as the printer. This logical name may, in fact, not be a printer at all, but it could be a disk file that is later scheduled for printing.

The EIOS uses the functions provided by the BIOS to synchronize individual I/O requests with results returned by device drivers. Most EIOS system calls are similar to the BIOS calls, except that they appear to suspend the operation of the calling task until the I/O requests are completed.

File Management

The iRMX I Operating System provides three distinct types of files to ensure efficient management of both program and data files: Named Files, Physical Files, and Stream Files. Each file type provides access to I/O devices through the standard device drivers mentioned earlier. The same device driver is used to access physical and named files for a given device.

NAMED FILES

Named files allow users to access information on secondary storage by referring to a file with its ASCII name. The names of files stored on a device are stored in special files called directories in a hierarchical file structure.

The iRMX I BIOS uses an efficient format for writing the directory and data information into secondary storage. This structure enables the system to directly access any byte in a file, often without having to do additional I/O to access space allocation information. The maximum size of an individual file is 4.3 billion bytes.

EASE OF ACCESS

The hierarchical file structure is provided to isolate and organize collections of named files. To give operators fast and simple access to any level within the file tree, an ATTACHFILE command is provided. This command allows operators to create a logical name to a point in the tree so that a long sequence of characters need not be typed each time a file is referred to.

ACCESS PROTECTION

Access to each Named File is protected by the rights assigned to each user by the owner of the file. Rights to read, append, update, and delete may be

selectively granted to other users of the system. In general, users of Named Files are classified into one of two categories: User and World. Users are used when different programmers and programs need to share information stored in a file. The World classification is used when rights are to be granted to all who can use the system.

PHYSICAL FILES

Physical Files allow more direct device access than Named Files. Each Physical File occupies an entire device, treated as a single stream of individually accessible bytes. No access control is provided for Physical Files as they are typically used for such applications as driving a printing device, translating from one device format to another, driving a paper tape device, real-time data acquisition, and controlling analog mechanisms.

STREAM FILES

Stream Files provide applications with a method of using iRMX I file management methods for data that does not need to go into secondary storage. Stream Files act as direct channels, through system memory, from one task to another. These channels are very useful to programs, for example, wishing to preserve file and device independence allowing data sent to a printer one time, to a disk file another time, and to another program on a different occasion.

BOOTSTRAP AND APPLICATION LOADERS

Two utilities are supplied with the System to load programs and data into system memory from secondary storage devices:

The iRMX I Bootstrap is typically used to load the initial system from the system disk into memory, and begin its execution. Error reporting and debug switch features have been added to the Bootstrap Loader. When the Bootstrap Loader detects errors such as: "File Does Not Exist" or "Device Not Ready", an error message is reported back to the user. The debug switch will cause the Bootstrap Loader to load the system but not begin its execution. Instead the Bootstrap Loader will pass control to the monitor at the first instruction to be executed by the system.

The Application Loader is typically used by application programs already running in the system to load additional programs and data from any secondary storage device. The Human Interface layer, for example, uses the Application Loader to load the non-resident Human Interface Commands. The Application Loader is capable of loading both relocatable and absolute code as well as program overlays.

Human Interface

The flexibility of the interface between computer controlled machines and their users often determines the usability and ultimate success of the machines. Table 11 lists iRMX I Human Interface functions giving users and applications simple access to the file and system management capabilities described earlier. The process, interrupt, and memory management functions described earlier, are performed automatically for Human Interface users.

MULTI-USER ACCESS

Using the multi-terminal support provided by the BIOS, the iRMX I Human Interface can support several simultaneous users. The real-time nature of the system is maintained by providing a priority for each user, and using the event-driven iRMX I Nucleus to schedule tasks. High-performance interrupt response is guaranteed even while users interact with various application packages. For example, multi-terminal support allows one person to be using the iRMX I Editor, while another compiles a FORTRAN 86 or PASCAL 86 program, while several others load and access applications.

Each terminal attached to the iRMX I multi-user Human Interface is automatically associated with a user, a memory pool, and an initial program to run when the terminal is connected. This association is made using a file that may be changed at any time. Changes are effective the next time the system is initialized.

The initial program specified for each terminal can be a special application program, a custom Human Interface, or the standard iRMX I Command Line Interpreter (CLI).

Specifying an application program as a terminal's initial program makes the interface between operators and the computer system much simpler. Each operator need only be aware of the function of a particular application.

Specifying the standard iRMX I Human Interface CLI as the initial program enables users of the terminals to access all iRMX I functions. This CLI makes it easy to manage iRMX I files, load and execute Intel-supplied and custom programs, and submit command files for execution.

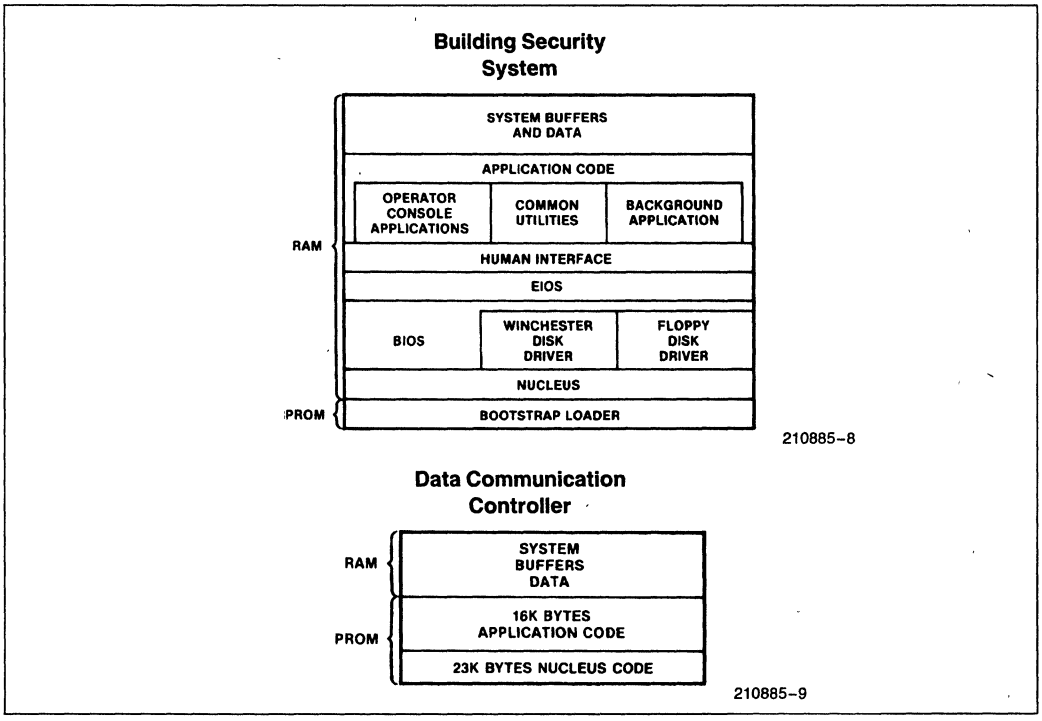


Figure 4. Typical iRMX® I Configurations

FEATURE OVERVIEW

The iRMX I Operating System is well suited to serve the demanding needs of real-time applications executing on complex microprocessor systems. The iRMX I System also provides many tools and featues needed by real-time system developers and programmers. The following sections describe features useful in both the development and execution environments. The description of each feature outlines the advantages given to hardware and software engineers concerned with overall system cost, expandability with custom and industry standard options, and long-term maintenance of iRMX I-based systems. The development environment features also describe the ease with which the iRMX I Operating System can be incorporated into overall system designs.

Execution Environment Features

REAL-TIME PERFORMANCE

The iRMX I Operating System is designed to offer the high performance, multi-tasking functions required by real-time systems. Designers can make use of VLSI devices such as the 8087, 80287 or 80387 Numeric Processor Extension.

Many real-time systems require high performance operation. To meet this requirement, all of iRMX I can be put into high-speed P(ROM). This approach eliminates the possibility of disk access times slowing down performance, while allowing system designers to take advantage of high performance memory devices.

CONFIGURABILITY

The iRMX I Operating system is configurable by system layer, and by system call within each layer. In addition all the I/O port addresses used by the System are configurable by the user. This flexibility gives designers the freedom to choose configurations of hardware and software that best suit their size and functional requirements. Two example configurations are shown in Figure 4.

Most configuration options are selected during system design stages. Others may be selected during system operation. For example, the amount of memory devoted to queues within a Mailbox can be specified at the time the Mailbox is created. Devoting more memory to the Mailbox allows more messages to be transmitted to other tasks without having to degrade system performance to allocate additional memory dynamically.

The chart shown in Table 6 indicates the actual memory size required to support these different configurations of the iRMX I System. Systems requiring only Nucleus level functions may require no more than 13 Kbytes for the Operating System. Other applications, needing I/O managment functions, may select portions of additional layers that fit their needs and size constraints.

This configurability also applies to the Terminal Handler, Dynamic Debugger, and System Debugger. The Terminal Handler provides a serial terminal interface in a system that otherwise doesn't need an I/O system. Either one of the debuggers need to be included only as debugging tools (usually only during system development).

MULTI-USER ACCESS

Many real-time systems must provide a variety of users access to system control functions and collected data. The iRMX I System provides easy-to-use support for applications to access multiple terminals. It also enables multiple and different users to access different applications concurrently.

Figure 5 illustrates a typical iRMX I application simultaneously supporting multi-terminal data collection and real-time environments. Shown is a group of terminals used by machinists on a shop floor to communicate with a job management program, a building security system that constantly monitors energy usage requirements, a system operator console capable of accessing all system functions, and a group of terminals in the Production Engineering department used to monitor job costs while developing new device control specifications instructions. The iSBC 544A Intelligent Terminal Interface supports multiple user terminals without degrading system performance to handle character I/O.

EXTENDABILITY

The iRMX I Operating System provides three means of extensions. This extendability is essential for support of OEM and volume end user value added features. This ability is provided by user-defined op-

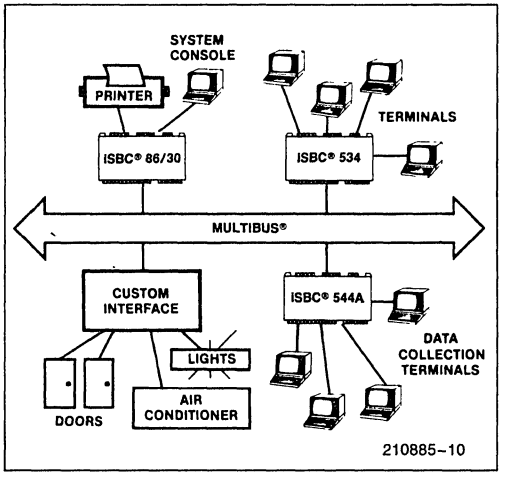


Figure 5. Multi-Terminal and Multi-User Real-Time System

Table 6. iRMX™ 86 Configuration Size Chart

System Layer	Min. ROMable Size	Max. Size	Data Size
Bootstrap Loader	1K	1.5K	6K*
Nucleus	10.5K	24K	2K
BIOS	26K	78K	1K
Application Loader	4K	10K	2K
EIOS	10.5K	12.5K	1K
Human Interface	22K	22K	15K
UDI	8K	8K	0
Terminal Handler	3K	3K	0.3K
System Debugger	20K	20K	1K
Dynamic Debugger	28.5K	28.5K	1K
Human Interface Commands			116K
Interactive Configuration Utility			308K

*Usable by System after bootloading.

Table 7. User Extension System Calls

System Call	Function Performed
RQ$CREATE$COMPOSITE	Creates a custom object built of previously defined objects.
RQ$DELETE$COMPOSITE	Deletes the custom object, but not the various objects from which it was built.
RQ$INSPECT$COMPOSITE	Returns a list of Token Identifiers for the component objects from which the specified composite object is built.
RQ$ALTER$COMPOSITE	Replaces a component object of a composite object.
RQ$CREATE$EXTENSION	Creates a new type of object and assigns a mailbox used for collecting these objects when they are deleted.
RQ$DELETE$EXTENSION	Deletes an extension definition.

erating system calls, user-defined objects (similar to Jobs, Tasks, etc.), and the ability to add functions later in the product life cycle. The modular, layered structure of the System easily facilitates later additions to iRMX I applications. User-defined objects are supported by the functions listed in Table 7.

Using standard iRMX I system calls, users may define custom objects, enabling applications to easily manipulate commonly used structures as if they were part of the original operating system.

EXCEPTION HANDLING

The System includes predefined exception handlers for typical I/O and parameter error conditions. The errors handling mechanism is both configurable and extendable.

SUPPORT OF STANDARDS

The iRMX I Operating System supports the many hardware and software standards needed by most application systems to ensure that commonly available hardware and software packages may be interfaced with a minimum of cost and effort. The iRMX I System supports the iSBC family of products built on the Intel MULTIBUS I (IEEE Standard 796), and a number of standard software interfaces such as the UDI and the common device driver interface (See Figure 6). The procedural interfaces of the UDI are listed in Table 9.

The Operating System includes support for the 8087 Numeric Data Processor and equivalent instructions and registers in the 80287 and 80387 Numeric Data

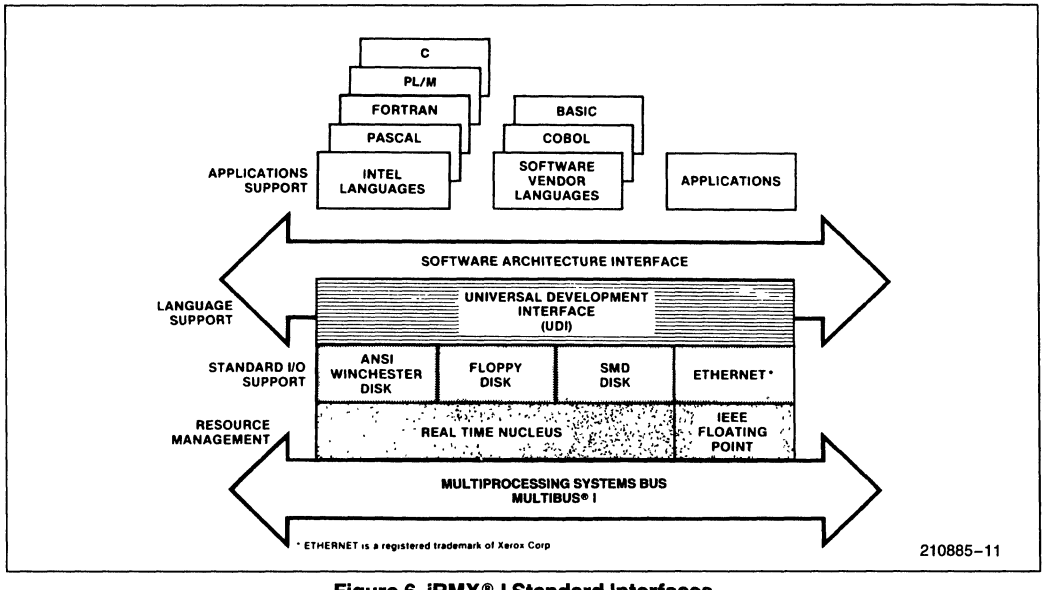


Figure 6. iRMX® I Standard Interfaces

Processors. Standards such as an Ethernet communication interface are supported by optional software packages available to run on the iRMX I System.

SPECTRUM OF CPU PERFORMANCE

The iRMX I Operating System supports a broad range of Intel processors. In addition to support for 8086, 8088, 80186 and 80188-based systems, the iRMX I system has been enhanced to support 80286/386, (16-bit, Real Address Mode)-based Systems. This new support enables the user to take advantage of the faster speed and higher performance of Intel's 80286 and 386 based microprocessors such as the iSBC 286/12 and iSBC 386/21 single board computers. By choosing the appropriate CPU, designers can choose from a wide range of performance options, without having to change application software.

COMPONENT LEVEL SUPPORT

The iRMX I System may be tailored to support specific hardware configurations. In addition to system memory, only an 8086, 8088, 80186, 80188, 80286, or 386 microprocessor, an 8259A Programmable Interrupt Controller (PIC), and either an 8253, 8254, or 82530 Programmable Interval Timer (PIT) are required as follows:

- 8086 and 8088 systems need either:
  - — 8253/4 PIT and 8259A PIC (master) or
  - — 80130 firmware (PIC is master)
- 80186 and 80188 systems where 186 PIC is slave, needs either:
  - — 8253/4 PIT and 8259A PIC (master) or
  - — 80130 firmware (PIC is master)

  where 186 PIC is master:
  - — Use 186 PIT for the system clock; no external PIT is needed
  - — Can use either 186 PIC (master) only or 8259A/80130 PIC (slave)
- 80286 systems need
  - — 8253/4 PIT and 8259A PIC

For systems requiring extended mathematics capability, an 8087, 80287, or 80387 Numeric Data Processor may be added to perform these functions up to 100 times faster than equivalent software. For applications servicing more than 8 interrupt sources, additional 8259A's may be configured as slave controllers.

BOARD LEVEL SUPPORT

The iRMX I Operating System includes device drivers to support a broad range of MULTIBUS I device controllers. The particular boards and types of devices supported are listed in Table 8. The device controllers all adhere to industry standard electrical and functional interfaces.

In addition to the on-CPU board terminal drivers, the iRMX I BIOS includes two iSBC board-level device drivers to support multiple terminal interfaces:

The iSBC 544A Intelligent Four-Channel Terminal Interface Device Driver provides support for multiple controllers each supporting up to four standard RS232 terminals. The iSBC 544A driver takes advantage of an on-board processor to greatly reduce the system processor time required for terminal I/O by locally managing input and output buffers. The iSBC 544A firmware provided with the operating system can offload the system CPU by as much as 75% when doing character outputting.

The iSBC 534 Four-Channel USART Controller Device Driver also provides support for multiple controller boards each supporting up to four standard RS232 terminals.

The RAM disk feature in iRMX I makes a portion of the memory address space look like a disk drive to the I/O system.

Table 8. Supported Devices

Function	Module/Device
Single Board Computer (Note 1)	iSBC® 86/C38, 86/05A, 86/12A, 86/30, 86/35, 186/03A, 186/51, 188/56, 286/10A, 286/12, 386/21, 386/31
Peripheral Controller	iSBC 186/03A SCSI, 208, 214, 215G, 220, 221, iSBX™ 217C, 218A
Terminal Controller/ Host Comm.	iSBC 188/56,534, 544A, 548, iSBX 351, 354
Network Controller	iSBC 552A, 186/51
Graphics	iSBX 279
Microprocessor (Note 1)	8086, 8088, 80186, 80188, 80286, 386
Math Coprocessor (Note 1)	8087, 80287, 80387
Serial Port	8251A, 8274, 82530
Interrupt Controller	8259A, 80130, 80186
Timer	8253, 8254, 80186
Parallel Port/ Line Printer	8255, iSBX 350
RAM Disk	SRAM, DRAM

NOTE:
1. Supports 16-bit, real address mode, 8086/8087 instruction set, functions and registers.

Development Environment Features

The iRMX I Operating System supports the efficient utilization of programming time by providing important tools for program development. Some of the tools necessary to develop and debug real-time systems are included with the Operating System. Others, such as language compilers, are available from Intel and from leading Independent Software Vendors.

LANGUAGES

The iRMX I Operating System supports 31 standard system calls known as the Universal Development Interface (UDI). Figure 6 shows the iRMX I standard interfaces to many compilers and language translators, including Intel's 8086 Macro Assembler and the Pascal 86, PL/M 86, FORTRAN 86 and C86 compilers.

Table 9. UDI System Calls

System Call	Function Performed
MEMORY MANAGEMENT	
DQ$ALLOCATE	Creates a Segment of a specified size.
DQ$FREE	Returns the specified segment to the System.
DQGETSIZE*	Returns the size of the specified Segment.
DQ$RESERVE$IO$MEMORY*	Reserves memory to OPEN and ATTACH files.
FILE MANAGEMENT	
DQ$ATTACH	Creates a Connection to a specified file.
DQ$CHANGE$ACCESS*	Changes the user access rights associated with a file or directory.
DQ$CHANGE$EXTENSION	Changes the extension of a file name in memory.
DQ$CLOSE	Closes the specified file Connection.
DQ$CREATE	Creates a Named File.
DQ$DELETE	Deletes a Named File.
DQ$DETACH	Closes a Named File and deletes its Connection.
DQ$OPEN	Opens a file for a particular type of access.
DQGETCONNECTION$STATUS*	Returns the current status of the specified file Connection.
DQ$FILES$INFO*	Returns data about a file Connection.
DQ$READ	Reads the next sequence of bytes from a file.
DQ$RENAME*	Renames the specified Name File.
DQ$SEEK	Moves the position pointer of a file.
DQ$TRUNCATE	Truncates a file.
DQ$WRITE	Writes a sequence of bytes to a file.
PROCESS MANAGEMENT	
DQ$EXIT	Exits from the current application job.
DQ$OVERLAY*	Causes the specified overlay to be loaded.
DQ$SPECIAL	Performs special I/O related functions on terminals with special control features.
DQ$TRAP$CC	Capture control when CNTRL/C is type.
EXCEPTION HANDLING	
DQGETEXCEPTION$HANDLER	Returns a pointer to the program currently being used to process errors.
DQ$DECODE$EXCEPTION	Returns a short description of the specified error code.
DQ$TRAP$EXCEPTION	Identifies a custom exception processing program for a particular type of error.
APPLICATION ASSISTANCE	
DQ$DECODE$TIME	Returns system time and date in binary and ASCII character format.
DQGETARGUMENT*	Returns the next argument from the character string used to invoke the application program.
DQGETSYSTEM$ID*	Returns the name of the underlying operating system supporting the UDI.
DQGETTIME*	Returns the current time of day as kept by the underlying operating system.
DQ$SWITCH$BUFFER	Selects a new buffer from which to process commands.

*Calls available only through the UDI.

Also included are other Intel development tools, language translators and utilities available from other vendors. The full set of UDI calls (which includes the URI system calls) is required to run a compiler.

These standard software interfaces (the UDI) ensure that users of the iRMX I Operating System may transport their applications to future releases of the iRMX I Operating System and other Intel and independent vendor software products. The calls avail able in the UDI are shown in Table 9.

The high performance of the iRMX I Operating System enhances the throughput of compilers and other development utilities.

TOOLS

Certain tools are necessary for the development of microcomputer applications. The iRMX I Human Interface includes many of these tools an non-resident commands. They can be included on the system disk of a application system, and brought into memory when needed to perform functions as listed in Table 10.

Table 10. Major Human Interface Utilities

Command	Function
BACKUP	Copy directories and files from one device to another.
COPY	Copy one or more files to one or more destination files.
CREATEDIR	Create a directory file to store the names of other files.
DIR	List the names, sizes, owners, etc. of the files contained in a directory.
ATTACHFILE	Give a logical name to a specified location in a file directory tree.
PERMIT	Grant or rescind user access to a file.
RENAME	Change the name of a file.
SUBMIT	Start the processing of a series of commands stored in a file.
SUPER	Change operator's ID to that of the System Manager with global access rights and privileges.
TIME	Set the system time-of-day clock.
VERIFY	Verify the structure of an iRMX I Named File volume, and check for possible disk data errors.

INTERACTIVE CONFIGURATION UTILITY

The iRMX I Operating System is designed to provide OEMs the ability to configure for specific system hardware and software requirements. The Interactive Configuration Utility (ICU) builds iRMX I configurations by asking appropriate questions and making reasonable assumptions. It runs on either an Intellec® Series IV development system or iRMX I development system that includes a hard disk and the UDI. Table 11 lists the hardware and support software requirements of different iRMX I development system environments.

Table 11. iRMX® Development Environment

Intellec Series IV: ASM 86 Assembler and Utilities PL/M 86 Compiler One hard disk and one diskette drive
iRMX I Development System: ASM 86 Assembler and Utilities PL/M 86 Compiler iSDM System Debug Monitor 640K Bytes of RAM 5M Byte On-Line Storage and one double-density diskette drive
SYSTEM 86/300, 286/300, or 386/300 Series: Microcomputer System Basic configuration

Figure 7 shows one of the many screen displayed during the process of defining a configuration. It shows the abbreviations for each choice on the left, a more complete description with the range of possible answers in the center, and the current (sometimes default) choice on the right. The bottom of the screen shows three changes made by the operator (lower case lettering), and a request for help on the Exception Mode question. In response to a request for help, the ICU displays an additional screen outlining possible choices and some overall system effects.

The ICU requests only information required as a result of previous choices. For example, if no Extended I/O System functions are required, the ICU will not ask any further questions about the EIOS. Once a configuration session is complete, the operator may save all the information in a file. Later when small changes are necessary, this file can be modified. A completely new session is not required.

REAL-TIME DEBUGGING TOOLS

The iRMX I Operating System supports two distinct debugging environments: Static and Dynamic. While the iRMX I Operating System does support a multi-user Human Interface, these real-time debugging aids are usually most useful in a single-user environment where modifications made to the system cannot affect other users.

System Debugger

The static debugging aid is the iRMX I System Debugger. This debugger is an extension of the iSDM System Debug Monitor. The System Debugger provides static debugging facilities when the system hangs or crashes, when the Nucleus is inadvertently overwritten or destroyed, or when synchronization requirements prevent the debugging of certain tasks. The System Debugger stops the system and allows you to examine the state of the system at that instant, and allows you to:

— Identify and interpret iRMX I system calls.

— Display information about iRMX I objects.

— Examine a task's stack to determine system call history.

iRMX® I Dynamic Debugger

The iRMX I Dynamic Debugger runs as part of an iRMX I application. It may be used at any time during program development, or may be integrated into an OEM system to aid in the discovery of latent errors. The Dynamic Debugger can be used to search for errors in any task, even while the other tasks in the system are running. The iRMX I Dynamic Debugger communicates with the developer via a terminal handler that supports full line editing.

PARAMETER VALIDATION

Some iRMX I System Calls require parameters that may change during the course of developing iRMX I applications. The iRMX I Operating System includes an optional set of routines to validate these parameters to ensure that correct numeric values are used and that correct object types are used where the System expects to manipulate an object. For systems based only on the iRMX I Nucleus, these routines may be removed to improve the performance and code size of the System once the development phase is completed.

START-UP SYSTEMS

Three ready-to-run start-up systems are included in the iRMX I Operating System package for 8086, 80286, and 386-based MULTIBUS I systems. These iRMX I start-up systems are fully configured, iRMX I Operating Systems ready to be loaded into memory by the Bootstrap Loader. The start-up systems are configured to include all of the system calls for each layer and most of the features provided by iRMX I software. iRMX I start-up systems include UDI support so that users may run languages such as PL/M-86, Pascal, FORTRAN, and software packages from independent vendors.

The start-up system for the 8086 processor is configured for Intel SYSTEM 86/300 Series microcomputers with a minimum of 384K bytes of RAM. The following devices are supported.

- iSBC 215G/iSBX 218 or iSBC 215G/iSBX 218A or iSBC 214
- Line Printer
- 8251A Terminal Driver
- iSBC 544A, Terminal Driver

```
Nucleus
(ASC)     All Sys Calls [Yes/No]                          Yes
(PV)      Parameter Validation [Yes/No]                   Yes
(ROD)     Root Object Directory Size [0-0FF0h]            0014H
(MTS)     Minimum Transfer Size [0-0FFFFH]                0040H
(DEH)     Default Exception Handler [Yes/No/Deb/Use]      Yes
(NEH)     Name of Ex Handler Object Module [1-32 chs]
(EM)      Exception Mode [Never/Program/Environ/All]      Never
(NR)      Nucleus in ROM [Yes/No]                         No

Enter Changes [Abbreviations ?/ = new-value]:   ASC = N
:pv = no
:rod = 48
:em?
```

Figure 7. ICU Screen for iRMX® I Nucleus

The start-up system for the 80286 processor is configured for Intel SYSTEM 286/300 Series microcomputers with a minimum of 512 Kbytes and a maximum of 896 Kbytes of RAM. The following devices are supported.

- iSBC 208
- iSBC 215G/iSBX 218 or iSBC 215G/iSBX 218A
- Line Printer for iSBC 286/1X
- 8274 Terminal Driver
- iSBC 544A Terminal Driver

A start-up system is also provided for 386-based designs.

The systems will run without hardware or software configuration changes and can be reconfigured on a standard system with at least 512 Kbytes of RAM. Definition files are also included for iSBC 186/03A, 186/51 and 188/56 configurations.

This start-up system may be used to run the ICU (if a Winchester disk is attached to the system) to develop custom configurations such as those pictured in Figure 5. As shipped, the Human Interface supports a single user terminal. However, the Start-up System terminal configuration file may be altered easily to support from two to five users.

SPECIFICATIONS

Supported Software Products

R86ASM86	8086 Assembler and Utilities
R86C86	C 86 Compiler
R86PAS86	PASCAL 86 Compiler
R86FOR86	FORTRAN 86 Compiler
R86PLM86	PL/M 86 Compiler
iRMX 864	AEDIT Screen-oriented Editor

Supported Hardware Products

COMPONENTS

8086 and 8088 Microprocessors

80186 and 80188 Microprocessors

80286/386 Microprocessors (16-bit, Real Address Mode Only)

8087 Numeric Data Processor Extension

80287/387 Numeric Data Processor Extension (8087 Functions and Registers)

8253 and 8254 Programmable Interval Timers

8259A Programmable Interrupt Controller

8251A USART Terminal Controller

8255 Programmable Parallel Interface

8274 Terminal Controller

82530 Serial Communications Controller

iSBC® MULTIBUS BOARD AND SYSTEM PRODUCTS

iSBC 86/C38, 86/12A, 80/05A, 86/30, 86/35, and 88/40A Single Board Computers

iSBC 186/03A Single Board Computer

iSBC 186/51 Ethernet Controller

iSBC 188/56 Communications Controller

iSBC 286/10A and 286/12 Single Board Computers (Real Address Mode only)

iSBC 386/21 and 386/31 (16-bit, Real Address Mode only)

iSBC 208 Diskette Controller

iSBC 214 and 215(G) Winchester Disk Controllers

iSBX 218A Flexible Diskette Multi-Module Controller

iSBC 220 SMD Disk Hard Controller

iSBC 221 Disk Controller

iSBC 534 4-Channel Terminal Interface

iSBC 544A Intelligent 4-Channel Terminal Interface and Controller

iSBC 548 Intelligent 8-Channel Terminal Controller

iSBC 552A Ethernet Controller

iSBX 350 Parallel Port (Centronics-type Printer Interface)

iSBX 351 and 354 Serial Communications Port

iSBX 279 Graphics Subsystem

SYSTEM 86/300 Family

SYSTEM 286/300 Family

SYSTEM 386/300 Family

USER MANUALS

The iRMX I Operating System is provided with one five volume set of reference manuals:

Volume I	iRMX I INSTALLATION AND PROGRAMMER'S GUIDES
Volume II	iRMX I OPERATING SYSTEM USER GUIDES
Volume III	iRMX I SYSTEM CALLS
Volume IV	iRMX I OPERATING SYSTEM UTILITIES
Volume V	iRMX I INTERACTIVE CONFIGURATION UTILITY REFERENCE GUIDE

Additional sets of manuals may be ordered.

Training Courses

Training courses are available on the iRMX I Operating System, Intel languages, and Intel microprocessor architectures.

ORDERING INFORMATION

iRMX I Operating System development software is available on both 5¼" and 8" iRMX-format diskettes. The software includes one set of user manuals and 90 days of initial support. This support includes: "TIPS" Technical Information Phone Service; software updates that occur during the support period; monthly ";Comments" magazine and quarterly Troubleshooting Guide; Software Problem Report Service; and membership in the Insite User's Program Library.

Please contact your local Intel Sales Office or authorized distributer for product order codes.

LICENSING

Before iRMX I software will be shipped, a customer must sign (or have already signed) Intel's Software License Agreement (SLA). Once the SLA is signed, the customer is licensed to use the iRMX software for application development. Customers who want to "incorporate" portions of the iRMX I Object Code in an application, will have to sign an Incorporation License which clearly spells out the terms and conditions under which incorporations can be made. Contact your local Intel office for more information and for appropriate licensing.

STANDARD UNIX SUPPORTS MULTIBUS® SYSTEM ARCHITECTURES*

The UNIX System V/386 operating system now delivers full support for Intel's MULTIBUS II System Architecture. Intel has built a complete System V/386 product family providing OEMs, System Integrators and Computing Manufacturers with industry standard UNIX, OpenNET™ networking, system hardware, and Ada development tools for the MULTIBUS II System Architecture. Together, these elements deliver a rich, complete UNIX development environment. System V/386 family products are also available for MULTIBUS I system architectures.

The UNIX System V operating system is the core of the family. It provides portability of applications and systems programs from one hardware architecture to another, and robust development support. Integrated with Intel's System 520 MULTIBUS II system, the UNIX operating system provides a powerful open system platform. The System V/386 product family is ideal as a base for developing custom multi-user systems, applications and multi-purpose network servers requiring open system configurability and flexible packaging.

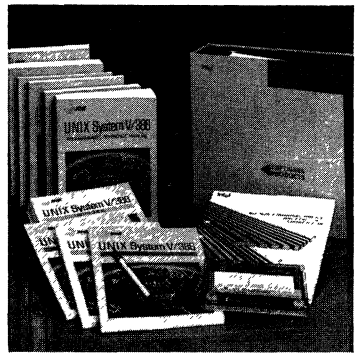
FEATURES:

- Standard UNIX System V/386, Release 3.2
- System V Interface Definition, Issue 2 (SVID2) certified; de facto 386 Application Binary Interface conformant
- Native 32-bit, 386™ UNIX port
- 80287 and 387™ numeric coprocessor support
- Integrated support for MULTIBUS II System Architecture (MSA) Standards
- UNIX System V STREAMS & TLI-based System V/OpenNET networking
- UNIX hosted and targeted Ada-386 compilation system support
- Complete documentation
- Worldwide service and support

September, 1989
Order Number 280692-001

SYSTEM V/386 OPERATING SYSTEM

STANDARD UNIX OPERATING SYSTEM AND TOOLS



Intel has worked with AT&T and others to produce a powerful UNIX System V technology base for the 386 microprocessor and MULTIBUS systems. The System V/386 package contains the complete, standard UNIX operating system and development environment. The C Programming Language Utilities, System Generation Utilities, the Advanced Programming Utilities, and Productivity tools are included in the product, in addition to all basic, runtime facilities.

STANDARD UNIX SYSTEM V FEATURES

Ongoing enhancements and improvements to System V/386 releases included are executable shared libraries, demand paging/ virtual memory, reliable signals, the 2KB file system, file and record locking, media independent UUCP, extensive terminal support utilities, the terminfo database and tools, and multiple installation and system administration tools. Support for international environments, including full support for 8-bit code sets, alternate date and time formats, and alternate character and conversion sets is also provided. In addition, MULTIBUS II MSA support has been integrated into AT&T's UNIX System V/386 source code package as a standard feature, available from both AT&T and Intel. Conformance to final IEEE POSIX standards will be provided in future releases.

MULTIBUS II SYSTEM ARCHITECTURE SUPPORT

Intel's System V/386 product comes with complete support for the MULTIBUS II System Architecture (MSA) in the UNIX kernel. MSA delivers open system interface and protocol standards that build on and extend the basic MULTIBUS II (IEEE/ANSI 1296) bus standard. MSA specifications define diagnostics control, built-in self-test, system initialization and boot loading, board configuration, transport message passing, and an OSI Transport Interface. The UNIX system developer has available an application level interface and a kernel driver level interface to MULTIBUS II transport message passing and interconnect space. Use of these MSA interfaces masks the system bus specifics from the system and application developer, facilitating system integration.

OSI STANDARD TRANSPORT

OSI transport services support the use of the MULTIBUS II backplane as an ultra-fast network. Multiple peer-to-peer 386 application processors can operate as independent networked UNIX "systems" over a single MULTIBUS II Parallel System Bus (PSB). Two UNIX application processors, each with its own hard disk, can be installed in a UNIX System 520, and each processor provides a TLI-based application interface for complete, transparent, inter-CPU and application communications. Larger system configurations with more processor boards can also be built using these standard interfaces. System V/OpenNET utilizes TLI to provide transparent, distributed file sharing, file transfer and Virtual Terminal facilities for networked UNIX systems.

SYSTEM V/386 "ABI" FOR UNIX BINARY APPLICATION PORTABILITY

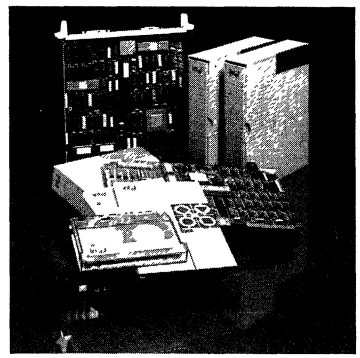
The creation of a defacto standard ABI (Application Binary Interface) for the Intel 386 architecture makes machine-independent execution of UNIX/386 binary applications a reality. This opens up the System V/386 operating system, so application developers no longer have to port applications and test them on different System V/386 machines. It makes possible a world of off-the-shelf, shrink-wrapped, UNIX binary applications for any and all 386 machines supporting the ABI standard.

SOURCE RELEASES

The complete MULTIBUS I and II source base is available to AT&T and Intel-licensed customer sites. Device driver source licenses and code are also available independent of the complete UNIX source base products.

SYSTEM V/OpenNET

COMPLETE OpenNET LAN SOLUTION FOR UNIX SYSTEM V/386

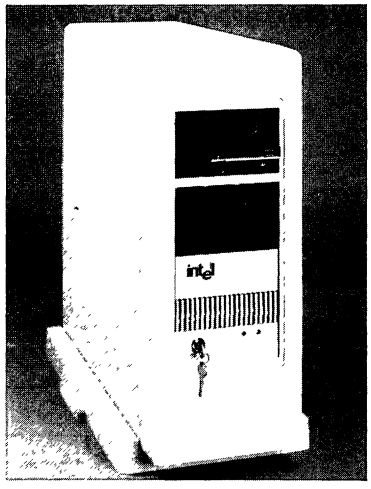


UNIX STREAMS and TLI (Transport Level Interface) facilities are a standard part of Intel System V/386. Intel has developed and is delivering a STREAMS/TLI-based version of OpenNET for MULTIBUS UNIX systems, called System V/OpenNET. It provides interoperation and communication with all OpenNET family products, including iRMX-NET, MSNET(PCLINK2), VMS*NET, XNXNET and iNDX OpenNET. The product comes packaged as a complete hardware/software solution including an Ethernet communications controller board, mail, virtual terminal, print spooler, nameserver interface library (NSI), and network management facilities. Support for TCP/IP networking is available from independent sources. System V/OpenNET is available for both MULTIBUS I and MULTIBUS II architectures.

THE UNIX SYSTEM 520

AN OEM MULTIBUS II DISTRIBUTED SYSTEM

The Intel System 520 is part of a family of customizable MULTIBUS II multiprocessor platforms designed for OEMs demanding the highest levels of flexibility, configurability and compatibility. The UNIX System 520 makes full use of Intel's standard MSA, the Intel 386 microprocessor and the UNIX System V standard. This powerful, compact MULTIBUS II system is available as a complete, System V/386 networked development system, or as an expandable, configurable OEM system or server. The System 520 allows users to add to the basic configuration or to purchase and integrate specific system contents separately.



STANDARD FEATURES:

- High performance Intel 386 MULTIBUS II OEM system
- System V/386, Release 3.2 multi-user operating system
- Complete UNIX (SVID2) Software Development Extension
- One 386 application processor
- 386 microprocessor-based SCSI disk controller
- 186/410 six-channel serial I/O controller
- 186/530 Ethernet controller
- iSBX™ 279 Hardware Window console controller (no graphics)
- Easy system expansion via Intel's MULTIBUS II System Architecture (MSA) & iSBC® board family
- OpenNET transparent remote file sharing & virtual terminal

SYSTEM V/386 SELF-TARGETED ADA*-386 COMPILATION SYSTEM

UNIX HOSTED ADA-386 DEVELOPMENT

Intel's System V/386 Self-Targeted Ada-386 Compilation System comprises a rich set of Ada language tools for the programmer wanting to develop Ada applications for the Intel System 520. This UNIX hosted, self-targeted and validated Ada toolset contains the Compiler & Library Tools, the Global Optimizer, the COFF Linker, an Ada Execution Environment, an Ada Symbolic Debugger, as well as other development environment tools for handling cross-referencing, source dependency, and source formatting. The compiler and its tools create a flexible, project-oriented development environment for commercial, industrial, and military applications.

ORDERING INFORMATION

BACKED BY INTEL MANUFACTURING AND CUSTOMER SUPPORT

All Intel software and hardware products are fully supported by Intel's worldwide staff of trained service and support engineers. Intel also provides system engineering and field applications consulting services, worldwide training workshops, a full range of maintenance services, and a software support hotline. Custom board and system configurations, as well as custom manufacturing, can also be provided by Intel. Every product includes a standard 90-day warranty replacement guarantee. Selected products include an extended warranty package. The complete UNIX System 520 development package (SYS520R1DKITSV) is backed by a full, one-year service and support package. See specific product literature for more details.

LICENSING

Each copy of UNIX is licensed for use on a single system, and Intel provides licenses for copying and distributing MULTIBUS I/II versions of System V/386. Licensing for the use of source code and distribution of binary, derived works is also available.

SYSTEM 520 CONFIGURATION/OPTIONS – TABLE 1

Product	SBC CSM Cent. Serv.	SBC 386/120 w/4 MB	SBC 386/258 w/4 MB	SBX 279	SBC 186/410 Term. Ctlr.	SBC 186/530 Enet Ctlr.	380 MB Hard Disk (SCSI)	1X0MB Hard Disk (SCSI)	125/150 MB ¼" Tape (SCSI)	1.2 MB Floppy w/Adapt (SCSI)
System 520 OEM* Base Plus I/O	✓	✓	✓	✓	✓	✓	✓	Add-[1] In.	✓	✓
System 520 OEM Base*	✓		✓					Add-[1] In.		✓
System 520** Development System	✓	✓	✓	✓	✓	✓	✓	Add-[1] In.	✓	✓

*Available in either floorstand or tabletop configurations
**iRMX and UNIX System V/386 options available
[1]Contact Intel for configuration availability information.

SYSTEM V/386 FAMILY ORDER CODES

The UNIX System 520 Products:

SYS520R1DKITSV	System V/System 520 MULTIBUS II Development System (h/w and s/w, floorstand)
SYP520R1BP	System 520 MULTIBUS II Base Plus OEM System-(h/w only, floorstand)
SYP520R1BPT	System 520 MULTIBUS II Base Plus OEM System-(h/w only, tabletop)
SYP520R1B	System 520 MULTIBUS II Base OEM System-(h/w only, floorstand)
SYP520R1BT	System 520 MULTIBUS II Base OEM System-(h/w only, tabletop)

System V/386, Release 3.2 Products:

SYSTEMV386II	MULTIBUS II UNIX binary tape and documentation
SYSTEMV386I	MULTIBUS I UNIX binary tape and documentation
SYSTEMVSRC	MULTIBUS II/I UNIX source tape and documentation

System V/OpenNET Products[1]:

pSVNET530KIT	SV-OpenNET with iSBC 186/530 on MULTIBUS II architecture
pSVNET552AKIT	SV-OpenNET with iSBC 552A on MULTIBUS I architecture

System V/386 Self-Targeted Ada-386 Products:

U386ADA386SW	UNIX hosted, Self-Targeted Ada-386 Compilation System for System 520 MULTIBUS II OEM Systems
U386ADA386SS	UNIX hosted, Self-Targeted Ada-386 Software

[1]These products are manufactured by Intel Puerto Rico.

For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada).

intel®

APPLICATION NOTE

AP-405

May 1987

Software Migration From iRMX® 86 to iRMX® 286

MAYNE MIHACSI
OSD Technical Marketing

Order Number 280608-001

INTRODUCTION

The iRMX® 286 operating system represents the evolution of the iRMX® 86 operating system to the protected-mode 80286 and 80386 microprocessors. Therefore, the iRMX 286 operating system has most of the same features of its 8086 counterpart.

Many Intel customers are going to migrate their software from iRMX 86 to iRMX 286. Most customers should be pleasantly surprised at the ease of migration between the two operating systems. This compatibility between the two operating systems was a key objective of the iRMX 286 project. Thus in the majority of cases, an iRMX user should encounter no changes or only trivial changes when porting their software to iRMX 286. In the other cases, iRMX users with a little patience, work, and the help of this paper, should quickly have their application running on iRMX 286.

Before reading this migration note, it is strongly suggested that readers acquaint themselves with the fundamentals of the 80286 architecture.

iRMX® 286 SYSTEM ARCHITECTURE

There are inherent differences between iRMX 86 and iRMX 286 due to the differences in microprocessor architectures. To take advantage of some unique 80286 features additional system calls have been added in the iRMX 286 operating system. These new calls can be identified by an RQE$ in their preface, with the E denoting "extended", to take advantage of the 80286's 16MB addressability.

Figure 1 lists the differences for each layer of iRMX 286.

iRMX® Layer	iRMX® 286 Changes
Nucleus	– 16MB address space – New hardware traps – Descriptor management – Privilege management – Round robin scheduling – Interrupt management – New calls
BIOS	– Memory buffer protection
EIOS	– New calls – Memory buffer protection
Application Loader	– Only loads 80286 OMF records – Only loads STL modules – New calls
Human Interface	– Enhanced CLI – New commands
UDI	– New calls
Bootstrap Loader	– New third stage interface
ICU	– Single stage ICU

Figure 1. iRMX® 286 Architectural Differences

iRMX® 286 NUCLEUS

16 Megabyte Address Space

Today's applications have pushed beyond the 1MB memory limitation of the 8086 architectures. Many Intel customers have chosen iRMX 286 simply because of its ability to address 16MB of memory. While the 80286 architecture allows for accessing 24 physical address lines, to yield 16MB physical and 16MB virtual addressability, the operating system is not automatically allowed the same abilities. As further generations of CPUs become available and memory becomes cheaper, operating systems will strive toward hardware independence. One method used is accessing memory logically, not physically. In the iRMX 286 operating system all memory addresses are logical addresses available via a descriptor table. A logical address may be thought of as a pointer consisting of a selector and an offset. The selector will point to an entry in a descriptor table containing the 24-bit physical address. Therefore, tokens are affected by containing selectors that reference an entry in the descriptor table. No longer do tokens contain the physical address of an object.

New Hardware Traps

Because the 80286 processor detects several types of exceptions and interrupts from exceptions, iRMX 286 also alerts programs generating these exception conditions. These hardware traps will be generated from the following conditions:

INTERRUPT VECTOR	FUNCTION
8	Double exception
9	Processor extension segment overrun
10	Invalid task state segment *
11	Segment not present *
12	Stack segment overrun or not present
13	General protection

*Seldom seen

Users porting iRMX 86 code to iRMX 286 should be aware that the working code in iRMX 86 might still contain errors that will be "trapped" in iRMX 286.

Descriptor Management

While the 80286 CPU is in Protected Virtual Address Mode (PVAM), all application programs deal exclusively with logical addresses. That is, the programs do not directly access actual physical addresses generated by the processor. Instead, a memory-resident table, called a descriptor table, records the mapping between the segment address and the physical locations allocated to each segment. Whenever the 80286 decodes a logical address, translating a full 32-bit pointer into a corresponding 24-bit physical address, it implicitly references one of several descriptor tables. One table is called the Global Descriptor Table (GDT) and provides a complete description of the global address space. Another table is provided, the Local Descriptor Table (LDT), to describe the local address space for one or more tasks. To the application programmer, much of the internal operation and management of the descriptor tables are transparent. However, the systems programmer will need to manage the descriptors to:

A. Gain access to undefined or allocated memory areas, and

B. Add device drivers to the system.

Several new calls were added to help manage descriptor tables:

1. RQE$CREATE$DESCRIPTOR
2. RQE$CHANGE$DESCRIPTOR
3. RQE$DELETE$DESCRIPTOR

For the applications programmer several features are available in iRMX 286.

1. Of the maximum 8K objects available, all are indexed in the GDT with the operating system using the LDT.
2. While using an iRMX 86-style task switch, iRMX 286 runs as one 80286 hardware task.

Privilege Management

Some means of protection is required to prevent programs from improperly using code or data that belongs to the operating system. The four privilege levels of the 80286 are numbered from 0 to 3, where 0 is the most trusted level. The privilege level is an attribute assigned to all segments in a hierarchical fashion. Operating system code and data segments are placed at the most privileged level (0) which is where iRMX 286 operates. (See Figure 2.)

The privilege levels apply to tasks and three types of descriptors:

1. Main memory segments
2. Gates
3. Task state segments (not used in iRMX 286)

Of particular interest to discussions concerning iRMX 286 is the gate descriptor and its usage in application programs.

Of the four types of gates in the 80286 processor, iRMX 286 uses call gates. Once invoked, control is transferred using only the selector portion. This address becomes fixed, allowing any program to invoke another. This prohibits tasks that have not used these entry points from jumping into the middle of the operating system. The use of gates is fundamental to the 80286 architecture and is reflected in other areas of iRMX 286.

All iRMX 286 system calls go through a call gate in order to invoke a given service procedure. In the iRMX 86 operating system, all calls were through software interrupts, invoking an operating system extension handler, then finally the service procedure. For iRMX code that was written for the iRMX 86 operating system, this will have little impact until it comes time to build the system, unless a conflict exists between the old and new nucleus calls. (See next section.) Analogous to the iRMX 86 operating system having a software interrupt at each level, iRMX 286 possesses call gates for each system call at each layer of the operating system, eliminating the need for an operating system extension handler. Call gates can be specified through system calls and the Interactive Configuration Utility (ICU). (See the example for RQESETOS$ EXTENSION.)

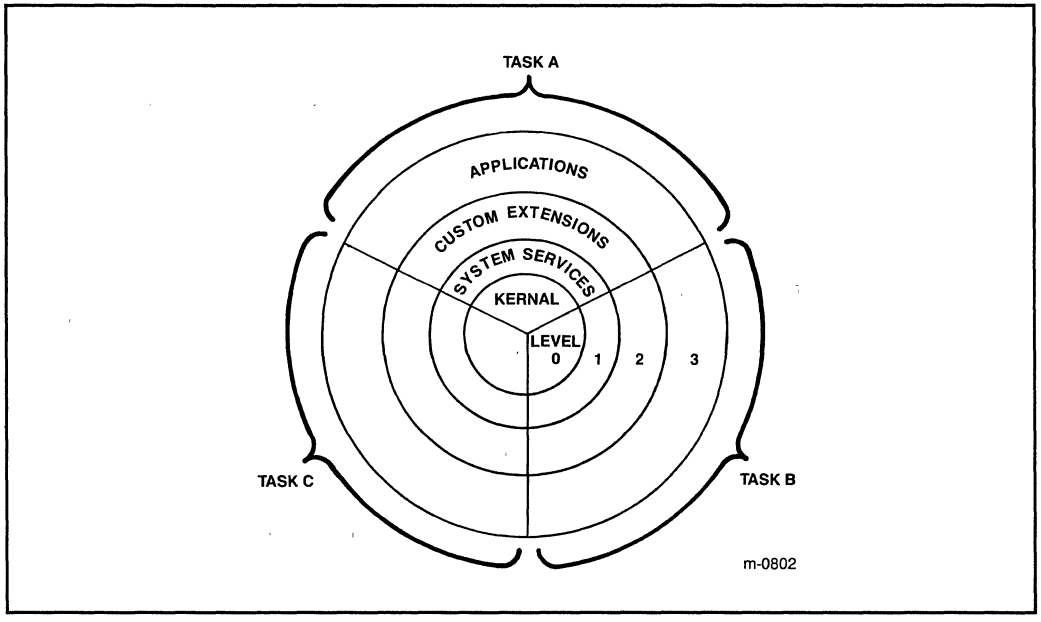
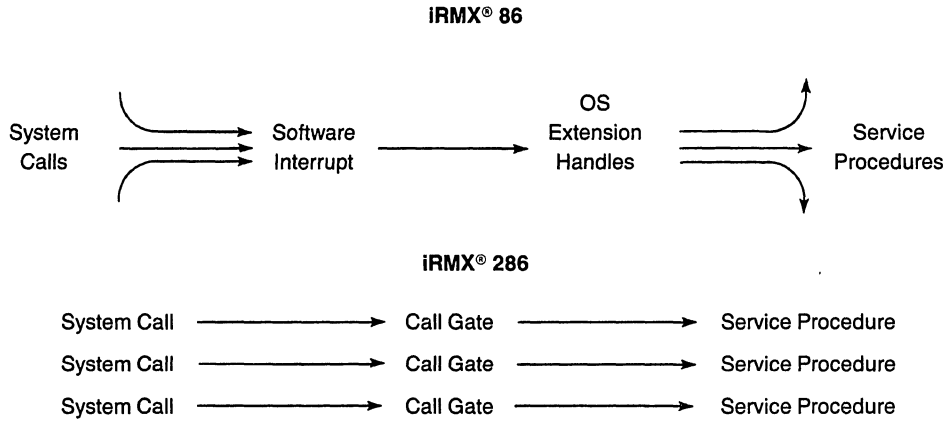


Figure 2. Example Privilege Level Assignments



Call Gates vs. Software Interrupts

Round Robin Scheduling

The iRMX 286 operating system schedules tasks based upon tasks competing for CPU resources. To prevent the occurrence of one or more tasks waiting indefinitely, round robin scheduling is available on the iRMX 286 operating system. One area that could benefit from this scheduling scheme is multi-user environments.

Round robin scheduling will permit equal priority tasks a finite time they may have control of the processor. Once the time expires, the task with the same priority and ready will gain CPU control. Hardware interrupts and higher-priority tasks can still bump any of the lower-priority tasks from running. This scheme allows all equal priority tasks an opportunity to execute.

This scheduling is determined in the "nucleus" screen of the Interactive Configuration Utility (ICU). (See the iRMX 286 Interactive Configuration Utility Reference Manual for details.)

Interrupt Management

In the iRMX 286 operating system interrupt management has changed. In the iRMX 86 operating system an interrupt vector table contains the address of an interrupt handler. In the iRMX 286 operating system this table has been called the Interrupt Descriptor Table (IDT) and is very similar to the GDT and LDT, except that it is referred to only when an interrupt occurs. Interrupt addresses can be entered into the IDT when using the iRMX 286 SET$INTERRUPT nucleus system call. Entering interrupts is still identical for both operating systems, however, with PL/M 286 not having a SET$INTERRUPT built in, interrupts have become easier to use. (See the section on PL/M 286.) The following is a description of the allocated interrupt entries. (Also see the section on BUILD 286 for another way to set interrupts.)

Entry Number	iRMX® 286 Interrupt Allocation Description
0	Divide by zero
1	Single step (used by iSDM™ 286)
2	Power failure (non-maskable interrupt, used by iSDM™ 286)
3	One-byte interrupt instruction (used by iSDM 286)
4	Interrupt on overflow
5	Run-time array bounds error
6	Undefined opcode
7	NPX not present/NPX task switch
8	Double fault
9	NPX segment overrun
10	Invalid TSS
11	Segment not present
12	Stack exception
13	General protection
14-15	<Reserved>
16	NPX error
17-55	<Reserved>
56-63	8259A PIC master
64-127	8259A PIC slaves
128-255	* Available to users *

New Calls

GENERAL RULES

IMPORTANT

Here are some general rules to apply.

1. All iRMX 286 system calls beginning with RQ$... are 100% compatible with iRMX 86.
2. All iRMX 286 system calls beginning with RQE$... are new to iRMX and exist only in iRMX 286.
   a. All iRMX|86 system calls beginning with RQ$... for which there is a like iRMX 286 system call beginning with a RQE$... use the function procedure of the RQE$... call.
3. All iRMX 286 system calls and user extensions use call gates.
4. All iRMX 86 BIOS, EIOS, and loader calls are 100% compatible with iRMX 286 calls.
5. All objects are identified by 16-bit tokens which represent an entry in the Global Descriptor Table (GDT).
6. The iRMX 286 system call RQESETOS$EXTENSION must be used in place of RQSETOS$EXTENSION. This call dynamically attaches an operating system extension to a call gate.

A few specific system calls merit further discussion.

RQESETOS$EXTENSION

This system call as mentioned in Rule 6 above will find the following usage.

```
DECLARE
                .
                . Typical PL/M 286 statements
                .
MY$OS$EXT: PROCEDURE EXTERNAL;

                .
                . Typical PL/M 286 statements
                .
END MY$OS$EXT;

CALL RQE$SET$EXTENSION (0141H, @MY$OS$EXT, @STATUS);
```

Where: 0141H represents the entry number of the call gate from the GDT. This parameter is named GATE$NUMBER.

: @MYOSEXT represents the pointer to first instruction of MYOSEXT. This parameter is named START$ADDR.

: @STATUS represents a pointer to a word containing the condition code for this call. This parameter is named EXCEPT$PTR.

A user-written operating system extension can also be attached to a call by the Interactive Configuration Utility (ICU).

Example of an ICU screen:

```
OS Extension
(GSN) GDT slot number                    [0140H-01FFEH] 0141H
(OCN) entry point name                   [1-45 characters] MY$OS$EXT

Enter changes [Abbreviation ?/= NEW_VALUE]:
Do you need any more O.S. extensions?
```

This causes the GDT slot 141H to be configured as a call gate whose entry point is MYOSEXT.

RQE$CREATE$JOB

This call is an example of Rule 2a where two calls perform nearly the same function. In this case the extended versions of POOL$MIN and POOL$MAX parameters are DWORDS instead of WORDS. This is to allow a memory pool of up to 16MB for tasks and objects. While RQ$CREATE$JOB will create a memory pool of up to 1MB, it will use the same function procedure as RQE$CREATE$JOB. This is possible because the RQ$CREATE$JOB interface procedure changes the word pool parameters to DWORDS by padding them with zeros, then calling the RQE$CREATE$JOB function procedure.

RQ$CREATE$SEGMENT

This call's first parameters, SIZE, yields a different value than in iRMX 86.

In iRMX 86: Segment = RQ$CREATE$SEGMENT (SIZE, EXCEPT$PTR);

Where: SIZE is a word containing the size, in bytes, of the requested segment in MULTIPLES OF 16 BYTES.

In iRMX 286: SEGMENT = RQ$CREATE$SEGMENT (SIZE, EXCEPT$PTR);

Where: SIZE is a word containing the actual memory size in bytes.

RQGETPOOL$ATTRIB

In this case more parameters have been added.

In iRMX 86: RQGETPOOL$ATTRIB (ATTRIB$PTR, EXCEPT$PTR);

Where: ATTRIB$PTR is a pointer to the following structure.

```
Structure (POOLMAX WORD,
           POOLMIN WORD,
           INITIAL$SIZE WORD,
           ALLOCATED WORD,
           AVAILABLE WORD);
```

In iRMX 286: RQEGETPOOL$ATTRIB has a different structure though everything else is the same

```
Structure (TARGET$JOB TOKEN,
           PARENT$JOB TOKEN,
           POOLMAX DWORD,
           POOLMIN DWORD,
           INITIAL$SIZE DWORD,
           ALLOCATED DWORD,
           AVAILABLE DWORD,
           BORROWED DWORD);
```

RQSETINTERRUPT

Users should also be aware of the following when using this call in iRMX 286. When specifying interrupts in iRMX 286, a special descriptor table called the Interrupt Descriptor Table (IDT) is located at a user-specified address in memory. This table is accessible through an entry in the Global Descriptor Table (GDT). This makes an interrupt procedure entry point to be directly accessed via a jump to the specific SELECTOR:OFFSET pointer in the IDT. All interrupts will have a SELECTOR:OFFSET address just as in the iRMX 86 operating system. Therefore, the system calls syntax will remain the same, except the parameter called INTERRUPT$HANDLER as shown below:

Example: iRMX 286

```
CALL RQ$SET$INTERRUPT (LEVEL, INTERRUPT$FLAGS, INTERRUPT$HANDLER,
                       INT$HANDLER$DS, EXCEPT$PTR);
```

Where INTERRUPT$HANDLER, the entry point to the interrupt handler, should be coded directly, i.e., @MY$HANDLER.

By referencing a handler directly, all other intermediate steps are unnecessary. (See the example in the PL/M 286 section.)

BASIC INPUT/OUTPUT SYSTEM (BIOS)

The BIOS of the iRMX 86 operating system is nearly identical to the iRMX 86 operating system BIOS. The same system calls are available with no changes or additions. The significant differences between the two BIOS's are the 16MB addressability and memory protection available in the iRMX 286 operating system.

Protection

The memory protection offered by the iRMX 286 operating system BIOS protects the code and data by preventing any task from reading or writing a segment of memory unless explicit access has been granted. It also prevents memory reads or writes from crossing segment boundaries. Therefore any task using the A$READ or A$WRITE BIOS system calls must have read or write access privileges.

Device Drivers

Not all iRMX 86 operating system device drivers have been included in the iRMX 286 operating system. Consult the following list or the iRMX 286 Interactive Configuration Utility for the specific Intel-supplied drivers.

Intel Device Drivers Supplied With iRMX® 286 R. 2.0

- iSBC® 215G
- iSBC 214
- iSBX™ 218A
- iSBX 217C
- iSBC 220
- iSBC 208
- iSBX 251
- iSBC 264
- iSBX 350 Line Printer
- Line Printer for 286/10
- iSBC 534
- iSBC 544
- Terminal Comm Cntlr
  - to include:
    - iSBC 188/48
    - iSBC 188/56
    - iSBC 546
    - iSBC 547
- iSBC 548
- 8274
- 8251A
- 82530
- RAM disk
- iSBC 286/10
- iSBC 286/10A
- iSBC 286/1X
- iSBC 386/2X

Not included are the following device drivers:

- iSBC 204
- iSBC 206
- SCSI
- iSBC 226

EXTENDED INPUT/OUTPUT SYSTEM (EIOS)

The EIOS of the iRMX 286 operating system is nearly identical to the iRMX 86 operating system EIOS. The same system calls are available with few changes and additions. The significant differences between the two EIOS's are the 16MB addressability and memory protection available in the iRMX 286 operating system.

Protection

The memory protection offered by the iRMX 286 operating system EIOS protects the code and data by preventing any task from reading or writing a segment of memory unless explicit access has been granted. It also prevents memory reads and writes from crossing segment boundaries. The system calls S$READ$MOVE and S$WRITE$MOVE are two calls that will send an exception code called EBADBUFF whenever this occurs.

Extended Memory Pool

Since the iRMX 286 operating system supports the 16MB addressability of the 80286 processor, the memory pools created by I/O jobs can also be as large as 16MB. The new system call providing this feature is called RQE$CREATE$IO$JOB.

New Calls

Several new system calls have been added to the iRMX 286 operating system EIOS layer. They are:

1. RQE$CREATE$IO$JOB
   POOLMIN and POOLMAX parameters changed to DWORDS for 16MB addressability.
2. RQSGETDIRECTORY$ENTRY
   Retrieve name of any file in a directory.
3. RQSGETPATH$COMPONENT
   Retrieve name of any file as it is known in its parent directory.

iRMX® 286 APPLICATION LOADER

80286 OMF

Two utilities are supplied with the iRMX 286 operating system to load programs and data into system memory from secondary storage devices. They are the bootstrap loader and the application loader. Typically the bootstrap loader is used to load the initial system and begin its execution. The application loader will typically be called, by programs running in the system, to load additional programs. The application loader can load I/O jobs up to 16MB. These programs must be in the 80286 Object Module Format (OMF). This differs from the iRMX 86 operating system, which loads only 8086 OMF records. Further, the 80286 records must be in STL format. (See a later section called BND 286 for a discussion of STL format.)

New Calls

RQEALOADIOJOB

This calls memory pools changed to DWORD values from word. (See RQE$CREATE$JOB call in the Nucleus section.)

RQESLOADIOJOB

Same as above.

HUMAN INTERFACE

Enhanced Command Line Interpreter (CLI)

The new CLI provides line-editing features, as well as its own set of commands. With CLI commands, aliases can be created, background programs ran, output redirected or redefined for a terminal in the configuration file. The commands are:

!	ALIAS	BACKGROUND	CHANGEID	DEALIAS	EXIT	
HISTORY	JOBS	KILL	LOGOFF	SET	SUBMIT	SUPER

To include or customize features in the CLI, user extensions have been added to the Human Interface.

New Calls

ADDLOC	LOGOFF	SHUTDOWN	LOCK	UNLOCK	ZSCAN

Old Calls

The following Human Interface commands have been revised:

BACKUP	DISKVERIFY	FORMAT	LOCDATA	RESTORE

UDI

New System Calls

The iRMX 286 UDI contains three system calls not contained in the iRMX 86 UDI. They are:

DQ$MALLOCATE

DQ$MFREE

DQGETMSIZE

All of the calls have their counterparts in the iRMX 86 UDI, however, the new system calls use full pointers instead of selectors and DWORD instead of WORD for memory block start address and size specifications, respectively.

These three calls are only supported in programs compiled in the compact or large segmentation models. Also, earlier versions of these calls cannot be mixed. For example:

After using DQ$MALLOCATE to allocate memory, do not use DQ$FREE to free it.

Use DQ$MFREE instead.

BOOTSTRAP LOADER

Two Stage Loader

To facilitate loading an application so that it may execute has been known as "pulling it up by its bootstraps" or simply "booting" the application. iRMX bootstrap loaders have been divided into stages, each possessing a unique purpose and role.

In the iRMX 86 operating system, the bootstrap loader exists as only two stages. The first stage resides in PROM located on the CPU's board. If supplied by Intel, it will occupy less than 8Kb of memory within the PROM. Once running, it will identify the applications name and location, then load part of the second stage, passing control to it. The second stage finishes loading itself, loads the application into memory, then passes control to the application. While the first stage is user-configurable, the second stage is not. The second stage is only supplied by Intel and is present on all iRMX formatted, named volumes.

New Third Stage

In the iRMX 286 operating system, the bootstrap loader exists as three stages. The extra stage was added to be able to load 80286 OMF files. This will also permit loading 8086 OMF files with just the first and second stages. This means either system can be booted without compromising the other. To allow for this, some files have to be renamed and some new conventions adopted. (See below and Figure 3.)

1. All 80286 OMF bootloader application systems must have the extension ".286".
2. The third stage bootstrap loader must have the same name as the application, less the extension.
3. The third stage bootstrap loader must reside in the same directory as the bootloadable system.

File Name Conventions

Third Stage	System to be Loaded
/SYSTEM/RMX86	/SYSTEM/RMX86.286
/SYSTEM/RMX	/SYSTEM/RMX.286
/BOOT/RMX286	/BOOT/RMX286.286

This chart indicates to those wanting to boot the iRMX 86 operating system that their file /SYSTEM/RMX86 had better be renamed to avoid confusion.

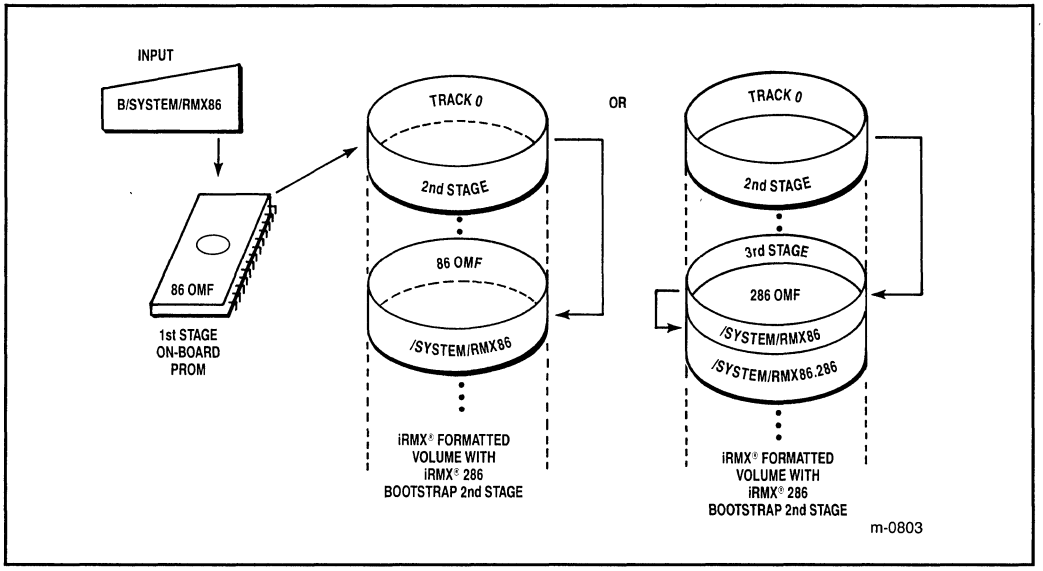


Figure 3.

When installing the iRMX 286 operating system on a system containing the iRMX 86 operating system, the "BS" option of the format command will install ONLY the new second stage bootstrap loader on track 0 of the hard disk. The installation process will also add new directories as required by the iRMX 286 operating system.

Memory Locations of the Three Stages

Bootstrap Loader Locations

Description	Default	Approx. Size	Config. File
1st STAGE CODE	Application dependent	12KB	BS1.CSD
2nd STAGE CODE 1st/2nd DATA and STACK	0B8000H	8KB	BS1.CSD
3rd STAGE (specific) CODE DATA and STACK	0BC000H	16KB	BS3.CSD
3rd STAGE (generic) CODE	0BC000H	8KB	BG3.CSD
3rd STAGE (generic) DATA and STACK	0B8000H	—	BG3.csd

CONFIGURATION SIZE CHART

Operating System Layer	iRMX® 286 Memory Requirements	iRMX® 86* Code Size	iRMX® 286* Code Size	iRMX® 86* Data Size	iRMX® 286* Data Size
Nucleus	34K	24K	27K	2K	3.5K
BIOS	95K	78K	67K	1K	19.5K
EIOS	19K	12.5K	16K	1K	16.75K
Application Loader	12K	10K	11K	2K	2K
HI	36K	22K	26K	15K	1K
UDI	11K	8K	9.4K	0K	0.1K
Bootstrap Loader	—	1.5K	32K	6K	6K
ICU	—	—	—	308K	384K

*These numbers reflect actual memory size required to support the different configurations of the operating systems.

FILE STRUCTURE

The file system of the iRMX 286 operating system provides for the same types of files as are on the iRMX 86 operating system. In fact, both file systems can exist on the same volume using the same hierarchical file structure. This is made possible through the installation of the iRMX 286 bootstrap loader's second stage onto the iRMX 86 operating system's volume. This second stage will allow either operating system to be booted from the same volume. One fact should be remembered: iRMX 286 uses the 80286 OMF, while iRMX 86 uses the 8086 OMF. This stops either operating system from loading and executing the other's files or programs. Copying, deleting or other maintenance operations can still be accomplished across the volume. iRMX 286 operating system will also read iRMX 86 back-up format files from another volume. The following Figure 4 shows a file system with both operating systems installed, including the changes to its structure. Remember, iRMX 286 can reside by itself or with iRMX 86 on the same volume.

Conventions

New file conventions have been adopted to differentiate between several types of files. They are:

*.P28 — PL/M 286 source files
*.P86 — PL/M 86 source files
*.A28 — ASM 286 source files
*.A86 — ASM 86 source files
*.GAT — Gate definition files

*.BLD — Build, file for BLD 286

*.286 — Bootable iRMX 286 system file

*.86 — Bootable iRMX 86 system file

After booting iRMX 286, the following assignments are assumed:

: SYSTEM : = /SYS286
: UTIL : = /UTIL286
: LANG : = /LANG286

After booting iRMX 86, the following still apply:

: SYSTEM : = /SYSTEM
: UTIL : = /UTILS
: LANG : = /LANG

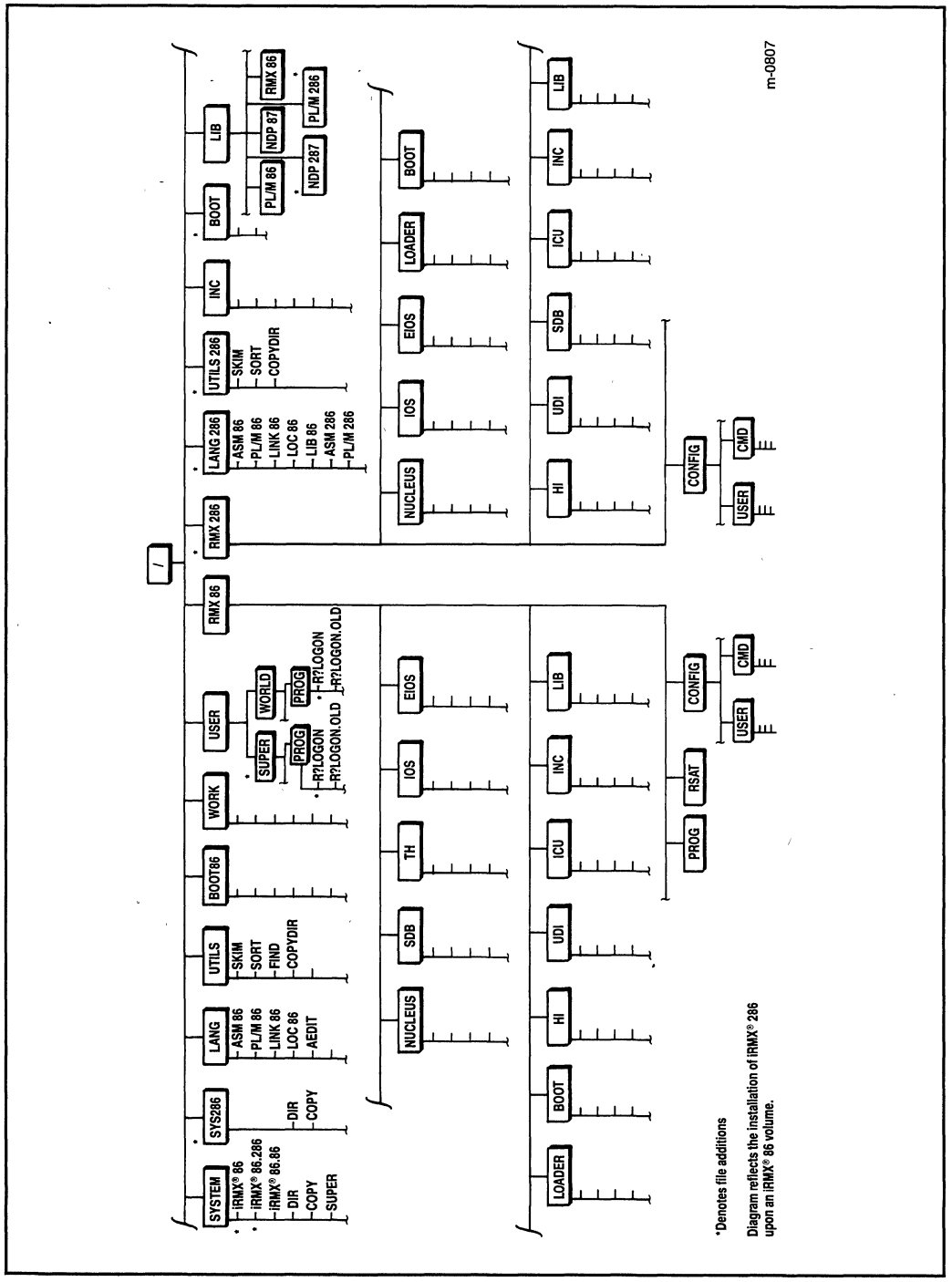

/
SYSTEM
iRMX® 86
iRMX® 86.286
iRMX® 86.86
DIR
COPY
SUPER
SYS286
DIR
COPY
LANG
ASM 86
PL/M 86
LINK 86
LOC 86
AEDIT
UTILS
SKIM
SORT
FIND
COPYDIR
BOOT86
WORK
SUPER
PROG
R?LOGON
R?LOGON.OLD
USER
WORLD
PROG
R?LOGON
R?LOGON.OLD
RMX 86
NUCLEUS
SDB
TH
IOS
EIOS
LOADER
BOOT
HI
UDI
ICU
INC
LIB
PROG
RSAT
CONFIG
USER
CMD
RMX 286
LANG 286
ASM 86
PL/M 86
LINK 86
LOC 86
LIB 86
ASM 286
PL/M 286
UTILS 286
SKIM
SORT
COPYDIR
INC
BOOT
LIB
PL/M 86
NDP 87
RMX 86
NDP 287
PL/M 286
NUCLEUS
IOS
EIOS
LOADER
BOOT
HI
UDI
SDB
ICU
INC
LIB
CONFIG
USER
CMD
m-0807
*Denotes file additions
Diagram reflects the installation of iRMX® 286
upon an iRMX® 86 volume.

Figure 4.

LANGUAGES: ASM 286

Because ASM 286 supports the 80286 in protected mode, ASM 286 has more changes than other languages. Often users converting their programs to ASM 286 from ASM 86 will assemble the programs in ASM 286 and store the error messages generated and change the code accordingly. A few notable changes are listed below.

Group Directive

ASM 286 does not possess a group directive as in ASM 86. By giving the segments the same name, they will be grouped together into one segment at link time.

Example: ASM 86

```
DATAGRP GROUP DATA1, DATA2
DATA1 SEGMENT
ABYTE DB 0
DATA1 ENDS

DATA2 SEGMENT
AWORD DW 0
DATA2 ENDS
ASSUME DS:DATAGRP

: ASM 286

DATA1 SEGMENT RW PUBLIC      }
ABYTE DB 0                   }   In one module
DATA1 ENDS                   }

DATA1 SEGMENT RW PUBLIC      }
AWORD DW 0                   }   In another module
DATA1 ENDS                   }
ASSUME DS:DATA1              }
```

Segment Directive

The fields of the SEGMENT directive are also different. ASM 286 does not use anything but para-aligned and access-type.

Example: ASM 86

```
NAME SEGMENT [ALIGN-TYPE] [COMBINE-TYPE]
  WHERE [ALIGN-TYPE] = PARA, BYTE, WORD, PAGE, INPAGE,
                       OR NONE

ASM 286

NAME SEGMENT [ACCESS-TYPE] [COMBINE-TYPE]
  WHERE SEGMENT IS ALWAYS PARA-ALIGNED AND
    [ACCESS-TYPE] = READ-ONLY (RO),
                    EXECUTE-ONLY (EO),
                    EXECUTE-READ (ER), or
                    READ-WRITE (RW)
```

Class name is also not present in ASM 286

Stack Segment

In ASM 286, stack segments are defined using the STACKSEG directive.

Example: ASM 286

```
PROG_STACK STACKSEG 10;                    /* MEANS 10 BYTES ON STACK */
```

The operator STACKSTART is used to define a label at the beginning of the stack to initialize the Stack Pointer (SP).

Example: ASM 286

```
MOV SP, STACKSTART PROG_STACK
```

Selector Access

In ASM 286 the selectors used for the DS, SS, and ES in the ASSUME directive must have certain access types.

Example: ASM 286

```
ASSUME DS:EDATA
EDATA SEGMENT RW PUBLIC
        WHERE DB 0
EDATA ENDS
```

Further, the ASSUME directive will not accept an assume for the code segment. The current code segment being assembled is automatically assumed into the CS. For more information regarding other changes in ASM 286 consult: *ASM 286 Reference Manual* (Appendix G), order #122671

LANGUAGES: PL/M 286

Users migrating their code to PL/M 286 should be aware of the following:

Pointer and Selector Variables

Pointer and selector variables cannot be assigned absolute values. All values must be assigned by reference to another variable or through based-variables.

Example: PL/M 86

```
Declare
   A$POINTER     POINTER;

Start: DO;
       A$POINTER = 0;
```

Example: PL/M 286

```
Declare
   A$POINTER     POINTER;

Start: DO;
       A$POINTER = NIL;
```

Similarly selectors can be assigned values as follows:

Example: PL/M 86

```
Declare token literally 'WORD',
        A$TOKEN     TOKEN;
Start: DO;
       A$TOKEN = 0;
```

Example: PL/M 286

```
Declare token literally 'SELECTOR',
        A$TOKEN     TOKEN;
Start: DO;
       A$TOKEN = SELECTOR$OF(NIL);
```

The only relational operations allowed in PL/M 286 for pointers and selectors are "equals" and "not equals".

Models of Compilation

In PL/M 86 the default is SMALL

In PL/M 286 the default is LARGE

Interrupt Vectors

In PL/M 286 all interrupt numbers on all interrupt procedures must be deleted. The required interrupt vectors will be assigned by the 80286 system builder if not already defined by the iRMX 286 operating system call RQSET INTERRUPT.

Consequently the PL/M 86 built-ins SET$INTERRUPT and INTERRUPT$PTR are unavailable in PL/M 286 and should be removed. Also, all calls to interrupt procedures are not allowed. As the conversion process takes shape, all of these changes turn out better than initially expected as the following example shows.

Example: PL/M 86

```
 1. DECLARE      ZERO      LITERALLY      '00001000b',
 2.              INTERRUPT_HANDLER POINTER;
                          .
                          . TYPICAL PL/M 86 STATEMENTS
                          .
 6. INTERRUPT_HANDLER : PROCEDURE INTERRUPT 56 PUBLIC REENTRANT;
                          .
                          . TYPICAL PL/M 86 STATEMENTS
                          .
10.                       CALL RQ$SIGNAL$INTERRUPT (ZERO, @STATUS);
11. END INTERRUPT_HANDLER;

12. INTERRUPT_TASK    : PROCEDURE PUBLIC REENTRANT;
                          .
                          . TYPICAL PL/M 86 STATEMENTS
                          .
16.          INTERRUPT_HANDLER = INTERRUPT$PTR (INTERRUPT_HANDLER);
17.                       CALL RQ$SET$INTERRUPT (ZERO, 1, INTERRUPT_HANDLER,
                                    DATA$SEG$ADDRESS.BASE, @STATUS);
                          .
                          . TYPICAL PL/M 86 STATEMENTS
                          .
21.                       CALL RQ$WAIT$INTERRUPT (ZERO, @STATUS);
22.     END INTERRUPT_TASK;
```

Comments

Line Number	Description
2.	INTERRUPT_HANDLER was defined as a pointer
6.	Interrupt entry 56 was "hard-coded"
16.	INTERRUPT_HANDLER was assigned the location (address) of the first instruction of the handler via the PL/M 86 built-in "INTERRUPT$PTR"
17.	This call could have looked like: RQSETINTERRUPT (ZERO, 1, INTERRUPT_PTR(INTERRUPT_HANDLER), etc eliminating lines 2 and 16.

Example: PL/M 286

```
1. DECLARE     ZERO     LITERALLY        '00001000b';
                        .
                        . TYPICAL PL/M 286 STATEMENTS
                        .
5. INTERRUPT_HANDLER  : PROCEDURE INTERRUPT PUBLIC REENTRANT;
                        .
                        . TYPICAL PL/M 286 STATEMENTS
                        .
9.                      CALL RQ$SIGNAL$INTERRUPT (ZERO, @STATUS);
10.    END INTERRUPTHANDLER;

11. INTERRUPT_TASK     : PROCEDURE PUBLIC REENTRANT;
                        .
                        . TYPICAL PL/M 286 STATEMENTS
                        .
15.                     CALL RQ$SET$INTERRUPT (ZERO, 1, @INTERRUPT_HANDLER,
                                                DATA$SEG$ADDRESS.BASE, @STATUS);
                        .
                        . TYPICAL PL/M 286 STATEMENTS
                        .
19.                     CALL RQ$WAIT$INTERRUPT (ZERO, @STATUS);
20.      END INTERRUPT_TASK;
```

Comments

Line Number	Description
5.	Notice PL/M 286 does not need to identify the interrupt in this statement
15.	The third parameter becomes simply a pointer to the first instruction of the handler.

DEVELOPMENT TOOLS — BND 286

All iRMX 86 programs linked using LINK 86 will instead have to be bound using BND 286. BND 286 is used to create all single-task application programs that will be dynamically loaded. (See Figure 5.) The following are tasks of the binder.

1. Creates a linkable or loadable module by combining input modules with other bindable modules.
2. Checks the type of variables and procedures.
3. Selects modules from libraries to resolve all symbolic references.
4. Combines logical segments by name, attribute, and privilege levels into physical segments that the processor can manipulate efficiently.
5. Can create a module the application loader can load.

Linkable Modules

In a process called incremental linking, BND 286 combines linkable object modules, including library modules, output by translators. The result is a file containing a linkable module.

Loadable Modules

A dynamically loadable module created by BND 286 is an executable module created by the combination of one or more linkable modules. Loadable modules can be of two types:

1. Single-task loadable (STL)
2. Variable-task loadable (VTL)

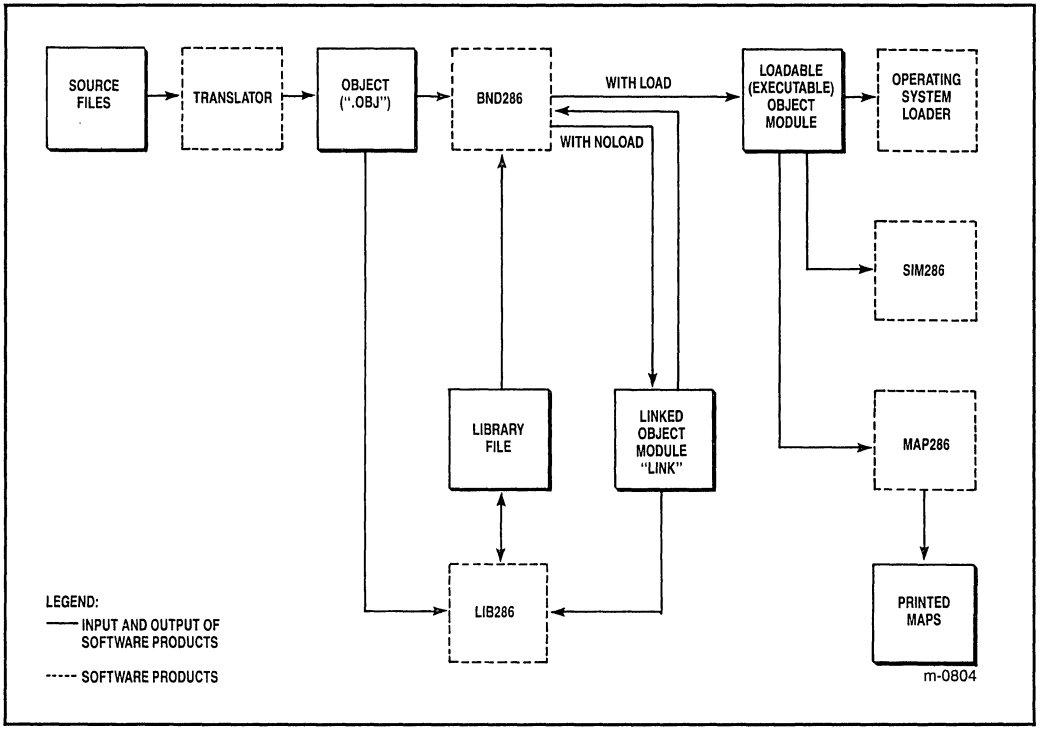


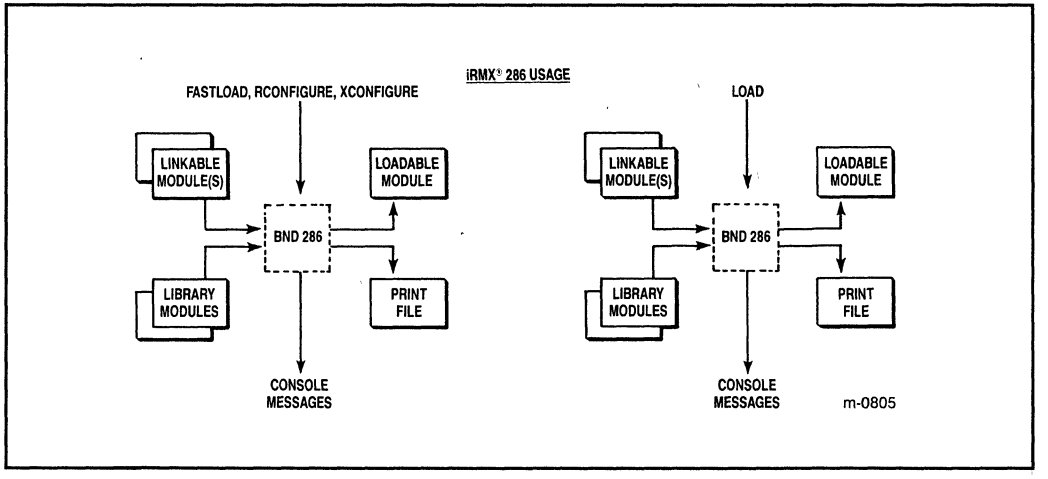
Figure 5. BND 286 Application Program Development

STL Modules

These modules are functionally similar to LTL-format records in the 8086 OMF. STL modules are designed to optimize loader execution time because each contains only one executable task. iRMX 286 and XENIX 286 operating systems will execute only files containing STL modules. BND 286 outputs STL modules when the FASTLOAD, RCONFIGURE, and XCONFIGURE controls are specified. In iRMX 286 only, the RCONFIGURE control is used.

VTL Modules

VTL modules are designed, when provided by BND 286, to also contain a single executable task, but in a format structured for multiple tasks. BND 286 outputs VTL modules when the LOAD control is specified.



BND 286 TO LINK 86 COMPARISON

BND 286 CONTROLS	LINK 86 CONTROLS
CONTROLFILE (pathname)	—
DEBUG/NODEBUG	SYMBOLS/NOSYMBOLS
ERRORPRINT (pathname)/ NOERRORPRINT	—
FASTLOAD/NOFASTLOAD	FASTLOAD/NOFASTLOAD
*LDTSIZE ([+] number)	—
LOAD/NOLOAD	—
NAME (modulename)	NAME (modulename)
OBJECT (pathname)/ NOOBJECT	—
PACK/NOPACK	—
PRINT (pathname)/ NOPRINT	PRINT (pathname)/ NOPRINT
PUBLICS/NOPUBLICS	PUBLICS/NOPUBLICS/PUBLICSONLY
RCONFIGURE (dm,m)	BIND and MEMPOOL
RENAMESEG (old to new)	—
RESERVE (number)	—
SEGSIZE (name(size))	SEGSIZE (name(size))
*TASKPRIVILEGE ( )	—
TYPE/NOTYPE	TYPE/NOTYPE
*XCONFIGURE	—

*Not used in iRMX 286

The following is an example of BND 286 for a simple human interface Commonly Used System Program (CUSP) used on an iRMX 286 Release 1.0 system.

```
BND 286                 EXAMPLE.OBJ,                    &
                        EXAMPLE.LIB,                    &
iRMX 286 Libraries——    /RMX286/LIB/UPIFC.LIB,          &
                        /RMX286/LIB/UDI.GAT,            &
                        /RMX286/LIB/HPIFC.LIB,          &
                        /RMX286/LIB/HI.GAT,             &
                        /RMX286/LIB/LPIFC.LIB,          &
                        /RMX286/LIB/LOA.GAT,            &
                        /RMX286/LIB/EPIFC.LIB,          &
                        /RMX286/LIB/EIO.GAT,            &
                        /RMX286/LIB/IPIFC.LIB,          &
                        /RMX286/LIB/IOS.GAT,            &
                        /RMX286/LIB/NUCIFC.LIB,         &
                        /RMX286/LIB/NUC.GAT             &
                        RCONFIGURE (DM(10000H, 10000H))
(Analogous to BIND&MEMPOOL)
                        SEGSIZE (STACK(1024))
(Analogous to segsize)
                        OBJECT (EXAMPLE)
(A new control)
```

iRMX 286 Library Privilege Gates

The following is an example of BND 286 for a simple human interface Commonly Used System Program (CUSP) on an iRMX 286 Release 2.0 system. Notice all of the .GAT files and many of the .LIB files are gone. All of these "missing" files are now contained in the files RMXIFC.LIB and UDIIFC.LIB for convenience.

```
BND 286                 EXAMPLE.OBJ,                    &
                        EXAMPLE.LIB,                    &
                        /RMX286/LIB/UDIIFC.LIB,         &
                        /RMX286/LIB/RMXIFC.LIB,         &
                        RCONFIGURE (DM(10000H,10000H))
(Analogous to BIND & MEMPOOL)
                        SEGSIZE (STACK(1024))
(Analogous to SEGSIZE)
                        OBJECT (EXAMPLE)
(A new control)
```

iRMX(R) XXX.BLD File

```
system_bld;
segment
        nucdat.code(dpl = 0),
        nucdat.data(dpl = 0),
memory
        (reserve = (0..0001FFFH,
         003A000H..0FFFFFFh));
        .
        .
gate
        Gate_CreateJob (entry =
            RqCreateJob, dpl = 0, wc = 0),
        .
        .
table
        ldt1 (limit = 00600h,dpl = 0,
        reserve = (2..2, 4..4AH,
        4CH..4EH, 51H..59h,
        122H..005FFh),
        entry = ( 0:nucdat.escape_ss,
                    3:nucdat.stack,
                   75:nucdat.jobdat,
                   79:nucdat.escape_ss,
                   80:nucdat.entry_code) );
task
        rmxtask (dpl = 0,object = nucdat,
                   ldt = ldt1, no ie);

table
        gdt (limit = 00600H, dpl = 0,
        reserve = (3..3BH, 3DH..4EH,
        51H..53H, 55H..59H, 0C1H..0C7H,
        0E3H..0E5H,0EAH..0EFH,
        101H..103H, 00137h..00140h),
        entry = (60:nucdat.data,
                   79:rmxtask,
                   80:nucdat.code,
                   84:ldt1,
                   90:Gate_AcceptControl,
                   91:Gate_AlterComposite,
                .
                .
                  308:sdbcnf.code,
                  309:sdbcnf.data,
                  310:sdbcnf.newstack,
                  291:bios_code,
                  292:bios_data,
                .
                .
table
        idt(limit = 00080h, dpl = 0);
end
```

DEVELOPMENT TOOLS — BLD 286

BLD 286 exceeds LOC 86 in capability and versatility. In many cases the use of BLD 286 is transparent to iRMX 286 users, due to the ICU 286 automatically generating the BUILD file. All iRMX 286 users, however, should possess an understanding of the following key functions:

A. Assigns physical addresses to entities, sets segment limits and access rights. (See XXX.BLD file)

B. Allows memory ranges to be reserved or allocated for specific entities. (See XXX.BLD file)

C. Creates one Global Descriptor Table (GDT), one Interrupt Descriptor Table (IDT), and one Local Descriptor Table (LDT). (See XXX.BLD file)

D. Creates gates. (See XXX.BLD file)

E. Creates task state segments and (task gates). (See XXX.BLD file)

F. Creates a bootloadable module. (See XXX.BLD file)

G. Creates object files containing exported system entries. (See XXX.BLD file)

H. Selects required modules from specified libraries automatically, as needed to resolve symbolic references.

I. Performs reference-resolution and typechecking.

J. Detects and reports errors and warnings found during processing (in the XXX.MP2 file)

See Figure 6 for an example of BLD 286 program development.

Usage

BLD 286 is primarily used for building an application program that deals extensively with system interfaces to a hardware environment. This could include configuring gates and/or segments that provide this interface, then place these interfaces in a separate file for later exportation.

The types of executable output produced by BLD 286 are bootloadable, loadable, or incremental-built. Bootloadable modules are absolutely-located object modules that are booted via a simple loader. Loadable modules consist of single- or multiple-task modules used for dynamic loading. Incrementally-built modules are non-executable modules used interactively to build large systems.

Many users will only use BLD 286 when they produce a new configuration using the ICU. ICU 286 generates a file called ICUBLD.CSD which invokes the builder using the file XXX.BLD as the builder definition file.

The following is a typical example of the contents of ICUBLD.CSD:

```
BLD 286,                                &
          NUCLUS.LNK,                   &
          SDB.LNK,                      &
          IOS.LNK,                      &
          EIOS.LNK,                     &
          LOADER.LNK,                   &
(Produced by BND 286)
          HI.LNK,                       &
          UDI.LNK                       &
          OBJECT (/BOOT/***.286)        &   ( Where to put the
                                            bootloadable file)
          NODEBUG NOTYPE                &
          BUILDFILE (***.BLD)               ( Where to obtain the
                                            build information)
```

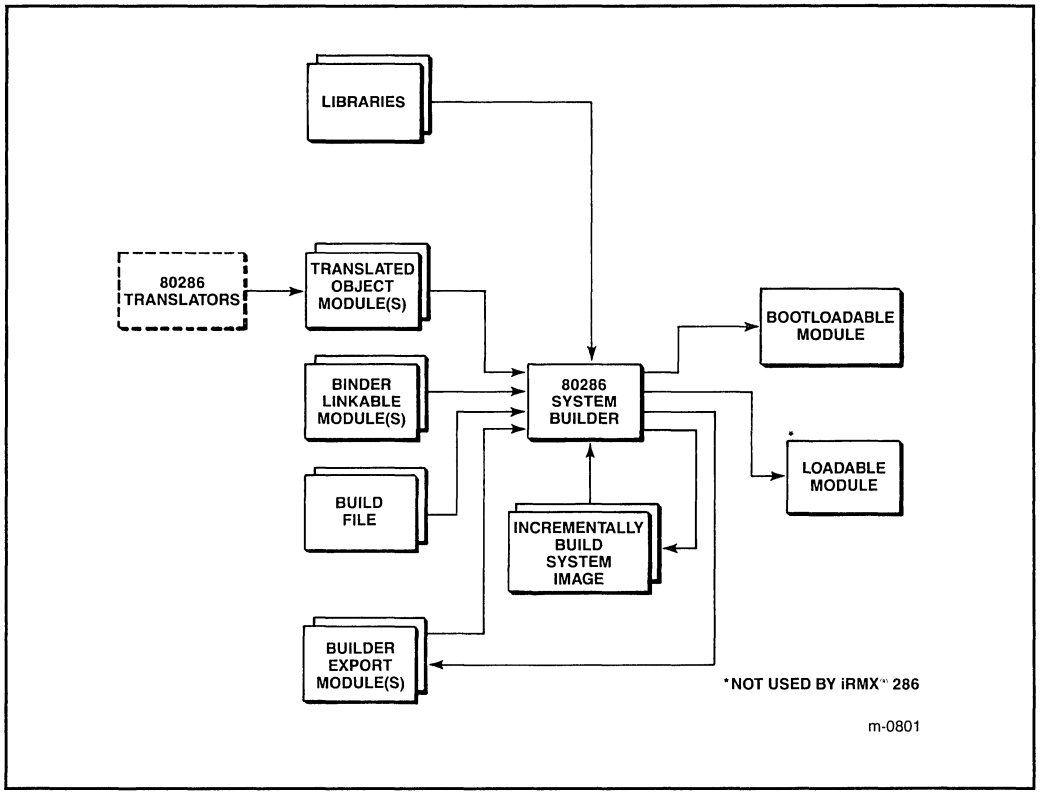


Figure 6. BLD 286 Application Program Development

The build file contains a specific language used by BLD 286 to produce the system or system program. BLD 286 takes all linked input modules and assigns all of the access and protection attributes for each subsystem. A build file is created to specify the characteristics of the relationships among the subsystems. Segment attributes, gates, descriptor tables, aliases, and memory allocation are also described in the build file and read by BLD 286.

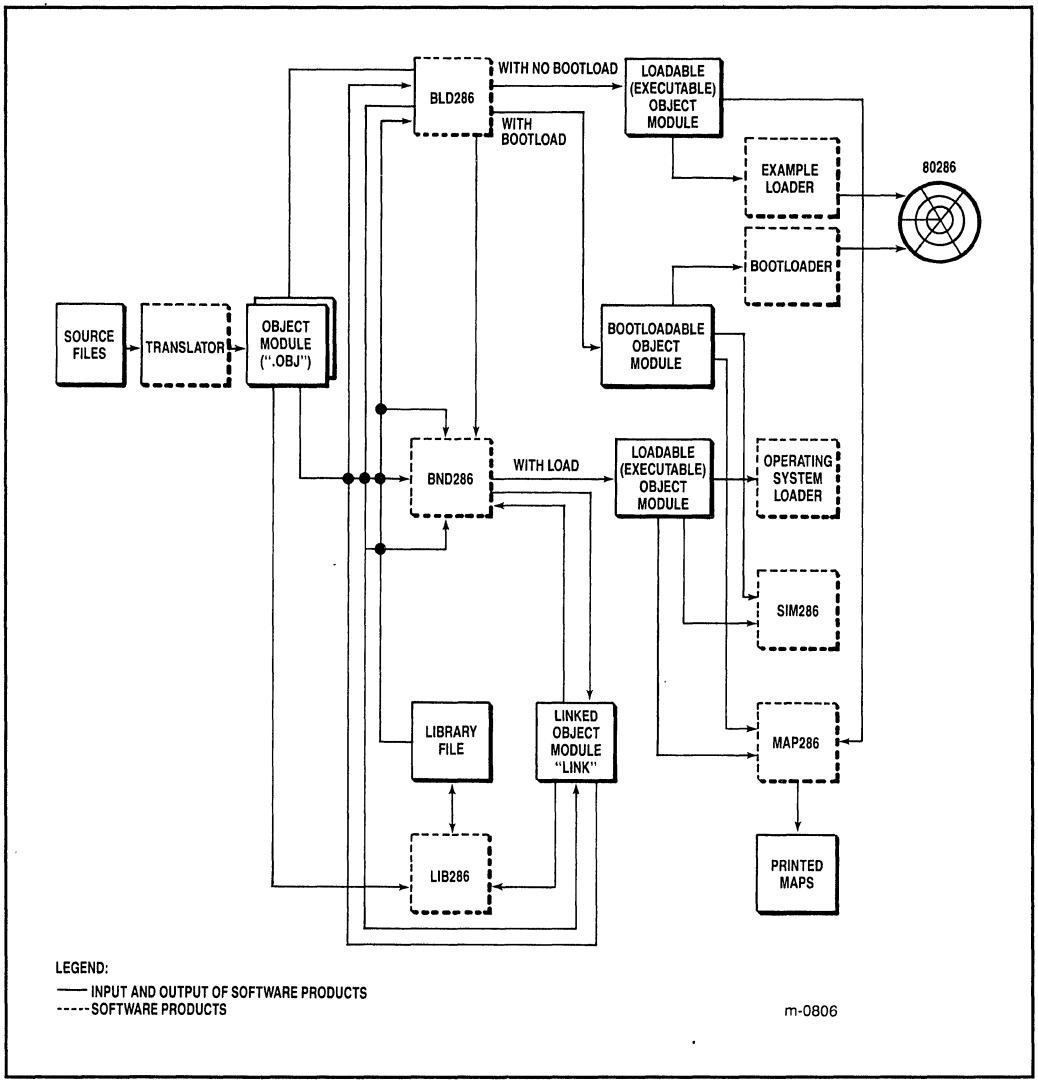

BLD286
WITH NO BOOTLOAD
WITH BOOTLOAD
LOADABLE (EXECUTABLE) OBJECT MODULE
EXAMPLE LOADER
80286
BOOTLOADER
SOURCE FILES
TRANSLATOR
OBJECT MODULE (".OBJ")
BOOTLOADABLE OBJECT MODULE
BND286
WITH LOAD
LOADABLE (EXECUTABLE) OBJECT MODULE
OPERATING SYSTEM LOADER
SIM286
LIBRARY FILE
LINKED OBJECT MODULE "LINK"
MAP286
LIB286
PRINTED MAPS
LEGEND:
—— INPUT AND OUTPUT OF SOFTWARE PRODUCTS
-----SOFTWARE PRODUCTS
m-0806

Figure 7.

MAP 286

The 80286 mapper is a noninteractive utility that generates object module information that BND 286 and BLD 286 do not produce. The utility is offered separately instead of having the builder and binder performing identical functions. The user should note that if debug information is contained in the invocation file, all of the maps will be produced.

MAP 286 will accept the following input:

A. Executable files containing a single executable module, and only one per invocation of MAP 286.

B. Executable files containing a single bootloadable module.

C. One or more linkable or library files.

MAP 286 produces the following output maps:

For executable input files:

A. An output object file with or without debug information.

B. Table MAP, segment MAP, gate MAP, public MAP, symbol MAP, task MAP, and crossreference MAP.

For linkable input files:

A. Only a cross reference map including a module list.

In iRMX 286 the following is a typical invocation of the mapper on an executable file called

```
MAP    286    MYPROG    <CR>
```

If debug information is in "MYPROG" all of the maps will be produced.

iRMX® 86 OPERATING SYSTEM PROGRAM MIGRATION

Compiling in PL/M 286

The following is an example of converting an iRMX 86 Commonly Used System Program (CUSP) called NOTE. To assist readers, all of the conversion steps will be described.

Source Program

The program NOTE is written in PL/M 86 for use on iRMX 86 operating system. When invoked, the utility will echo a line of keyboard input to the console.

The source code file name for NOTE is NOTE.P86. To adhere to PL/M 286 and iRMX 286 operating system file naming conventions, the file should be renamed to NOTE.P28. Next, the file has to be changed to reflect changes in PL/M 286 and iRMX 286 library files. Finally the file is compiled and bound with BND 286. See the following examples for further explanation.

STEP 1

Copy NOTE.P86 to NOTE.P28 <CR>

STEP 2

The NOTE.P28 file has to be edited to change

A. All '0' pointers to 'NIL'

B. All '0' selectors to 'SELECTORS$OF(NIL)'

Also notice all of the include files assume an iRMX 86 operating system and have to be changed to iRMX 286 libraries.

STEP 3

The new NOTE.P28 program is compiled and any errors are corrected.

```
$title('iRMX 86 HI NOTE command')
$subtitle('module header')
/*****************************************************************************
 TITLE: note
 ABSTRACT:
  This module contains the main routine for the HI note command.

    NOTE  message

  Message will be printed on EO.
*****************************************************************************/
hnote: DO;

$include(:sd:inc/hstand.lit)
$include(:sd:rmx86/inc/hgtchr.ext)
$include(:sd:rmx86/inc/hsneor.ext)
$include(:sd:inc/hutil.ext)

DECLARE
 version(*) BYTE  DATA( 'program_version_number=F001',
       'program_name=Note',0);

1  main: DO;
    /* local variables */
2 DECLARE
3  excep   WORD,
4  char   BYTE,
5  count   WORD,

6  msg  STRUCTURE(
7       length    BYTE,
8       char(STRING$MAX) BYTE);

9     count = 0;
10    char = rq$C$get$char( @excep);
11    DO WHILE( (char := rq$C$get$char( @excep)) <> 0);
12   IF  count < LAST(msg.char)  THEN
13    DO;
14     msg.char(count) = char;
15     count = count + 1;
16    END;
17     END;
18     msg.char(count) = cr;
19     count = count + 1;
20     msg.char(count) = lf;
21     count = count + 1;
22     msg.length = count;
23    CALL rq$C$send$EO$response( 0, 0, @msg, @excep);

    /* exit from command */

24    CALL cusp$error( excep, @(0), @(0), ABORT);
25   END main;

END hnote;
```

THIS POINTER NEEDS CHANGING.

PLM 86 Example

```
$title('iRMX 286 HI NOTE command')
$subtitle('module header')
/*************************************************************************
 TITLE: note
 ABSTRACT:
  This module contains the main routine for the HI note command.

    NOTE  message

  Message will be printed on EO.
*************************************************************************/
hnote: DO;

$include(:sd:inc/hstand.lit)
$include(:sd:rmx86/inc/hgtchr.ext)
$include(:sd:rmx86/inc/hsneor.ext)
$include(:sd:inc/hutil.ext)

DECLARE
 version(*) BYTE  DATA( 'program_version_number=F001',
       'program_name=Note',0);

1  main: DO;
    /* local variables */
2 DECLARE
3  excep   WORD,
4  char   BYTE,
5  count   WORD,

6  msg  STRUCTURE(
7      length    BYTE,
8      char(STRING$MAX) BYTE);

9   count = 0;
10   char = rq$C$get$char( @excep);
11   DO WHILE( (char := rq$C$get$char( @excep)) <> 0);
12  IF  count < LAST(msg.char)  THEN
13   DO;
14    msg.char(count) = char;
15    count = count + 1;
16   END;
17   END;
18    msg.char(count) = cr;
19    count = count + 1;
20    msg.char(count) = lf;
21    count = count + 1;
22   msg.length = count;
23   CALL rq$C$send$EO$response( NIL, 0, @msg, @excep);

    /* exit from command */

24    CALL cusp$error( excep, @(0), @(0), ABORT);
25  END main;

END hnote;
```

THIS IS OK NOW.

PLM 286 Version Example

Binding an iRMX® 286 Application

STEP 1

If a program was previously linked in iRMX 86, we then examine the original LINK file used and notice the following:

```
PLM86 %0.P86 COMPACT ROM OPTIMIZE(3) NOTYPE PW(132)
;
LINK86      %0.obj,                          &
   /rmx86/hi/hutil.lib,          &
   /lib/plm86/plm86.lib,         &
   /rmx86/lib/hpifc.lib,         &
   /rmx86/lib/epifc.lib,  &
   /rmx86/lib/ipifc.lib,  &
   /rmx86/lib/rpifc.lib  &
   to %.86
   bind  mempool(10000,0B0000H)
   nosb noty
```

1. The library names will change
2. The pathnames to access the libraries will change
3. BIND and MEMPOOL will change

STEP 2

The following is the iRMX 286 Release 1.0 version of the file in Step 4. Remember the libraries have changed names between iRMX 286 Release 1.0 and 2.0.

```
PLM286 %0.p28 COMPACT ROM OPTIMIZE(3) NOTYPE PW( 132)
;
bnd286   %0.obj,                                          &
         /rmx286/lib/hutil.lib,                           &
         /rmx286/lib/plm286.lib,                          &
         /rmx286/lib/hpifc.lib, /rmx286/lib/hi.gat,       &
         /rmx286/lib/epifc.lib, /rmx286/lib/eio.gat,      &
         /rmx286/lib/ipifc.lib, /rmx286/lib/ios.gat,      &
         /rmx286/lib/nucifc.lib, /rmx286/lib/nuc.gat      &
         renameseg(hi_code to code, hi_data to data) segsize (stack(1000H)) &
         object(%0) rc(dm(12000,1000000))
         nodebug noty
```

STEP 3

This is an example of the Step 4 file modified to run on iRMX 286 Release 2.0. Notice the reduction of library statements.

```
PLM286 %0.p28 COMPACT ROM OPTIMIZE(3) NOTYPE PW( 132)
;
bnd286   %0.obj,                              &
         /RMX286/hi/hutil.lib,                &
         /RMX286/lib/plm286/plm286.lib,       &
         /RMX286/lib/rmxifc.lib               &
         renameseg(hi_code to code, hi_data to data) segsize (stack(1000H)) &
         object(%0)  rc(dm(12000,1000000))
         nodebug noty
```

Though these few migration examples reflect trivial modifications, larger and more complex applications might require a little more attention.

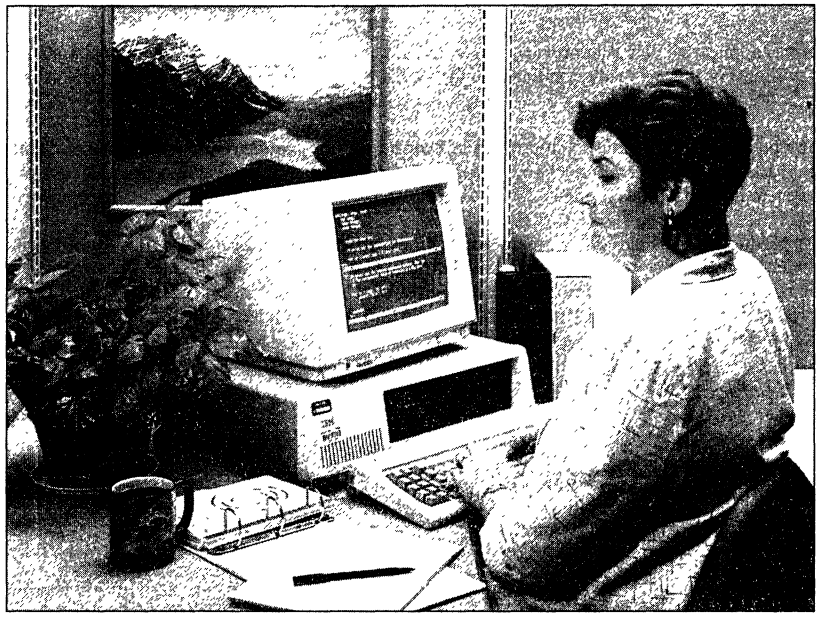
SUMMARY

The purpose of this application note is to provide insight and direction to those individuals contemplating using the iRMX 286 operating system. For those already familiar with the iRMX 86 operating system, this paper's focus is to provide the pathway to a superior product.

The iRMX 286 operating system is a vast improvement over its previous counterpart. Some notable changes are round robin scheduling, hardware-enforced protection, hardware-assisted debugging, and access to the 80386 processor. With this operating system the capabilities of the 80286 processor can be fully utilized for multiple environments.

Since the iRMX product line was introduced, many applications, programs, and lines of code have been written to support a tangible demand for real-time processing; in manufacturing, in medicine, and in finance, to name a few. As a result more time is being spent on designing, writing, and testing software than ever before. The iRMX 286 operating system is the preferred product for generating error-free programs while utilizing the highest CPU technology available in the OEM modules market.

AEDIT SOURCE CODE AND TEXT EDITOR



PROGRAMMER SUPPORT

AEDIT is a full-screen text editing system designed specifically for software engineers and technical writers. With the facilities for automatic program block indentation, HEX display and input, and full macro support, AEDIT is an essential tool for any programming environment. And with AEDIT, the output file is the pure ASCII text (or HEX code) you input—no special characters or proprietary formats.

Dual file editing means you can create source code and its supporting documents at the same time. Keep your program listing with its errors in the background for easy reference while correcting the source in the foreground. Using the split-screen windowing capability, it is easy to compare two files, or copy text from one to the other. The DOS system-escape command eliminates the need to leave the editor to compile a program, get a directory listing, or execute any other program executable at the DOS system level.

There are no limits placed on the size of the file or the length of the lines processed with AEDIT. It even has a batch mode for those times when you need to make automatic string substitutions or insertions in a number of separate text files.

AEDIT FEATURES

- Complete range of editing support—from document processing to HEX code entry and modification
- Supports system escape for quick execution of PC-DOS System level commands
- Full macro support for complex or repetitive editing tasks
- Hosted on PC-DOS and RMX operating systems
- Dual file support with optional split-screen windowing
- No limit to file size or line length
- Quick response with an easy to use menu driven interface
- Configurable and extensible for complete control of the editing process

September, 1989
Order Number 280804-002

FEATURES

POWERFUL TEXT EDITOR

As a text editor, AEDIT is versatile and complete. In addition to simple character insertion and cursor positioning commands, AEDIT supports a number of text block processing commands. Using these commands you can easily move, copy, or delete both small and large blocks of text. AEDIT also provides facilities for forward or reverse string searches, string replacement and query replace.

AEDIT removes the restriction of only inserting characters when adding or modifying text. When adding text with AEDIT you may choose to either insert characters at the current cursor location, or over-write the existing text as you type. This flexibility simplifies the creation and editing of tables and charts.

USER INTERFACE

The menu-driven interface AEDIT provides makes it unnecessary to memorize long lists of commands and their syntax. Instead, a complete list of the commands or options available at any point is always displayed at the bottom of the screen. This makes AEDIT both easy to learn and easy to use.

FULL FLEXIBILITY

In addition to the standard PC terminal support provided with AEDIT, you are able to configure AEDIT to work with almost any terminal. This along with user-definable macros and full adjustable tabs, margins, and case sensitivity combine to make AEDIT one of the most flexible editors available today.

MACRO SUPPORT

AEDIT will create macros by simply keeping track of the command and text that you type, "learning" the function the macro is to perform. The editor remembers your actions for later execution, or you may store them in a file to use in a later editing session.

Alternatively, you can design a macro using AEDIT's powerful macro language. Included with the editor is an extensive library of useful macros which you may use or modify to meet your individual editing needs.

TEXT PROCESSING

For your documentation needs, paragraph filling or justification simplifies the chore of document formatting. Automatic carriage return insertion means you can focus on the content of what you are typing instead of how close you are to the edge of the screen.

SERVICE, SUPPORT, AND TRAINING

Intel augments its development tools with a full array of seminars, classes, and workshops; on-site consulting services; field application engineering expertise; telephone hot-line support; and software and hardware maintenance contracts. This full line of services will ensure your design success.

SPECIFICATIONS

HOST SYSTEM

AEDIT for PC-DOS has been designed to run on the IBM* PC XT, IBM PC AT, and compatibles. It has been tested and evaluated for the PC-DOS 3.0 or greater operating system.

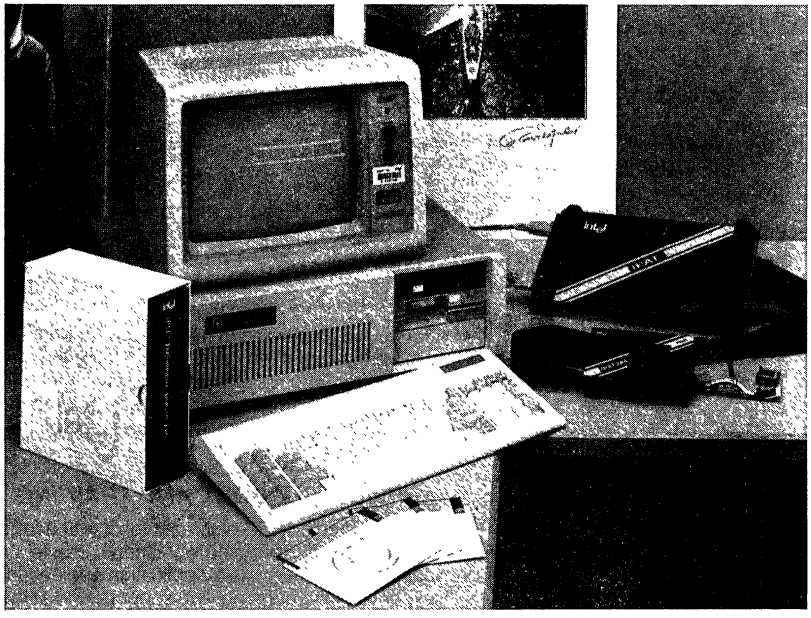
Versions of AEDIT are available for the iRMX™-86 and RMX II Operating System.

ORDERING INFORMATION

D86EDINL	AEDIT Source Code Editor Release 2.2 for PC-DOS with supporting documentation
122716	AEDIT-DOS Users Guide
122721	AEDIT-DOS Pocket Reference
RMX864WSU	AEDIT for iRMX-86 Operating System
R286EDI286EU	AEDIT for iRMX II Operating System

For direct information on Intel's Development Tools, or for the number of your nearest sales office or distributor, call 800-874-6835 (U.S.). For information or literature on additional Intel products, call 800-548-4725 (U.S. and Canada).

iPAT™ PERFORMANCE ANALYSIS TOOL



REAL-TIME SOFTWARE ANALYSIS FOR THE 8086/88, 80186/188, 80286, AND 80386

Intel's iPAT™ Performance Analysis Tool enables OEMs developing applications based on the 8086/88, 80186/188, 80286, or 80386 microprocessors to analyze real-time software execution in their prototype systems at speeds up to 20 MHz. Through such analysis, it is possible to speed-tune applications with real-time data, optimize use of operating systems (such as Intel's iRMX® II Real-Time Multitasking Executive for the 80286 and 80386, and iRMK™ Real-Time Multitasking Kernel for the 80386), characterize response characteristics, and determine code execution coverage by real-time test suites. Analysis is performed symbolically, non-intrusively, and in real-time with 100% sampling in the microprocessor prototype environment. iPAT supports analysis of OEM-developed software built using 8086, 80286, and 80386 assemblers and compilers supplied by Intel and other vendors.

All iPAT Performance Analysis Tool products are serially linked to DOS computer systems (such as IBM* PC AT, PC XT, and PS/2* Model 80) to host iPAT control and graphic display software. Several means of access to the user's prototype microprocessor system are supported. For the 80286 (real and protected mode), a 12.5 MHz iPAT-286 probe can be used with the iPATCORE system. For the 8086/88 (MAX MODE designs only), a 10 MHz iPAT-88 probe can be used with the iPATCORE system. iPATCORE systems also can be connected to sockets provided on the ICE™-286 and ICE-186 in-circuit emulators, or interfaced to IICE in-circuit emulators with probes supporting the 8086/88, 80186/188, or 80286. The 20 MHz iPAT™-386™ probe, also supported by the common iPATCORE system, can be operated either in "piggyback" fashion connected to an Intel ICE in-circuit emulator for the Intel386™, or directly connected to a prototype system independent of an ICE. iPAT-386 supports all models of 80386 applications anywhere in the lowest 16 Megabytes of the 80386 linear address space.

iPAT FEATURES

- Up to 20 MHz real-time analysis
- Histograms and analysis tables
- Performance profiles of up to 125 partitions
- Code execution coverage over up to 252K
- Hardware or software interrupt analysis
- Simple use with function keys and graphics
- Use with or without Intel ICEs

intel

September, 1989
Order Number 280786-002

FEATURES

MOST COMPLETE REAL-TIME ANALYSIS AVAILABLE TODAY

iPAT Performance Analysis Tools use in-circuit probes containing proprietary chip technology to achieve full sampling in real-time non-intrusively.

MEETS THE REAL-TIME DESIGNER'S NEEDS

The iPAT products include support for interactions between real-time software and hardware interrupts, real-time operating systems, "idle time," and full analysis of real-time process control systems.

SPEED-TUNING YOUR SOFTWARE

By examining iPAT histogram and tabular information about procedure usage (including or not including their interaction with other procedures, hardware, operating systems, or interrupt service routines) for critical functions, the software engineer can quickly pinpoint trouble spots. Armed with this information, bottlenecks can be eliminated by means such as changes to algorithms, recoding in assembler, or adjusting system interrupt priorities. Finally, iPAT can be used to prove the acceptibility of the developer's results.

EFFICIENCY AND EFFECTIVENESS IN TESTING

With iPAT code execution coverage information, product evaluation with test suites can be performed more effectively and in less time. The evaluation team can quickly pinpoint areas of code that are executed or not executed under real-time conditions. By this means, the evaluation team can substantially remove the "black box" aspect of testing and assure 100% hits on the software under test. Coverage information can be used to document testing at the module, procedure, and line level. iPAT utilities also support generation of instruction-level code coverage information.

ANALYSIS WITH OR WITHOUT SYMBOLICS

If your application is developed with "debug" symbolics generated by Intel 8086, 80286, or 80386 assemblers and compilers, iPAT can use them—automatically. Symbolic names also can be defined within the iPAT environment, or conversion tools supplied with the iPAT products can be used to create symbolic information from virtually any vendor's map files for 8086, 80286, and 80836 software tools.

REAL OR PROTECTED MODE

iPAT supports 80286 and 80386 protected mode symbolic information generated by Intel 80286 and 80386 software tools. It can work with absolute addresses, as well as base-offset or selector-offset references to partitions in the prototype system's execution address space.

FROM ROM-LOADED TO OPERATING SYSTEM LOADED APPLICATIONS

The software analysis provided by iPAT watches absolute execution addresses in-circuit in real time, but also supports use of various iPAT utilities to determine the load locations for load-time located software, such as applications running under iRMXII, DOS, Microsoft Windows*, or MS*OS/2.

USE STANDALONE OR WITH ICE

The iPAT-386, iPAT-286, and iPAT-86/88 probes, together with an iPATCORE system, provide standalone software analysis independent of an ICE (in-circuit emulator) system. The iPATCORE system and DOS-hosted software also can be used together with ICE-386, ICE-286, and I²ICE-86/88, 186/188, or 286 in-circuit emulators and DOS-hosted software. Under the latter scenario, the user can examine prototype software characteristics in real-time on one DOS host while another DOS host is used to supply input or test conditions to the protype through an ICE. It also is possible to use an iPATCORE and I²ICE system with integrated host software on a single Intel Series III or Series IV development system or on a DOS computer.

UTILITIES FOR YOUR NEEDS

Various utilities supplied with iPAT products support generation of symbolic information from map files associated with 3rd-party software tools, extended analysis of iPAT code execution coverage analysis data, and convenience in the working environment. For example, symbolics can be generated for maps produced by most software tools, instruction-level code execution information can be produced, and iRMXII-format disks can be read/written in DOS floppy drives to facilitate file transfer.

WORLDWIDE SERVICE AND SUPPORT

All iPAT Performance Analysis Tool products are supported by Intel's worldwide service and support. Total hardware and software support is available, including a hotline number when the need is there.

CONFIGURATION GUIDE

For all of the following application requirements, the iPAT system is supported with iPAT 2.0 (or greater) or iPAT/IICE 1.2 (or greater) host software, as footnoted.

Application Software	Option	iPAT Order Codes	Host System
80386 Embedded	#1	iPAT386DOS1, iPATCORE	DOS
iRMK on 80386	#1	iPAT386DOS, iPATCORE	DOS
iRMXII OS-Loaded or Embedded on 386	#1	iPAT386DOS, iPATCORE	DOS
OS/2-Loaded on 386	#1	iPAT386DOS, iPATCORE	DOS
iRMXII OS-Loaded or Embedded	#1	iPAT286DOS, iPATCORE	DOS
80286 Embedded	#1	iPAT286DOS, iPATCORE	DOS
	#2	ICEPATKIT2	DOS
	#3	IICEPATKIT3	DOS
	#4	IIIPATD, iPATCORE3	DOS4
	#5	IIIPATB, iPATCORE3	Series III4
	#6	IIIPATC, iPATCORE3	Series IV4
DOS OS-Loaded 80286	#1	iPAT286DOS, iPATCORE	DOS
OS/2 OS-Loaded 80286	#1	iPAT286DOS, iPATCORE	DOS
80186/188 Embedded	#1	ICEPATKIT2	DOS
	#2	IICEPATKIT3	DOS
	#3	IIIPATD, iPATCORE3	DOS4
	#4	IIIPATB, iPATCORE3	Series III4
	#5	IIIPATC, iPATCORE3	Series IV4
DOS OS-Loaded 8086/88	#1	iPAT88DOS, iPATCORE	DOS
8086/88 Embedded	#1	iPAT88DOS, iPATCORE	DOS
	#2	IICEPATKIT3	DOS
	#3	IIIPATD, iPATCORE3	DOS4
	#4	IIIPATB, iPATCORE3	Series III4
	#5	IIIPATC, iPATCORE3	Series IV4

Notes:

1. Operable standalone or with ICE-386 (separate product; separate host). iPAT-386 probe connects directly to prototype system socket, or to optional 4 probe-to-socket hinge cable (order code TA386A), or to ICE-386 probe socket.
2. Requires ICE-186 or ICE-286 in-circuit emulator system.
3. Requires IICE in-circuit emulator system.
4. Includes iPAT/IICE integrated software (iPAT/ICE 1.2 or greater), which only supports sequential iPAT and ICE operation on one host, rather than in parallel on two hosts (iPAT 2.0 or greater).

SPECIFICATIONS

HOST COMPUTER REQUIREMENTS

All iPAT Performance Analysis Tool products are hosted on IBM PC AT, PC XT, or PS/2 Model 80 personal computers, or 100% compatibles, and use a serial link for host-to-iPAT communications. At least a PC AT class system is recommended. The DOS host system must meet the following minimum requirements:

- 640K Bytes of Memory
- 360K Byte or 1.2M Byte floppy disk drive
- Fixed disk drive
- A serial port (COM1 or COM2) supporting 9600 baud data transfer
- DOS 3.0 or later
- IBM or 100% compatible BIOS

PHYSICAL DESCRIPTIONS

Unit	Width Inches	Width Cm.	Height Inches	Height Cm.	Length Inches	Length Cm.
iPATCORE	8.25	21.0	1.75	4.5	13.75	35.0
Power Supply	7 75	20.0	4.25	11.0	11.0	28.0
iPAT-386 probe	3.0	7.6	0.50	1.3	4.0	10.1
iPAT-286 probe	4.0	10.2	1.12	2.8	6.0	15.3
iPAT-86 probe	4 0	10 2	1 12	2.8	6.0	15 3
iPATCABLE (to ICE-186/286)	4.0	10.2	25	.6	36.0	91.4
IIIPATB,C,D (IICE board)	12.0	30.5	12.0	30.5	.5	1.3
Serial cables PC AT/XT PS/2					144.0	370.0

ELECTRICAL CONSIDERATIONS

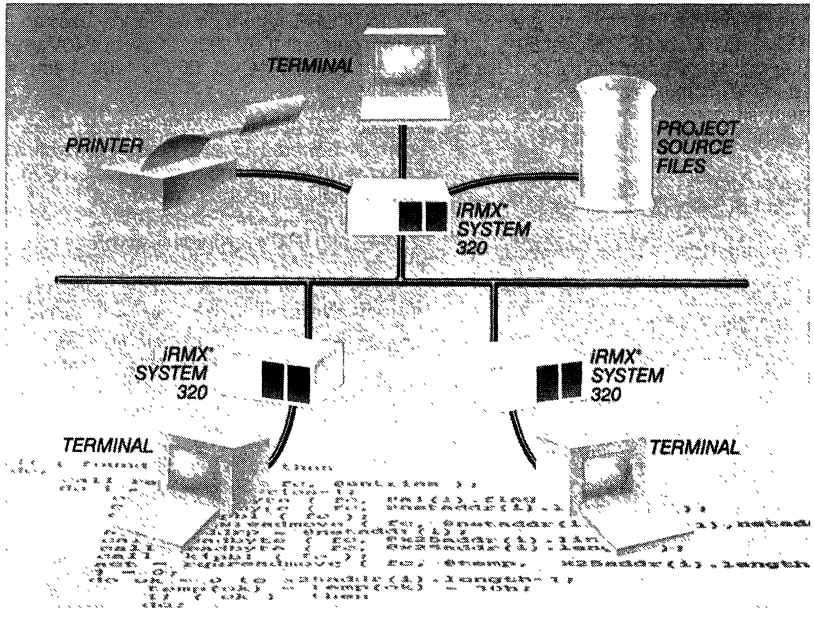
The iPATCORE system power supply uses an AC power source at 100V, 120V, 220V, or 240V over 47Hz to 63Hz. 2 amps (AC) at 100V or 120V; 1 amp at 220V or 240V.

iPAT-386, iPAT-286 and iPAT-86/88 probes are externally powered, impose no power demands on the user's prototype, and can thus be used to analyze software activity through power down and power up of a prototype system. For ICE-386, ICE-286, ICE-186, and IICE microprocessor probes, see the appropriate in-circuit emulator factsheets.

ENVIRONMENTAL SPECIFICATIONS

Operating Temperature: 10°C to 40°C (50°F to 104°F) ambient

Operating Humidity: Maximum of 85% relative humidity, non-condensing



iRMX® SOURCE CONTROL SYSTEM

The iRMX Source Control System (SCS) provides an integrated version control and generation management system for users in an iRMX software development cycle. This facility is useful for large and small software projects to assist in bringing more control, order and methodology to the software development process. SCS can be effectively used on a single iRMX System or across the OpenNET™ network.

FEATURES

- Controls access to source files
- Tracks changes to source files
- Approachable and efficient
- Generates any version of project
- Supports range of iRMX languages

September, 1989
Order Number 280739-001

CONTROLS ACCESS TO SOURCE FILES

With iRMX Source Control System the system manager (project leader) has certain privileged commands. These commands can be useful to designate those team members who can access the source files only for object generation and those who can access the source files for updating or changing. Other such privileged commands include the ability to archive a specific version of source and combine several versions of a source file.

TRACKS CHANGES TO SOURCE FILES

The iRMX Source Control System keeps track of changes made to any source files. These changes are stored as backward deltas for disk economy and fast access to the latest version. The project team can now better interact and synchronize using the latest updated version for integration and testing, especially as projects grow increasingly complex. The specific versions of tools used to produce the source code is also tracked.

APPROACHABLE AND EFFICIENT

The iRMX Source Control System has several facilities that help make it very approachable by the user. The tutorial leads the first time SCS user through the structure and capabilities of the iRMX Source Control System. The menu interface helps even the experienced SCS user learn and take advantage of the powerful capabilities of SCS. An on-line help facility assists in quick reminders for using the referenced commands.

The iRMX Source Control System makes efficient use of the system storage area and the development engineer's time.

The iRMX Source Control System can be used on a single iRMX System or can be utilized by a networked/distributed development team.

GENERATES ANY VERSION OF PROJECT

The iRMX Source Control System can be of particular use to both new active development projects as well as the evolving enhancement and maintenance of previous product releases. SCS provides for generation of any version of a project so that users can support (or test) different releases of a project from one source data base. Versions can be tagged for retrieval with symbolic names, state attributes or programmer name. Parallel development paths can be more easily and automatically merged using SCS.

SUPPORTS RANGE OF iRMX LANGUAGES

The iRMX Source Control System can be utilized by developers using any of the popular iRMX languages—PL/M, Assembler, FORTRAN, 'C,' PASCAL. The user can also configure support other special language requirements.

SPECIFICATIONS

PREREQUISITE HARDWARE

iRMX System 320 with at least 2MB, random access memory, 140MB winchester disk, and tape drive.

PREREQUISITE SOFTWARE

iRMX 286 R2.0 and AEDIT for single node system access. The above software prerequisite and iRMXNET R2.0 are required for networked utilization.

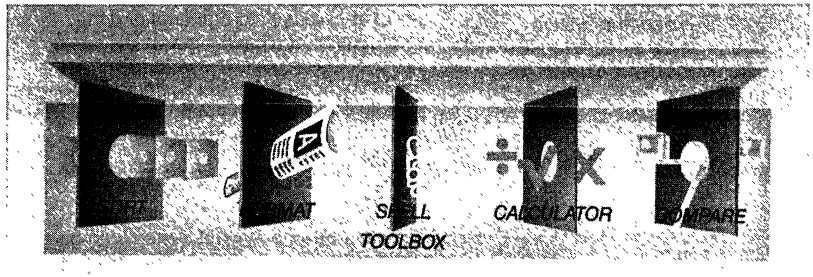
ORDER CODE

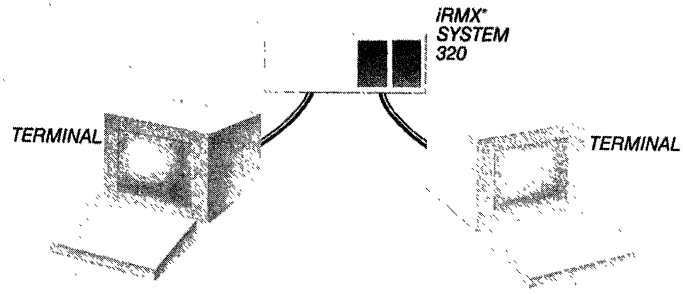
RMXSCSSU

ORDERING INFORMATION

For more information or the number of your nearest sales office call 800-548-4725 (good in the U.S. and Canada).

iRMX® TOOLBOX





The iRMX toolbox is a set of utilities to provide assistance to the software developer in the housekeeping aspects of program development. These utilities offer facilities for text processing and document preparation.

Sort facilities and a desk calculator are also included.

FEATURES:

- Text formatting
- Spelling verification
- File comparisons
- Sort
- Floating point desk calculator
- Pocket reference guide

September, 1989
Order Number 280737-001

FEATURES

TEXT FORMATTING (SCRIPT)

The SCRIPT utility is a text formatting program that streamlines document formatting and preparation. Commands include facilities to do paging, centering, left and right margins, justification, subscripts, superscripts, page headers and footers, underlines, boldface type, upper and lower case, etc.

Input text which has been prepared using the AEDIT utility can be formatted using the SCRIPT utility and the output directed to a printer or stored on disk for future manipulation. A short tutorial example is provided to help the first time user of this formatter.

SPELLING VERIFICATION (SPELL, WSORT)

The SPELL utility finds misspelled words in a text file. The included dictionary can be expanded by the user for any additions as well as specialized vocabularies. This utility can be used interactively or in a batch mode. Another utility (WSORT) then can be used to sort and compress the user created dictionary.

FILE COMPARISONS (COMP)

The COMP utility performs line oriented text file comparisons showing changes between text or source files. This utility can also compare object files.

SORT (ESORT, HSORT)

Files can be sorted on multiple keys (or fields) in ascending or descending order and the resultant sorted files stored.

Another utility can be invoked to sort records or data in ASCII lexical order.

FLOATING POINT DESK CALCULATOR (DC)

The DC utility accepts lines of text as input. Each line containing an expression is parsed, evaluated and the result displayed on the console. Expressions can contain embedded assignment statements and single letter variables.

POCKET REFERENCE GUIDE

In addition to the User's Guide provided with iRMX Toolbox, a reference guide in small pocket format provides a handy reference to commands and functions.

SPECIFICATIONS

OPERATING ENVIRONMENT

iRMX 286 Operating System Release 2.0 or later running on an Intel Series 300 System or equivalent hardware with Numeric Data Processor (NDP) support and at least 1MB of memory. The AEDIT utility is required for use of the SCRIPT text formatting program.

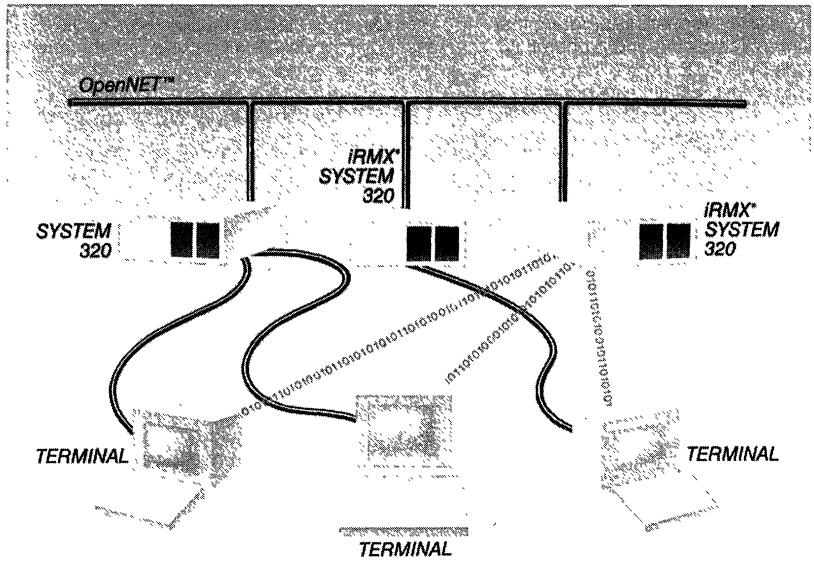
DOCUMENTATION

An iRMX 286 Toolbox User's Guide and Pocket Reference Guide are shipped with the product.

ORDERING INFORMATION

Product Code: RMX286TLB

The product is shipped on a 5¼" iRMX formatted floppy diskette.



Virtual Terminal is a network service of Intel's iRMX-NET network file access (NFA) product. Virtual Terminal allows local iRMX users to "Logon" to a remote Intel iRMX node within an OpenNET network. This capability enables users to access all the available resources on the remote system. In addition the iRMX Virtual Terminal is fully interoperable with DOS-NET Virtual Terminal and with the XENIX-NET Virtual Terminal. Now a PC or XENIX user can "connect" to a remote iRMX system without the need to use a locally connected iRMX terminal.

FEATURES

- User configurable
- Interoperable with Intel's OpenNET VT products
- Administration utility included

September, 1989
Order Number 280741-001

USER CONFIGURABLE

The iRMX Virtual Terminal server can be configured to support from 1 to 32 virtual terminal connections per system.

INTEROPERABILITY WITH OpenNET VT PRODUCTS

The iRMX Virtual Terminal interoperates with both the DOS-NET VT and XENIX-NET VT products.

A user on a PC who has the DOS-NET VT product installed can "Logon" to an iRMX system on the network.

A user on an iRMX system can "Logon" to a XENIX system and a user on a XENIX system with the XENIX-NET VT product can "Logon" to an iRMX system.

ADMINISTRATION UTILITY

The administration utility allows the system manager to disable, terminate or start the iRMX Virtual Terminal server. The capability to report on the status of all the virtual terminal connections to the local server is also supported.

SPECIFICATIONS

SOFTWARE PREREQUISITE

iRMX 286 Release 2.1 or later
iRMX-NET Release 2.1 or later

HARDWARE PREREQUISITE

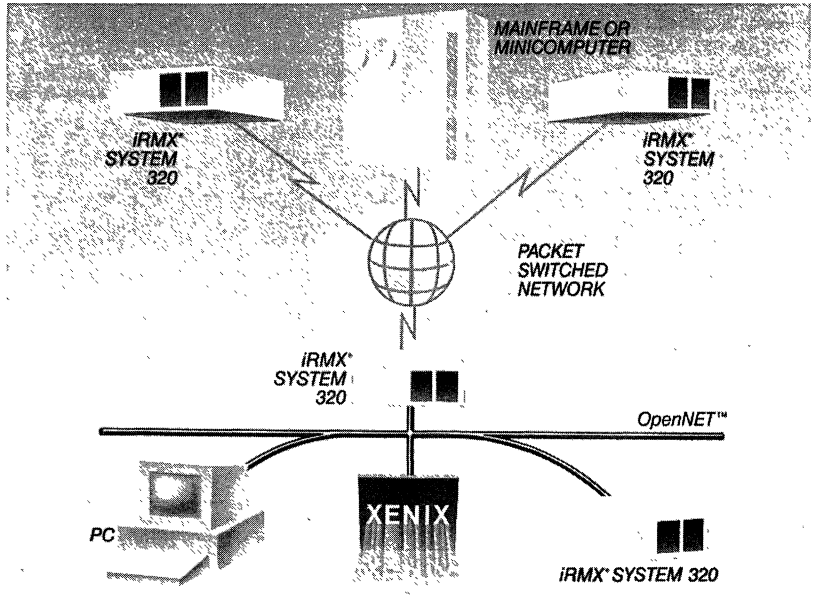
System 320 with NLAN option
or
System 310 with iSBC 552A

ORDER CODE

RMXNETVTSU

ORDERING INFORMATION

For more information or the number of your nearest sales office call 800-548-4725 (good in the U.S. and Canada).



iRMX® X.25 COMMUNICATIONS SOFTWARE

The iRMX X.25 Communications Software provides routines to connect an iRMX System 320 to a Packet Switch Network (PSN). The iRMX X.25 software allows connections of similar as well as dissimilar computer types that support the CCITT X.25 1980/1984 recommendation.

The iRMX X.25 software has been designed to allow the programmer the greatest flexibility in accessing packet-switch networks. In order to achieve this functionality, the programmer has access to a full-function programmatic interface. The design of iRMX X.25 allows not only host computer access as a Data Terminal Equipment (DTE) device, but in addition as a Data Circuit-terminating Equipment (DCE) device. The DCE configuration makes possible the programming of a complete packet-switch network service.

SOFTWARE FEATURES

- Application interface library
- Interactive utility package
- Conforms to CCITT X.25 1980
- User Selectable X.25 variants
- User Configurable
  - —Four physical links supported
  - —Software configurable Baud Rates
  - —Configurable as DTE/DCE
  - —255 Configurable Virtual Circuits (Permanent or Switched)

September, 1989
Order Number 280738-001

APPLICATION INTERFACE LIBRARY

Intel's software provides a three-level application interface library. Library routines are grouped into packet transfer services, network services, and management services. The user can choose the level of application interface which matches his X.25 experience. Those new to X.25 may prefer to start with network services routines, while proficient users will work directly with the packet transfer routines.

INTERACTIVE UTILITY PACKAGE

Several utility packages are included with the iRMX X.25 Communication Software that make it very approachable by the user. One of these tools is the User Confidence Test (UCT). The UCT has two modes of operation: a tutorial mode that demonstrates the use of the interface routines to help users quickly learn the calls to X.25; and, an interpreter mode that provides facilities to confirm the correct operation of iRMX X.25.

The UCT has been designed to assist users in testing X.25 applications. In addition to the UCT is CXTEST and CXPerform. Both of these utilities allow the user to gain more familiarity with X.25.

The product includes user documentation with detailed interface procedures, application examples with source code, and performance tools.

USER CONFIGURABLE

A configuration utility is provided to assist users in selecting the appropriate certification interface (see list under specification) for their X.25 network.

Once the user has selected their required network interface and specific parameters, the appropriate X.25 software routines are downloaded into the memory of the system's intelligent communication subsystem. This download capability allows the user application to run independently of the communications subsystem.

Under program control user may change the network configuration parameters. Some of these parameters are line baud rates (300 baud to 64K baud), packet size (maximum 1024 bytes supported) and retransmission limits.

The X.25 Communication Software can also be configured to support point-to-point interfaces via a serial link and a pair of modems.

SPECIFICATIONS

NETWORK CERTIFICATIONS

The products and services incorporating versions of X.25 have undergone extensive network certifications around the world.

A list of the countries where the software is known to have been successfully connected to the national network is given below:

COUNTRY	*NETWORK*	*APPROVAL*
Finland	Datapak	Yes
France	Transpac	N/A
Germany	Datex-P	#
Italy	Itapac	N/A
Netherlands	Datanet-1	#
South Africa	Saponet	Yes
Spain	Iberpac	N/A
Switzerland	Telepac	N/A
UK	PSS	Yes
USA	DDN	Yes
USA	GTE Telenet	Yes
USA	ATT Accunet	#

N/A—these PTTs have no formal approval procedure
#—is being certified

Many Packet switching networks are derivatives of early national implementations of the X.25 (1980) recommendation. The X.25 product is believed to be suitable for use on the following networks, based on these derivations:

COUNTRY	NETWORK
Australia	Austpac
Belgium	DCS
Canada	Datapac
Denmark	Datapak
Ireland	Eirepac
Israel	Isranet
Luxembourg	Luxpac
Norway	Datapac
Portugal	Telepac
Singapore	Telepac

HARDWARE REQUIREMENTS:

System 320 with H4 Communications Option supports up to four (4) links (2 links full DMA, 2 links with transmit only DMA)

SOFTWARE PREREQUISITE:

iRMX 286 Release 2.0 or later

ORDER CODE:

System 320 Option HRX25SU (software)

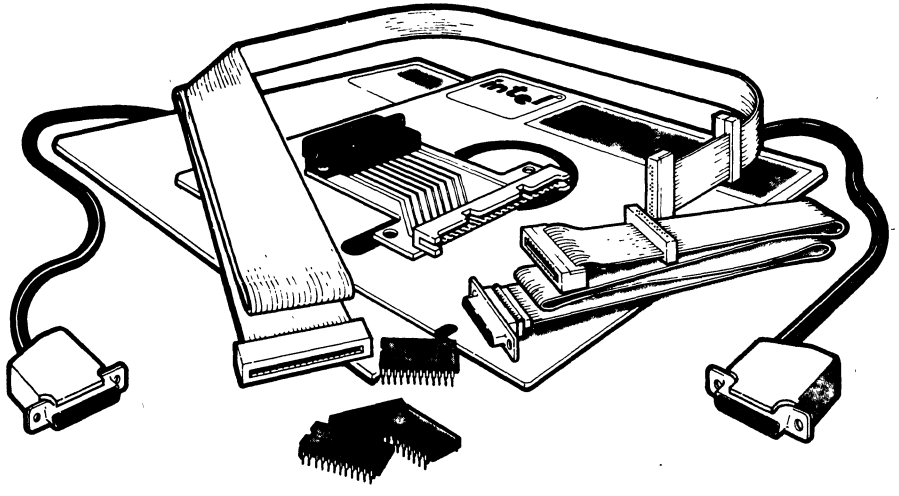
ORDERING INFORMATION

For more information or the number of your nearest sales office call 800-548-4725 (good in the U.S. and Canada).

iSDM™
SYSTEM DEBUG MONITOR

- **Supports Target System Debugging for iSBC® 8086, 8088, 80186, 80188, 80286 and 386™ CPU-Based Applications**
- **Provides Interactive Debugging Commands Including Single-Step Code Execution and Symbolic Displays of Results**
- **Supports 8087, 80287, and 80387 Numeric Processor Extensions (NPX) for High-Speed Math Applications**
- **Allows Building of Custom Commands Through the Command Extension Interface (CEI)**
- **Supports Application Access to ISIS-II Files**
- **Provides Program Load Capability from iSBC 8086, 80286 and 386 CPU-Based iRMX® I and II Development Systems and from an Intellec® Development System**
- **Contains Configuration Facilities which Allow an Applications Bootstrap from iRMX® File Compatible Peripherals**
- **Modular to Allow Use from an Intellec® Development System, from a Stand-Alone Terminal or from iRMX I or iRMX II Based Systems**

The Intel iSDM™ System Debug Monitor package contains the necessary hardware, software, cables, EPROMs and documentation required to interface, through a serial or parallel connection, an iSBC 86/05A, 86/12A, 86/14, 86/30, 86/35, 88/25, 88/40A, 88/45, 186/03A, 186/51, 188/48, 188/56, 286/10A, 286/12/14/16, 386/2X, 386/3X, 386/1XX or 8086, 8088, 80186 or 80188, 80286 and 386 CPU-based target system to a Series III, or Series IV Intellec® Microcomputer Development System or iRMX I or II Based System for execution and interactive debugging of applications software on the target system. The Monitor can: load programs into the target system; execute the programs instruction by instruction or at full speed; set breakpoints; and examine/modify CPU registers, memory content, and other crucial environmental details. Additional custom commands can be built using the Command Extension Interface (CEI). The Monitor supports the OEM's choice of the iRMX I Operating System, the iRMX II Operating System, or a custom system for the target application system. OEM's may utilize any iRMX supported target system peripheral for a bootstrap of the application system or have full access to the ISIS-II files of the Intellec System or the iRMX file system.



230882-1

October 1988
Order Number: 230882-005

FUNCTIONAL DESCRIPTION

Overview

The iSDM Monitor extends the software development capabilities of an iRMX or Intellec system so the user can effectively develop applications to ensure timely product availability.

The iSDM package consists of four parts:

- The loader program
- The iSDM Monitor
- The Command Extension Interface (CEI)
- The UDI Library Interface

The user can use the iSDM package to load programs into the target system from the development system, execute programs in an instruction-by-instruction manner, and add custom commands through the command extension interface. The user also has the option of using just the iSDM Monitor and the CEI in a stand-alone application, without the use of a development system.

Powerful Debugging Commands

The iSDM Monitor contains a powerful set of commands to support the debugging process. Some of the features included are: bootstrap of application software; selective execution of program modules based on breakpoints or single stepping requests; examination, modification and movement of memory contents; examination and modification of CPU registers, including NPX registers. All results are displayed in clearly understandable formats. Refer to Table 1 for a more detailed list of the iSDM monitor commands.

Numeric Data Processor Support

Arithmetic applications utilizing the 8087, 80287 or 80387 Numeric Processor Extension (NPX) are fully supported by the iSDM Monitor. In addition to executing applications with the full NPX performance, users may examine and modify the NPX's registers using decimal and real number format.

This feature allows the user to feel confident that correct and meaningful numbers are entered for the application without having to encode and decode complex real, integer, and BCD hexadecimal formats.

Command Extension Interface (CEI)

The Command Extension Interface (CEI) allows the addition of custom commands to the iSDM Monitor commands. The CEI consists of various procedures that can be used to generate custom commands. Up to three custom commands (or sets of commands) can be added to the monitor without programming new EPROMs or changing the monitor's source code.

Table 1. Monitor Commands

Command	Function
B	**Bootstrap** application program from target system peripheral device
C	**Compare** two memory blocks
D	**Display** contents of memory block
E*	**Exit** from loader program to iRMX or ISIS-II Interface
F	**Find** specified constant in a memory block
G	**Execute** application program
I	**Input** and display data obtained from input port
K	**Echo** console display to a file
L*	**Load** absolute object file into target system memory
M	**Move** contents of memory block to another location
N	**Display and execute** single instruction
O	**Output** data to output port
P	**Print** values of literals
R*	**Load and execute** absolute object file in target system memory
S	**Display and (optionally) modify** contents of memory
U, V, W	**User** defined custom commands extensions
X	**Examine and (optionally) modify** CPU and NPX registers
Y	**Display/Define** 80286 compiler symbol information

*Commands require an attached development system.

Universal Development Interface

The Universal Development interface (UDI) consists of libraries that contain interfaces to iRMX and ISIS II I/O calls. A program running on an 8086, 8088, 80186, 80188, 80286, or 386 CPU-based system can use UDI and access iRMX and ISIS II I/O calls. The interface allows the inclusion of these calls into the program; however, most of the calls require an iRMX or Intellec host system. Table 2 contains a summary of the major I/O calls.

Program Load Capability

The iSDM loader allows the loading of 8086, 8088, 80186, 80188, 80286 or 386 CPU-based programs into the target system. It executes on a development system and communicates with the target system through a serial or a parallel load interface.

Configuration Facility

The monitor contains a full set of configuration facilities which allows it to be carefully tailored to the requirements of the target system. Pre-configured EPROM-resident monitors are supplied by Intel for the iSBC 86/05A, 86/12A, 86/14, 86/30, 86/35, 88/25, 88/10A, 88/45, 186/03A, 186/51, 188/48, 188/56, 286/10A, 286/12/14/16, 386/2X/3X, and 386/1XX boards. The monitor must be configured by the user for other 8086, 8088, 80186, or 80188 applications. iRMX I and iRMX II system users may use the configuration facilities to include the Bootstrap Loader (V5.0 or newer) in the monitor.

Variety of Connections Available

The physical interface between the development system and the target system can be established in one of three ways. The systems can be connected via a serial link, a parallel link or a fast parallel link. The cabling arrangement is different depending upon the development system being used.

The iSDM Monitor does not require the use of a development system. The monitor can be used by simply attaching a stand-alone terminal to the target system.

Table 2. Routines for Services Available to Target System Applications

Routine	Target System Function
DQ$ATTACH	Creates a connection to a specified file.
DQ$CLOSE	Closes the specified file connection.
DQ$CREATE	Creates a file for use by the application.
DQ$DELETE	Deletes a file.
DQ$DETACH	Closes a file and deletes its connection.
DQGETCON-NECTION$STATUS	Returns status of a file connection.
DQ$OPEN	Opens a file for a particular type of access.
DQ$READ	Reads the next sequence of bytes from a file.
DQ$RENAME	Renames the specified file.
DQ$SEEK	Moves the current position pointer of a file.
DQ$SPECIAL	Defines options and actions for the program execution environment.
DQ$TRUNCATE	Truncates a file to the specified length.
DQ$WRITE	Writes a sequence of bytes to a file.

SPECIFICATIONS

Hardware

- Supported iSBC Microcomputers:
  - iSBC 86/05A Single Board Computer
  - iSBC 86/12A Single Board Computer
  - iSBC 86/14 Single Board Computer
  - iSBC 86/30 Single Board Computer
  - iSBC 86/35 Single Board Computer
  - iSBC 88/25 Single Board Computer
  - iSBC 88/40A Single Board Computer
  - iSBC 88/45 Single Board Computer
  - iSBC 186/03A Single Board Computer
  - iSBC 186/51 Single Board Computer
  - iSBC 188/48 Single Board Computer
  - iSBC 186/56 Single Board Computer
  - iSBC 286/10A Single Board Computer
  - iSBC 286/12/14/16 Single Board Computer
  - iSBC 386/2X/3X Single Board Computer
  - iSBC 386/1XX Single Board Computer
- Supported iSBX™ MULTIMODULE Boards:
  - iSBX 351 Serial I/O MULTIMODULE Board
  - iSBX 354 Serial I/O MULTIMODULE Board
- Supported Microcomputer Systems
  8086/8088/80186/80188/80286/386/CPU
  8087/80287/80387 NPX with Serial Controller:
  8274 Serial Controller and 8253/8254 timer, or
  8251A Serial Controller and 8253/8254 timer, or
  82530 Serial Controller 4 KB RAM, and 32 KB EPROM

iSDM™ Package Contents

Cables:
- 4—RS232 Cable Assemblies (for iRMX/Intellec host system and standard terminals)
- Hardware package for the cable assemblies

Interface and Execution Software Diskettes:
- 2—DS/DD, iRMX-Format 5¼″
- 2—SS/DD, iRMX—Format 8″
- 2—SS/DD, ISIS II-Format 8″

System Monitor EPROMs:

Intel Board	EPROM Description
iSBC 86/05A iSBC 86/14 iSBC 86/30 iSBC 86/35	Two 27128 EPROMs
iSBC 86/12A	SUBMIT Files on the Release Diskette

System Monitor EPROMs: (Continued)

Intel Board	EPROM Description
iSBC 88/25	Two 27128 EPROMs
iSBC 88/40A	Two 27128 EPROMs
iSBC 88/45	Two 27128 EPROMs
iSBC 186/03A	Two 27128 EPROMs
iSBC 186/51	Two 27128 EPROMs
iSBC 188/48 iSBC 188/56	Two 27128 EPROMs
iSBC 286/10A iSBC 286/12/14/16	Two 27128 EPROMs
iSBC 386/2X/3X	Two 27256 EPROMs
iSBC 386/1XX	Two 27256 EPROMs

Reference Manual (Supplied):

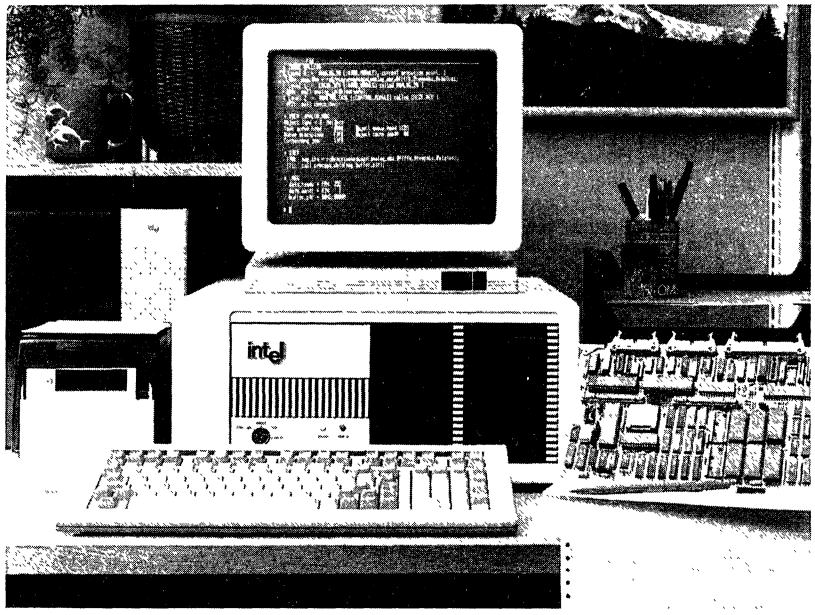
iSDM System Debug Monitor Installation and Configuration

iSDM System Debug Monitor User's Guide

ORDERING INFORMATION

Part Number	Description
SDMSC	Object Software iRMX and Intellec host to target system interface and target system monitor, suitable for use on iSBC 86, 88, 186, 188, 286, 386 computers, or other 8086, 8088, 80186, 80188, 80286, 386 microcomputers. Package includes cables, EPROMs, software and reference manual. The OEM license option listed here allows use on a single host/target system and incorporation into their applications. Each incorporation requires payment of an Incorporation Fee. The iSDM package also includes 90 days of support services that include Software Program Report Services. As with all Intel Software, purchase of any of these options requires execution of a standard Intel Software License Agreement.
SDMRFX	Incorporation fee. Permits incorporation of a configured iSDM monitor into a target system.

SOFT-SCOPE* II SOURCE-LEVEL DEBUGGER



SOURCE-LEVEL ON-TARGET DEBUGGER FOR iRMX® II APPLICATIONS

The SOFT-SCOPE II Debugger is an interactive debugging tool specifically designed for software developed to execute with the iRMX II Operating Systems on Intel's broad set of system and board-level products. It reduces the time required to debug real-time software and allows the developer to debug at the most effective level, in the original source code itself.

FEATURES:

- Complete High-Level Debugging Functionality
- Source Code Interface and On-line Listings
- Automatic Expansion of Data Types
- Symbolic Display of iRMX II Objects
- Second Terminal Option for "Remote" Debugging
- Multitasking Support
- Handling of 80286/386™ Protection Traps and Software Exceptions

* SOFT-SCOPE is a registered trademark of Concurrent Sciences, inc

intel

September, 1989
Order Number. 280637-002

COMPLETE HIGH-LEVEL DEBUGGING

For real-time applications running with iRMX II Operating Systems on Intel 80286/386 CPU boards, software professionals want to focus on original source code for most debugging operations. SOFT-SCOPE II does just that. It integrates the original source code into the debugging process directly. All breakpoint prompts and high-level stepping operations prompt with original source code rather than reporting what line number the program has reached or what assembly instruction is next.

SOURCE CODE INTERFACE AND ON-LINE LISTINGS

The source code interface frees the programmer from having to divide attention between the console and program listings, eliminates the need to get a fresh program listing each time a small change is made, and reduces the time needed to make software modifications.

AUTOMATIC EXPANSION OF DATA TYPES

Symbols declared in the program are accessible by name for display and modification of contents. These symbols include arrays, structures, static variables, based variables, and stack-based variables (including local variables, re-entrant variables, and passed parameters). Memory can also be displayed with absolute references or with register-relative references.

SYMBOLIC DISPLAY OF ALL IRMX II SYSTEM OBJECTS

The VIEW command allows viewing the status of any iRMX II object including tasks, jobs, mailboxes, semaphores, regions, and segments. With VIEW, the stack of a task can be examined to determine which iRMX II call the task has made most recently. Any job's object directory and the list of ready and sleeping tasks can be examined.

SECOND TERMINAL OPTION

Because so many applications today are screen-intensive, the SOFT-SCOPE Debugger allows the option of using a second terminal for all debugger I/O, freeing the main console for exclusive use by the application for application output.

MULTI-TASKING SUPPORT

The SOFT-SCOPE II Debugger supports simultaneous debugging of concurrent tasks when they are all linked together as a Human Interface command and each concurrent task is coded in a separate module. SOFT-SCOPE loads and then allows the user to suspend and resume execution of the tasks from the command line with the SUSPEND and RESUME commands. In this way the developer can observe the effect of dynamic changes on the software under test.

HANDLING OF 80286/386 PROTECTION TRAPS AND SOFTWARE EXCEPTIONS

Exception Handling: The exact source line which causes an exception can easily be reached and displayed. All environmental and programmer exceptions are trapped and reported, without causing a SOFT-SCOPE debugger exit.

Most of the 80286/386 hardware traps are handled by the SOFT-SCOPE II Debugger, including Bounds Check (INT 5), Invalid Opcode (INT 6), Double Fault (INT 8), Stack Fault (INT 12), and General Protection (INT 13). Upon encountering one of these interrupts, the SOFT-SCOPE II Debugger breaks execution with a message similar to the following:

```
< General Protection fault (INT 13) >?
[ Break near line #145 in TESTPROC
(:TESTMODULE) ]
145: ARRAYX(INDEX) = XYZ;
```

In the above example, the General Protection trap could have been caused by the variable INDEX being too large for the segment which contained ARRAYX, or by ARRAYX being based on an undefined pointer. Because the debugger handles these traps directly, other users in a multi-user system won't even be aware in most cases that there was a hardware fault.

INTEL QUALITY—YOUR GUARANTEE

The SOFT-SCOPE Debugger is built to the same exacting standards as Intel's component and board products. This product's reliability is been proven in many real-time product settings over the past several years.

WORLDWIDE SUPPORT AND SERVICE—AN INTEL STANDARD

Standard support products for the SOFT-SCOPE Debugger are available through Intel's support organization.

SOFT-SCOPE II COMMANDS

Function	Commands	Function	Commands
Invoking the Debugger	SSCOPE	Debugger I/O	CONSOLE, ECHO
Viewing the Source Code	LIST, LINE, FIND	Examining Registers	REG, NPXREG
Opening a Module	LINE, OPEN, MODULES, ASSIGN, DETACH	Requesting Help	HELP
		Setting Options	OPTIONS
		View iRMX II Objects	VIEW
Controlling Execution	GO, STEP, ASTEP	Start/Stop Tasks	SUSPEND, RESUME
Setting Breakpoints	GO, STEP, BREAKPOINT, RANGE	Execute a Submit File	SUBMIT
		Run an RMX Program	RUN
Examine Data/ Descriptor	.variable, TYPE, EVAL, DUMP, STACK, ADDR	Stack Trace	NEST
		Initialize Stack	STACKINIT
Examining Code	LINE, ASM		
Port I/O	IN, OUT		

SPECIFICATIONS

The SOFT-SCOPE II Debugger supports the following languages:

Intel PL/M 286
Intel C 286
Intel Pascal 286
Intel ASM 286
Intel FORTRAN 286

The SOFT-SCOPE Debugger requires a full configuration of the iRMX II Operating System, including the Human Interface and UDI. Either the iSDM System Debug Monitor or the D-MON monitor must be included in the firmware of the CPU board to use the VIEW command. The debugger utilizes approximately 130k bytes of memory beyond the load size of the program being debugged.

ORDERING INFORMATION

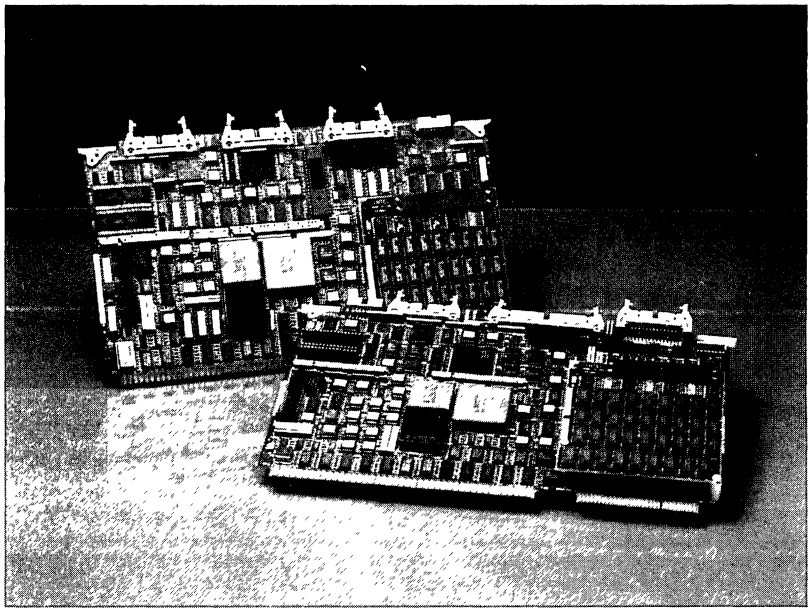
Product Code	Product Contents
RMXIISFTSCP	SOFT-SCOPE II Debugger for iRMX II Operating Systems of 5¼" iRMX media diskettes

For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada).

MULTIBUS® I
Single Board Computers

8

iSBC® 386/12* AND iSBC® 386/12S* SINGLE BOARD COMPUTERS



HIGH INTEGRATION 386™ MICROPROCESSOR-BASED MULTIBUS®I SINGLE BOARD COMPUTERS

The iSBC® 386/12 and iSBC® 386/12S products are high performance, high integration 386™ microprossor-based MULTIBUS®I Single Board Computers. These boards share a common core design that provides functional compatibility with the iSBC 286/12 Single Board Computer with a 2X performance improvement.

The iSBC 386/12 Single Board Computer combines 386 microprocessor performance and the iSBC 286/12 I/O functionality. The iSBC 386/12S board provides the CPU performance and I/O functionality supplied on the iSBC 386/12 board, plus support for the industry standard, high-speed SCSI bus (Small Computer System Interface) on a single MULTIBUS I SBC board. SCSI data transfers to local DRAM on the iSBC 386/12S are DMA supported and FIFO buffered allowing for sustained synchronous SCSI data rates up to 5 MBytes/sec.

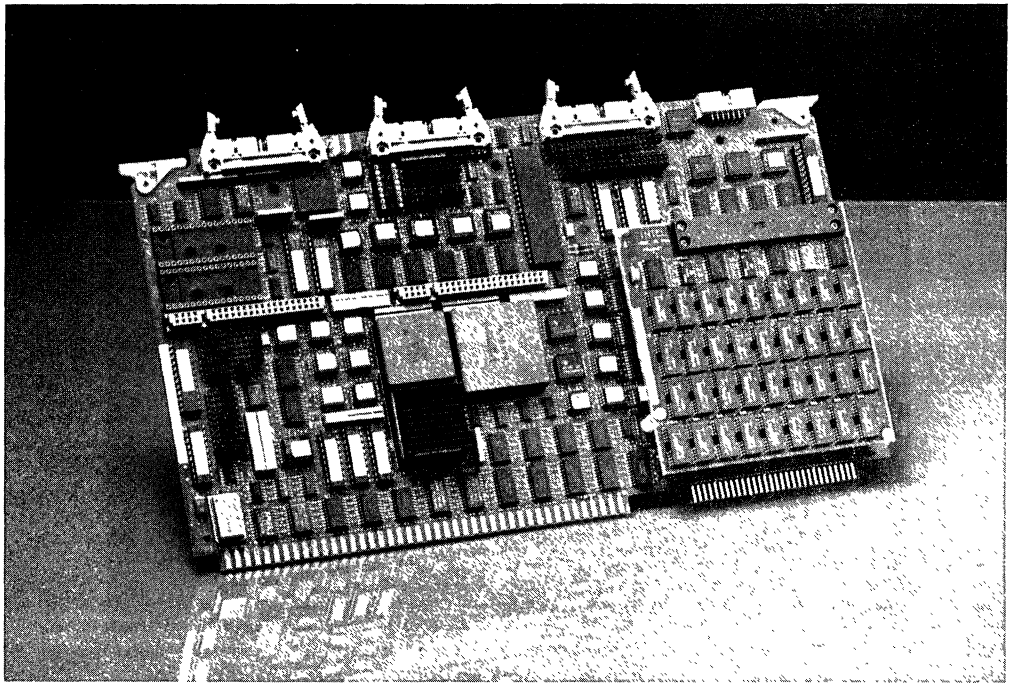
FEATURES

- 20 MHz 386™ Microprocessor
- Optional 387™ Numeric Coprocessor
- 82258 Advanced DMA Controller
- Dual Bus Architecture—Dedicated 32-bit processor execution bus
- 1–16MB on-board 32-bit dual-port parity DRAM
- SCSI Interface—5.0 MBytes/sec synchronous data rate
- Two 32-pin JEDEC EPROM sites
- Two serial ports via 8274
- 82C55A parallel port
- Three on-board timers via 82C54
- Two iSBX™ Bus interface connectors
- iLBX™ Bus support
- Multiprocessor support, including memory aliasing
- Functional superset of iSBC® 86/XX and 286/1X Single Board Computers

* The iSBC 386/12 and iSBC 386/12S boards are also manufactured under product codes pSBC386/12 and pSBC386/12S by Intel Puerto Rico, Inc

September, 1989
Order Number 281014-001

FEATURES



iSBC® 386/12 Single Board Computer

386™ MICROPROCESSOR PERFORMANCE

The iSBC 386/12 and 386/12S CPU boards feature a 386™ microprocessor running at 20 MHz. The on-board 32-bit processor execution bus optimizes the compute performance of the 386 microprocessor. The on-board 32-bit memory bus provides support for up to 16MB of zero wait-states (pipelined read hit) parity DRAM. The iSBC 386/12 board is also available with a 16 MHz 386 microprocessor.

387™ NUMERIC COPROCESSOR

Numeric processing speed may be enhanced with the 32-bit 387™ floating point math coprocessor. The 387 provides 80-bit precision, accelerating floating point calculations through hardware execution.

DUAL-BUS ARCHITECTURE

The iSBC 386/12 and 386/12S CPU boards use multiple on-board busses, a design technique that increases overall system performance. The board utilizes both a high speed 32-bit execution bus for optimized CPU, memory and math operations and a dedicated I/O bus for on-board I/O, iSBX and EPROM operations.

HIGH-SPEED MEMORY CONTROLLER

1–16 MB of on-board DRAM memory is provided on modules attached to the CPU baseboard. On-board access is optimized by a high-speed memory controller and the latest DRAM technology. DRAM memory is dual-ported and fully addressable from the 32-bit on-board bus or from MULTIBUS I. On-board DRAM memory may be addressed above the 16 MB address space, allowing use of the first 16 MB of address space (no dual port) for EPROM, iLBX and/or MULTIBUS I memory.

The iSBC 386/12 and 386/12S boards provide support for two 32-pin JEDEC EPROM sockets. EPROM capacities up to 256 KBytes are attainable with 27010 devices. The iSBC 386/12 and 386/12S boards will support byte-wide 27020 and 27040 devices upon their respective availability.

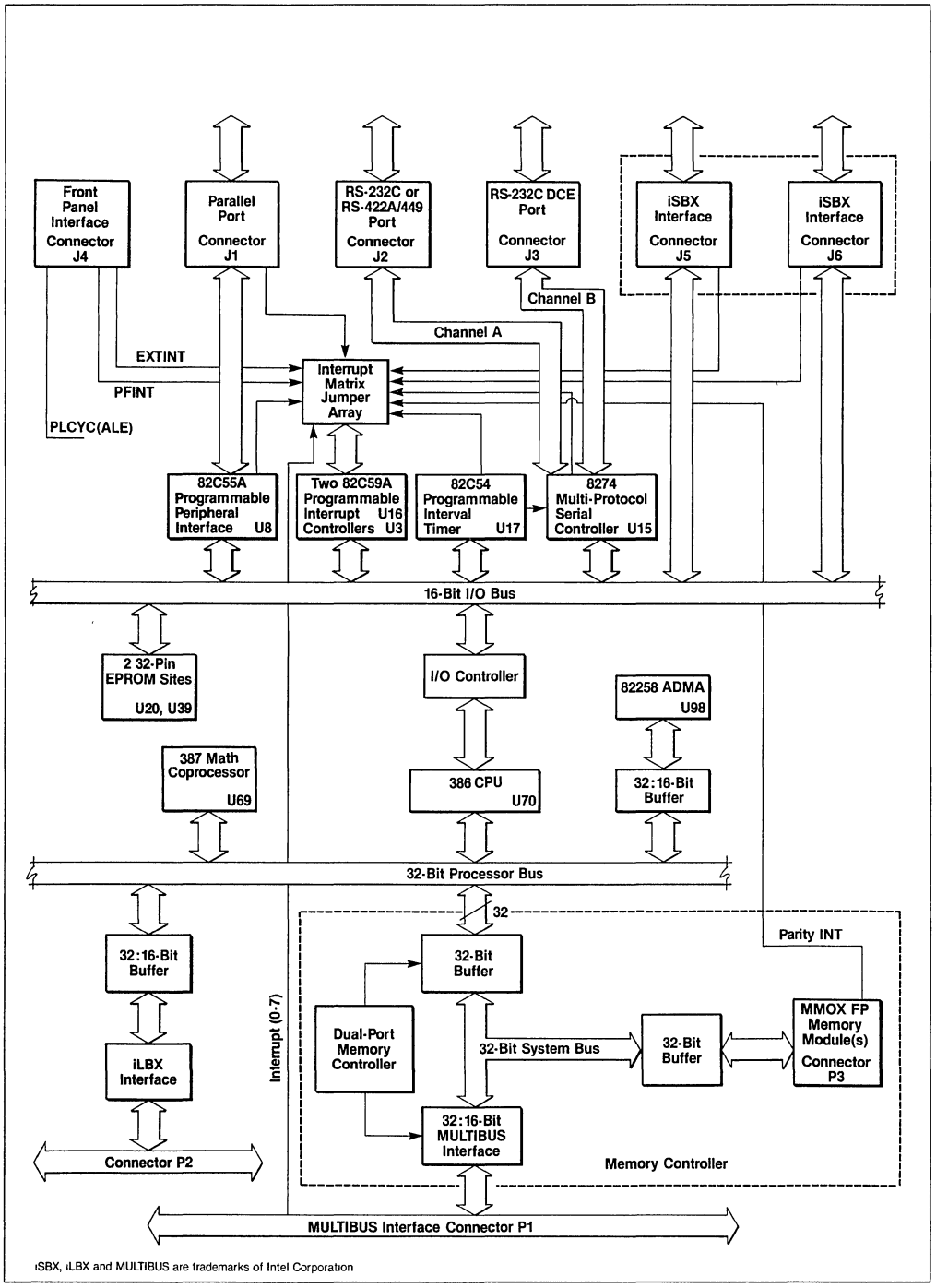
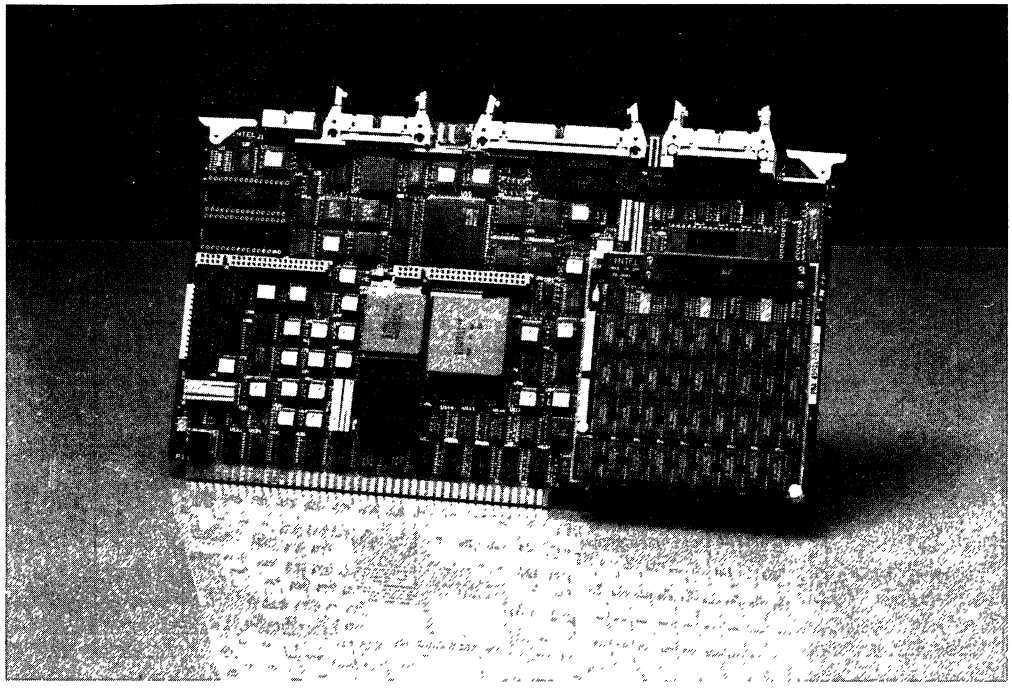


Figure 1: iSBC® 386/12 Block Diagram



iSBC® 386/12S Single Board Computer

SCSI BUS INTERFACE

The iSBC 386/12S board integrates a 10 MHz Fujitsu 87033B SCSI Protocol Chip (SPC). The 87033B SPC device supports a synchronous data rate of 5 MBytes/sec and an asynchronous rate of 2 MBytes/sec. SCSI data transfers are buffered into 16-bit wide, 512-bit deep bi-directional FIFOs and DMA transferred into (or from) local RAM at rates above the maximum incoming synchronous data rate. The SCSI implementation is ideal for applications requiring sustained high-speed SCSI data transfers.

The 87033B SPC device has integrated single-ended driver/receiver circuitry. These SCSI bus drivers support host to target connections of up to 6 meters.

The 87033B SPC device supports both SCSI bus initiator and target modes. In addition, the SPC device supports selection/reselection commands for optimum SCSI bus utilization. The iSBC 386/12S board supports target mode features in hardware only.

An iRMX Operating System driver for the SCSI interface is available from Intel. This driver, which resides on the system host, supports SCSI bus initiator mode only.

COMPLETE ON-BOARD I/O FEATURE SET

The iSBC 386/12 and 386/12S boards provide a fully integrated set of on-board I/O functionality. Two serial ports are proivded via Intel's 8274 Multi-Protocol Serial Controller. An 82C55A Programmable Peripheral Interface provides support for the parallel line printer interface. The iSBC 386/12 and 386/12S boards also provide interrupt support based on two 82C59A Programmable Interrupt Controllers and the 386 microprocessor's own Non Maskable Interrupt line. Programmable timer/counter functions are provided by an 82C54 Programmable Interval Timer.

Direct memory access transfers are provided by the 82258 advanced DMA (ADMA) controller. The ADMA controller has been further enhanced to increase flexibility and support synchronous transfers over the SCSI interface (386/12S), serial ports and iSBX Bus for high speed data transfers.

ISBX™ BUS FOR I/O EXPANSION

Two iSBX Bus interface connectors are provided for I/O expansion. Functionalities such as additional serial/parallel ports, graphics, BITBUS™, network interface or analog to digital modules can be added onto the iSBC 386/12 and 386/12S boards via this industry standard I/O expansion interface. The iSBX Bus interface is also ideal for integration of custom-built modules. The iSBC 386/12 and 386/12S boards support either two single-wide modules or one double-wide module.

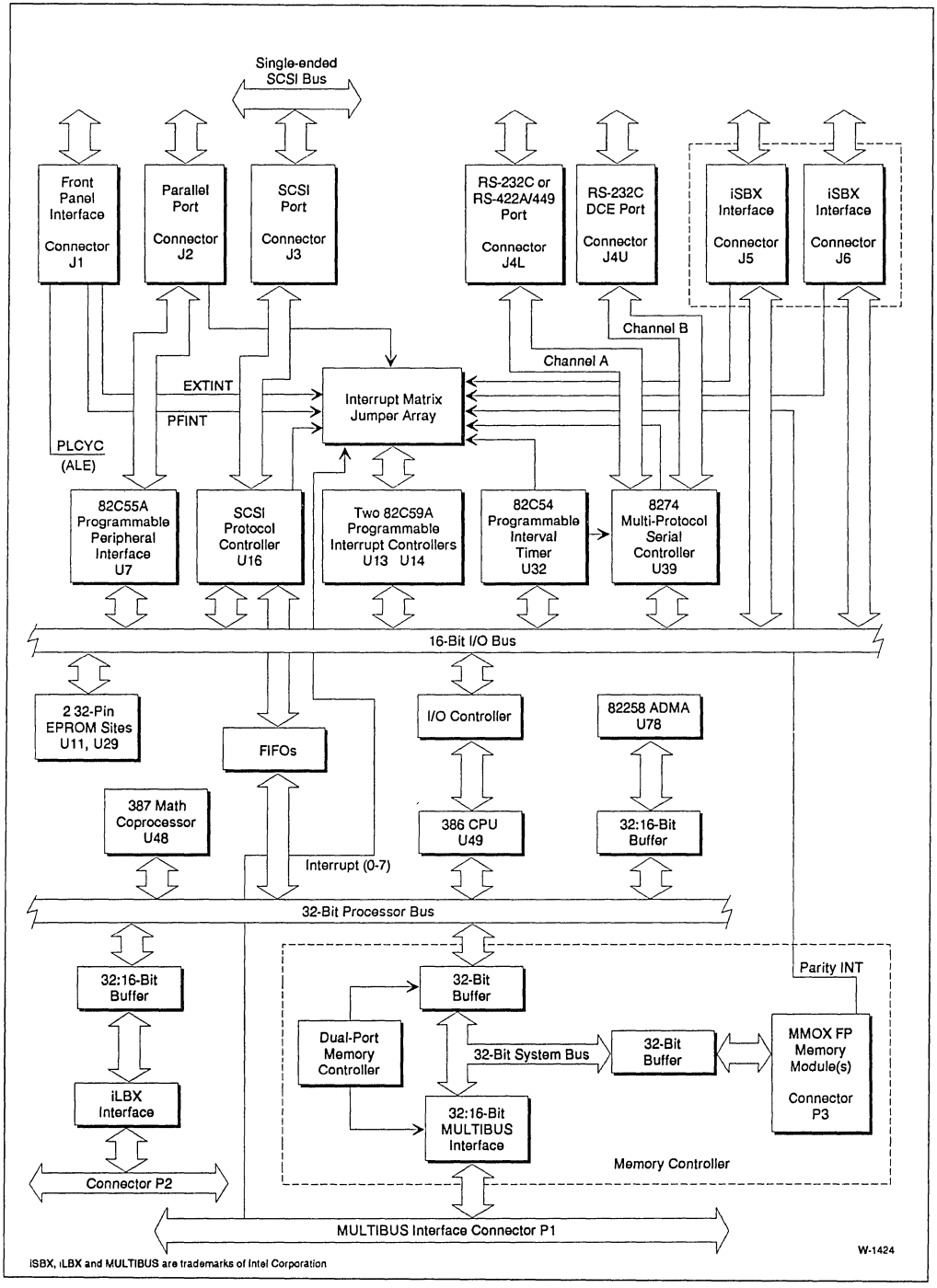


Figure 2: iSBC® 386/12S Block Diagram

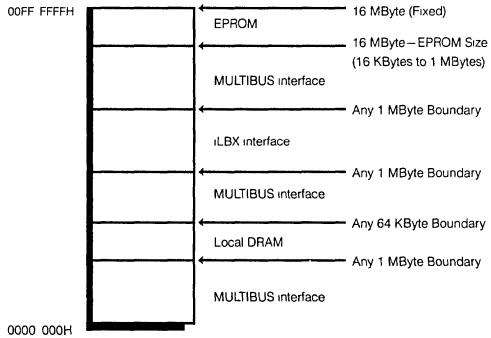
MULTIBUS®I, iSBX™ AND iLBX™ COMPLIANCE

The iSBC 386/12 and 386/12S boards support the full MULTIBUS I specification (including full Multimaster and bus-vectored interrupt support) and the published iSBX and iLBX (asynchronous) specifications.

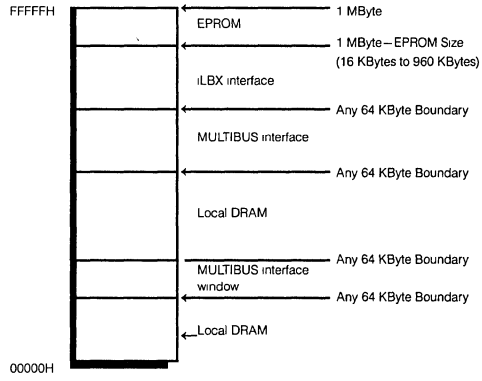
MULTI-PROCESSING SUPPORT

Multiprocessor support is enhanced through high performance dual-port arbitration control logic and features such as memory aliasing over MULTIBUS I, bus-vectored interrupts and real mode page registers. In addition, the addressing approach used for the iSBC 386/2X/3X series of boards is also supported.

MEMORY MAPS



Protected Mode Memory Map



Real Mode Memory Map

Figure 3: Memory Maps

SPECIFICATIONS

WORD SIZE

Instruction: 8, 16, 24, 32 or 40 bits
Data: 8, 16, or 32 bits

SYSTEM CLOCK

386 Microprocessor @ 20 MHz (or 16 MHz on iSBC 386/12 only)
387 Numeric Coprocessor @ 20 MHz (or 16 MHz on iSBC 386/12 only)

INTERRUPT CAPACITY

Interrupt sources: 26 total, 5 hard-wired to the 8259A PIC, 21 jumper selectable
Interrupt levels: 16 vectored requests using two 8259A devices and the 386 CPU's NMI line.
Supports both on-board and bus-vectored interrupts.

I/O CAPABILITY

Expansion: Two 8/16-bit iSBX MULTIMODULE connectors, supporting up to two single-wide or one double-wide iSBX MULTIMODULE board.
Parallel: Line printer interface, on-board functions, and 3-bit board installed options code.
Serial: Two programmable channels using one 8274 device
SCSI: Single-ended, synchronous/asynchronous
Timers: Three programmable timers using one 82C54 device
DMA: 10 MHz 82258 advanced DMA (ADMA) controller. Supports DMA block transfers between on-board memory over the MULTIBUS I interface, SCSI interface (386/12S), iLBX interface, both iSBX interfaces and both serial channels.

SERIAL COMMUNICATIONS CHARACTERISTICS

Synchronous: 5–8 bit characters; internal or HDLC/SDLC character synchronization; automatic sync insertion; even or odd parity; baud rates from 600 baud to 615 KB.

Asynchronous: 5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection; even or odd parity; baud rates from 75 baud to 19.2 KB.

INTERFACES

MULTIBUS I Bus. All signals TTL compatible
iSBX Bus: All signals TTL compatible
iLBX Bus: All signals TTL compatible
SCSI Interface: ANSI X3.131–1986

Serial I/O: Channel A: RS232C/RS422/RS449 compatible, DCE or DTE
Channel B: RS232C compatible, DCE
NOTE: For RS422/RS449 operation, line drivers and resistor terminators must be supplied.

Timer: All signals TTL compatible
Interrupt Requests: All TTL compatible

PHYSICAL CHARACTERISTICS

Width: 12.00 in. (30.48 cm)
Width: 7.05 in. (18.00 cm)
Depth: 0.86 in. (2.18 cm), 1.62 in. (4.11 cm) with added memory module

Recommended slot spacing: 1.2 in. (3.0 cm), 1.8 in. (4.6 cm) for double stacked memory modules.

ELECTRICAL CHARACTERISTICS

DC Power Requirements (Maximum):

	Volts	Current
Baseboard	+5.25	14.5A
Baseboard	±12	50mA
1 MByte memory module	+5.25	0.89A
2 MByte memory module	+5.25	1.2A
4 MByte memory module	+5.25	0.89A
8 MByte memory module	+5.25	1.2A
387™ math coprocessor	+5.25	0.31A

ENVIRONMENTAL CHARACTERISTICS

Operating Temperature: 0°C to 60°C at 8 CFM airflow across board (default configuration)
Relative Humidity: 0% to 90% (without condensation)
Storage Temperature: –40°C to +70°C

REFERENCE MANUALS

455528-001 iSBC 386/12 Series Hardware Reference Manual (order separately). Provides complete information on hardware features, installation, jumpering, memory maps, addressing, and schematics.

459913-001 iSBC 386/12S Series Hardware Reference Manual (order separately)

462435-001 iSBC 386/12 Series Software Upgrade Guide (order separately). Provides complete information on upgrading existing applications (hardware and software), from an iSBC 286/10A/12/14/16, iSBC 86/30/35, or iSBC 386/2X/3X to an iSBC 386/12 board.

ORDERING INFORMATION

Part Number	*Description*
iSBC 386/12S-F01	20 MHz iSBC 386/12S with 1 MB DRAM
iSBC 386/12S-F02	20 MHz iSBC 386/12S with 2 MB DRAM
iSBC 386/12S-F04	20 MHz iSBC 386/12S with 4 MB DRAM
iSBC 386/12S-F08	20 MHz iSBC 386/12S with 8 MB DRAM
iSBC 386/12-20F01	20 MHz iSBC 386/12 with 1 MB DRAM
iSBC 386/12-20F02	20 MHz iSBC 386/12 with 2 MB DRAM
iSBC 386/12-20F04	20 MHz iSBC 386/12 with 4 MB DRAM
iSBC 386/12-20F08	20 MHz iSBC 386/12 with 8 MB DRAM
iSBC 386/12-16F01	16 MHz iSBC 386/12 with 1 MB DRAM
iSBC 386/12-16F02	16 MHz iSBC 386/12 with 2 MB DRAM
iSBC 386/12-16F04	16 MHz iSBC 386/12 with 4 MB DRAM
iSBC 386/12-16F08	16 MHz iSBC 386/12 with 8 MB DRAM

Models with an installed 387 coprocessor may be ordered by adding an M suffix to the above order codes. For example, an iSBC 386/12-20F08M is the order code for a 20 MHz iSBC with 8 MB DRAM memory and installed 387 math coprocessor.

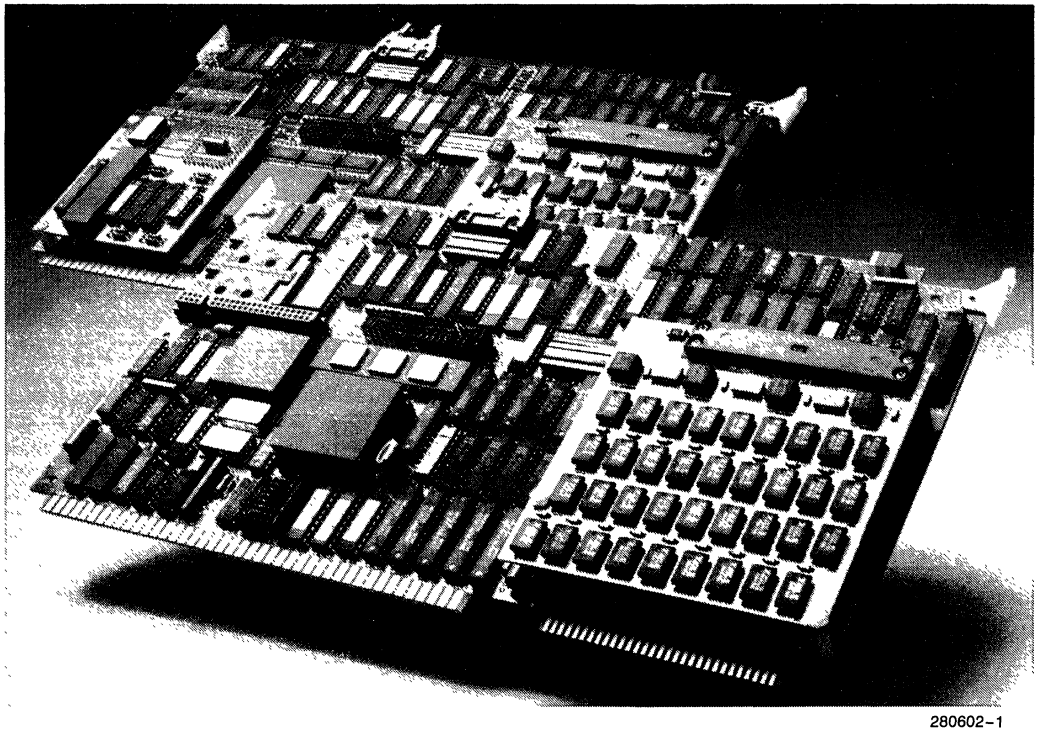
Part Number	*Description*
iSBC MM01FP	1 MB, 85ns parity DRAM memory expansion module
iSBC MM02FP	2 MB, 85ns parity DRAM memory expansion module
iSBC MM04FP	4 MB, 85ns parity DRAM memory expansion module
iSBC MM08FP	8 MB, 85ns parity DRAM memory expansion module

For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada).

iSBC® 386/21/22/24/28 AND 386/31/32/34/38* SINGLE BOARD COMPUTERS

- **Choice of 16 MHz or 20 MHz 386™ Microprocessor**
- **Available with 1, 2, 4, or 8 Megabytes of On-Board 32-Bit Memory, expandable to 16 Megabytes**
- **High Speed 80387 Floating Point Math Coprocessor**
- **Two 32-Bit JEDEC Sites for up to 512 Kilobytes of EPROM Memory**
- **RS232C Interface for Local/Remote Control and Diagnostics**
- **iSBX® Interface for Low Cost I/O Expansion**
- **16 Levels of Direct Vectored Interrupt Control**
- **64 Kilobyte 0 Wait-State Cache Memory**

The iSBC® 386/2x and 3x series boards (iSBC 386/21/22/24/28 and iSBC 386/31/32/34/38) are Intel's highest performance MULTIBUS® I CPU boards. These boards feature either a 16 MHz or 20 MHz 386 CPU, an 80387 math coprocessor, a 64k byte, 0 wait-state cache memory to support the CPU, and a 32-bit interface to 1, 2, 4, or 8 megabytes of dual-port parity DRAM memory. An additional 1, 2, 4, or 8 MB iSBC MM0x series memory module may be installed to provide up to 16 MB of on-board DRAM memory. The iSBC 386/2x and 3x boards also feature an 8/16-bit iSBX MULTIMODULE interface for low-cost I/O expansion, an asynchronous RS232C interface to support a local terminal or modem, two 16-bit programmable timer/counters, a 16-level direct-vectored interrupt controller, two 32-pin JEDEC sites and multimaster MULTIBUS arbitration logic.



280602-1

*The iSBC® 386/21/22/24/28 and iSBC® 386/31/32/34/38 Boards are also manufactured under product code piSBC® 386/21/22/24/28 and piSBC® 386/31/32/34/38 by Intel Puerto Rico, Inc.

*XENIX is a registered trademark of Microsoft Corp.
*UNIX is a trademark of AT&T.

October 1989
Order Number: 280602-002

OVERVIEW—iSBC 386/2x AND 3x SERIES CPU BOARDS

The iSBC 386/21/22/24/28 and iSBC 386/31/32/34/38 boards (iSBC 386/2x and 3x series) are 16 MHz and 20 MHz versions of Intel's first MULTIBUS I 386™ microprocessor CPU boards. The boards employ a dual-bus structure, a 32-bit CPU bus for data transfers between the CPU and memory, and a 16-bit bus for data transfers over the MULTIBUS interface, iSBX interface, EPROM local memory, and I/O interfaces. The boards take advantage of the 386 CPU's 32-bit performance while maintaining full compatibility with the MULTIBUS I interface and iSBX MULTIMODULE boards.

The DRAM memory, which is on a module that is secured to the baseboard, may be expanded by installing a second 1, 2, 4, or 8M byte memory module.

16 MHz or 20 MHz Central Processor Unit

The heart of the iSBC 386/2x and 3x CPU board is the 386 microprocessor. The complete series includes two lines, with a choice of CPU speed. The iSBC 386/21/22/24/28 boards use the 16 MHz 386 microprocessor and the iSBC 386/31/32/34/38 boards use the 20 MHz 386 microprocessor. The 386 CPU utilizes address pipelining, a high speed execution unit, and on-chip memory management/protection to provide the highest level of system performance. The 386 microprocessor also features an Address Translation Unit that supports up to 64 terabytes of virtual memory.

The 386 CPU is upward compatible from Intel's 8088, 8086, 80186, and 80286 CPUs. Application software written for these other 8- and 16-bit microprocessor families can be recompiled to run on the 80386 microprocessor. Some changes to the software such as adjustment of software timing loops and changing I/O address references may be required. The 386 microprocessor resides on the 32-bit wide CPU bus which interconnects the CPU with the math coprocessor and dual-port memory.

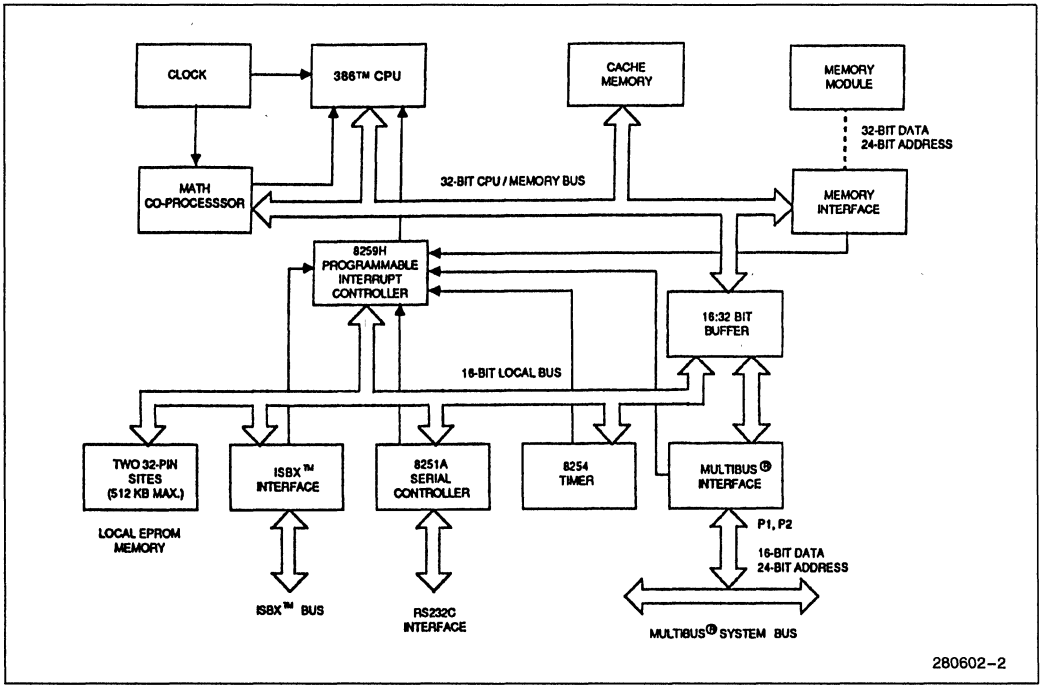


Figure 1. iSBC® 386/2x and 3x CPU Board Block Diagram

Instruction Set

The 386 CPU instruction set includes: variable length instruction format (including double operand instructions; 8-, 16-, and 32-bit signed and unsigned arithmetic operators for binary, BCD and unpacked ASCII data; and iterative word and byte string manipulation functions. All existing instructions have been extended to support 32-bit addresses and operands. New bit manipulation and other instructions have been added for extra flexibility in designing complex software.

Numeric Data Processor

For enhanced numerics processing compatibility, the iSBC 386/2x and 3x boards include an 80387 numeric coprocessor. Over 60 numeric instructions offer arithmetic, trigonometric, transcendental, logarithmic and exponential instructions. Supported data types include 16-, 32-, and 64-bit integer, 32- and 64-bit floating point, 18-digit packed BCD and 80-bit temporary. Data transfers to and from the CPU are 32-bits wide.

Architectural Features

The 8086, 8088, 80188, 80286, and 386 microprocessor family contain the same basic sets of registers, instructions, and addressing modes. The 80386 processor is upward compatible with the 8086, 8088, 80186, 80188, and 80286 CPU's.

Architectural Features

The 386 CPU operates in two modes: protected virtual address mode; and 8086 real address mode. In protected virtual address mode (also called protected mode), programs use virtual addresses. In this mode, the 386 CPU automatically translates logical addresses to physical addresses. This mode also provides memory protection to isolate the operating system and ensure privacy of each task's programs and data. In 8086 real address mode, programs use real addresses with up to one megabyte of address space. Both modes provide the same base instruction set and registers.

Interrupt Control

Incoming interrupts are handled by two cascaded on-board 8259A programmable interrupt controllers and by the 386's NMI line. Twenty interrupt sources are routed to the programmable controllers and the interrupt jumper matrix. Using this jumper matrix, the user can connect the desired interrupt sources to specific interrupt levels. The interrupt controllers prioritize interrupts originating from up to 15 sources and send them to the CPU. The user can connect a sixteenth interrupt to the 386 NMI line. Table 1 includes a list of devices and functions suported by interrupts. Bus vectored interrupts are not supported.

Table 1. Interrupt Request Sources

Source	Function	Number of Interrupts
MULTIBUS® Interface	Requests from MULTIBUS® resident peripherals or other CPU baords	8
8251A Serial Controller	Indicates status of transmit and receive buffers and RI lead of the RS232C interface	3
8254 Timers	Timer 0, 1 outputs; function determined by timer mode (hardwired to interrupt controller)	2
iSBX™ Connector	Function determined by iSBX™ MULTIMODULE™ board	4
Bus Timeout	Indicates addressed MULTIBUS® or iSBX™ resident device has not responded to a command within 10 ms	1
Power Fail Interrupt	Indicates AC power is not within tolerance (signal generated by system power supply)	1
Parity Interrupt	Indicates on-board parity error	1
Programmable Register	Generate interrupt under program control	1

Memory Capabilities

The iSBC 386/2x and 3x boards support both EPROM local memory and dynamic RAM (DRAM), which is located on-board. The DRAM is supported by a high speed on-board cache memory.

DRAM Memory

The iSBC 386/2x and 3x series CPU boards come with 1, 2, 4, or 8M bytes of DRAM memory. This memory is on a low profile module that is installed on the baseboard. The module measures approximately 4″ x 4″ and uses surface mount DRAM devices. The DRAM memory supports byte-parity error detection and has a 32-bit wide data path to the 80386 CPU and 16-bit wide data path to the MULTIBUS interface.

The memory may be expanded by installing an additional iSBC MM0x series memory module, which is available in 1, 2, 4, or 8M byte sizes. All mounting hardware is included. Maximum DRAM memory is 16M bytes using an iSBC 386/28 or 386/38 CPU board and an 8M byte iSBC MM08 memory module. This combination requires 1.8 inches of cardcage space.

Cache Memory

A 64K byte cache memory on the iSBC 386/2x and 3x boards supports the 386 CPU and provides 0 wait-state reads for data and program code resident in the cache memory. The cache memory is updated whenever data is written into the dual-port memory or when the CPU executes a read cycle and the data or program code is not present in cache memory. This process is controlled by the cache replacement algorithm. Cache "misses" require additional wait-states to retrieve data from the DRAM memory. If the processor is in pipelined mode, 2 wait-states (4 clock cycles) are required to retrieve data. If the processor is in non-pipelined mode, 3 wait-states are required. All writes to DRAM memory require 2 (pipelined) or 3 (non-pipelined) wait-states.

The cache memory supports 16K entries, with each entry comprised of a 32-bit data field and an 8-bit tag field. The tag field is used to determine which actual memory word currently resides in a cache entry. The cache memory size and effective replacement algorithm are designed to optimize both the probability of cache "hits" and local bus utilization.

EPROM Memory

The EPROM memory consists of two 32-pin JEDEC sites that are intended for boot-up and system diagnostic/monitor routines, application code, and ROM-able operating system software. Maximum local memory capacity is 512K bytes using Intel 27020 (256k x 8) 2 megabit EPROM devices. The EPROM memory resides at the upper end of the 386 device's memory space for both real address mode and PVAM operation.

Memory Map

In real address mode, the maximum amount of addressable physical memory is 1 Mbyte. In protected virtual address mode (PVAM), the maximum amount of addressable physical memory is 16 Mbytes. The system designer can easily change the CPU memory map to adapt the CPU board to the required overall system memory map. Reconfiguration is usually necessary for multiple processor-based systems with more than two CPU boards and/or intelligent I/O boards. By changing PAL devices and/or by moving jumpers, the designer can set:

- EPROM memory space
- Starting address of DRAM memory
- Amount of DRAM memory that is dual-ported to the CPU and MULTIBUS interface or single-ported to the CPU
- Access to off-board MULTIBUS address space

EPROM Memory

The EPROM memory space is set using four jumpers to accommodate 27256 (256 kb), 27512 (512 kb), 27010 (1 Mb), or 27020 (2 Mb) byte-wide devices. Smaller EPROM devices may be used, however the EPROM will appear more than once within the EPROM address space. Using a pair of 27020 EPROMs will provide 512k bytes of memory. The iSBC 386/2x and 3x series boards are designed to accommodate EPROM devices with access times ranging from 130 ns–320 ns. In real address mode, the ending address of EPROM memory is always 1M byte (FFFFFH). In PVAM, the ending address of EPROM memory is always 4G bytes (FFFF FFFFH), which is the top of the 386 CPU address space.

DRAM Memory Size/Location

The iSBC 386/2x and 3x boards allow the user to control the location and size of the DRAM memory (on the iSBC 386/2x and 3x boards) available for use by the CPU and other boards in the system. In PVAM, the starting address of DRAM can be set to start on any 1M byte boundary up through 15M bytes by setting jumpers and by installing a custom-programmed PAL device. In real address mode, the DRAM memory always starts at 0H (hex).

The ending address can be set on 64k byte boundaries using jumpers in both PVAM and real address mode. Setting the ending address at lower than the actual amount of installed memory effectively deselects a portion of DRAM and creates additional MULTIBUS address space.

MULTIBUS Address Space

Any address space not set aside as EPROM or DRAM memory automatically becomes address space the CPU can use to access other boards in the system. For example, Figure 2A shows a real address mode CPU memory map for a 1M byte iSBC 386/21 board. With the DRAM ending address set at 512k bytes and 128k bytes of installed EPROM, 384k bytes of MULTIBUS address space is accessable by the CPU. Figure 2B shows a typical PVAM configuration where the 4 Mbytes of DRAM has been set to start at 1M byte and end at 4.5M bytes. The address space from 0 to 1M byte and 4.5 to 16M bytes is the MULTIBUS address space accessable by the CPU.

Figure 2C illustrates another way the board can establish additional MULTIBUS address space. If the DRAM memory starts at 0, a jumper on the board can be used to create additional MULTIBUS address space between 512k bytes and 1M byte. This feature is available both in real address mode and PVAM.

Dual-Port/Local Memory

A portion or all of the DRAM memory can be selected to be dual-port (shared) memory. Both the starting and ending addresses are set on 256k byte boundaries using jumpers on the board. Any DRAM memory that is not configured as dual-port memory is local (single-port) memory available only to the CPU.

Programmable Timer

Three 16-bit, programmable interval timer/counters are provided using an 8254 device, with one timer dedicated to the serial port for use as a baud rate generator. The other two timers can be used to generate accurate time intervals under software control. The timers are not cascadable. Four timer/counter modes are available as listed in Table 2. Each counter is capable of operating in either BCD or binary modes. The contents of each counter may be read at any time during system operation.

Table 2. Programmable Timer Functions

Function	Operation
Interrupt on terminal count	When terminal count is reached, an interrupt request is generated. This function is extremely useful for generation of real-time clocks.
Rate generator	Divide by N counter. The output will go low for one input clock cycle, and the period from one low going pulse to the next is N times the input clock period.
Square-wave rate generator	Output will remain high until one-half the count has been completed, and go low for the other half of the count.
Software triggered strobe	Output remains high until software loads count (N). N counts after count is loaded, output goes low for one input clock period.

Serial I/O

The iSBC 386/2x and 3x boards include one RS232C serial channel, which is configured as an

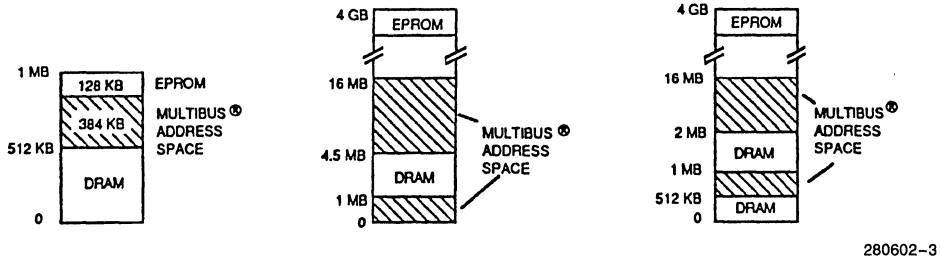


Figure 2A. Real Address Mode iSBC® 386/21 Board Memory Map

Figure 2B. PVAM iSBC® 386/24 Board Memory Map

Figure 2C. PVAM iSBC® 386/22 Board Memory Map

asynchronous, DTE interface. Data rates up to 19.2k baud may be selected. The serial channel can connect either to a host system for software development or to a stand alone terminal for field diagnostic support. For stand alone use, unhosted monitor software needs to be programmed by the user into the local EPROM memory. The serial channel may also be connected to a modem to provide remote diagnostic support or to download program codes. The physical interface is a 10-pin ribbon-style connector located on the front edge of the board.

iSBX™ Interface

For iSBX MULTIMODULE support, the iSBC 386/2x and 3x CPU boards provide an 8/16-bit iSBX connector that may be configured for use with either 8- or 16-bit, single or double-wide iSBX MULTIMODULE boards. Using the iSBX interface, a wide variety of specialized I/O functions can be added easily and inexpensively to the iSBC 386/2x and 3x boards.

Reset Functions

The iSBC 386/2x and 3x boards are designed to accept an Auxilliary Reset signal via the boards' P2 interface. In this way, system designs that require front panel reset switches are supported. The iSBC 386/2x and 3x boards use the AUX reset signal to reset all on-board logic (excluding DRAM refresh circuitry) and other boards in the MULTIBUS system. The iSBC 386/2x and 3x boards will also respond to an INIT reset signal generated by another board in the system.

LED Status Indicators

Mounted on the front edge of the iSBC 386/2x and 3x boards are four LED indicators that indicate the operating status of the board and system. One LED is used to show that an on-board parity error or a MULTIBUS bus parity error has occurred. A second LED indicates that a MULTIBUS or iSBX bus access timeout has occurred. The third LED is triggered by the start of an 386 bus cycle and will turn off if the 386 CPU stops executing bus cycles. The fourth LED will light under software control if the program writes to a specific I/O location.

MULTIBUS® SYSTEM ARCHITECTURE

Overview

The MULTIBUS system architecture includes three bus structures: the MULTIBUS system bus, the iLBX local bus extension and the iSBX MULTIMODULE expansion bus. Each bus structure is optimized to satisfy particular system requirements. The system bus provides a basis for general system design including memory and I/O expansion as well as multiprocessing support. The iLBX bus, which is usually used for memory expansion, is not supported by the iSBC 386/2x and 3x boards since all DRAM memory is located on-board. The iSBX bus povides a low cost way to add I/O to the board.

System Bus—IEEE 796

The MULTIBUS system bus is Intel's industry standard, IEEE 796, microcomputer bus structure. Both 8- and 16-bit single board computers are supported on the IEEE 796 structure with 24 address and 16 data lines. In its simplest application, the system bus allows expansion of functions already contained on a single board computer (e.g., memory and digital I/O). However, the IEEE 796 bus also allows very powerful distributed processing configurations using multiple processors, I/O boards, and peripheral boards. The MULTIBUS system bus is supported with a broad array of board level products, VLSI interface components, detailed published specifications and application notes.

System Bus—Expansion Capabilities

The user can easily expand or add features to his system by adding various MULTIBUS boards to his system. Products available from Intel and others include: video controllers; D/A and A/D converter boards; peripheral controller cards; communications/networking boards; voice synthesis and recognition boards; and EPROM memory expansion boards.

System Bus—Multimaster Capabilities

For those applications requiring additional processing capacity and the benefits of multiprocessing (i.e., several CPUs and/or controllers sharing system tasks through communication over the system bus), the iSBC 386/2x and 3x boards provide full system bus arbitration control logic. This control logic allows up to four bus masters to share the system bus using a serial (daisy chain) priority scheme. By using an external parallel priority decoder, this may be extended to 16 bus masters. In addition to multiprocessing, the multimaster capability also provides a very efficient mechanism for all forms of DMA (Direct Memory Access) transfers.

iSBX™ Bus MULTIMODULE™ On-Board Expansion

Through this interface, additional on-board I/O functions may be added, such as parallel and serial I/O, analog I/O, small mass storage device controllers (e.g., floppy disks), BITBUS Control, and other custom interfaces to meet specific needs. Compared to other alternatives such as MULTIBUS I boards, iSBX modules need less interface logic and power, and offer simpler packaging and lower cost. The iSBX interface connector on the iSBC 386/2x and 3x boards provides all the signals necessary to interface to the local on-board bus, and is compatible with both 8-bit and 16-bit MULTIMODULES.

SOFTWARE SUPPORT

Operating Systems

The iSBC 386/2x and 3x boards are supported by a variety of operating systems, including the iRMX I, iRMX II, XENIX (from Intel) and System V/386 operating systems (third party vendors).

The iRMX II operating system is a realtime multitasking and multi-programming software system capable of executing all the configurable layers of the iRMX II operating system on the 386 microprocessor and the iSBC 386/2x and 3x single board computers. Up to 16 MB of physical system memory is supported. The iRMX II Operating System also allows the user to take advantage of the hardware traps built into the 386 processor that provide expanded debug capabilities and increased code reliability.

The iRMX II operating system is designed to support time-critical applications requiring real time performance in the industrial automation, financial; medical, communications, and data acquisition and control (including simulation) marketplaces.

Application code written under the iRMX I operating system can also run on the iSBC 386/2x and 3x boards. The code may either be run directly on the iRMX I operating system, or may be recompiled using Intel's 286 compilers and then run under iRMX II software. Application code will require only minor changes and may then take advantage of the added memory addressability, code reliability, and debug capability of the iRMX II operating system.

Languages and Tools

A wide variety of languages is available for the iRMX, XENIX and System V/386 operating systems. For the iRMX II operating system, Intel offers ASM 286, PASCAL 286, PL/M 286, C 286, and FORTRAN 286. For the XENIX operating system Intel offers ASM 386, PL/M 386, C 386, and PASCAL 386. For the System V/386 Operating System several different software vendors provide selections of languages, including ASM, C, PASCAL, FORTRAN, COBOL, RPG, PL1, BASIC, and Artificial Intelligence programming languages LISP and Arity/Prolog. Software development tools include PSCOPE Monitor 386 (PMON 386 and DMON 386), Softscope (for iRMX II), and an ICE 386 in-circuit-emulator.

BOARD SPECIFICATIONS

Word Size

Instruction—8, 16, 24, 32 or 40 bits
Data—8, 16, 32 bits

System Clock

386 CPU—16 MHz or 20 MHz
Numeric Processor—80387 module—16 MHz or 20 MHz

DRAM Memory

On-board parity memory
- iSBC 386/21/31 board—1M byte
- iSBC 386/22/32 board—2M bytes
- iSBC 386/24/34 board—4M bytes
- iSBC 386/28/38 board—8M bytes

Memory expansion—One additonal plug-in module:
- iSBC MM01—1M byte
- iSBC MM02—2M bytes
- iSBC MM04—4M bytes
- iSBC MM08—8M bytes

Maximum Addressable Physical Memory—16 Megabytes (protected virtual address mode) 1 Megabyte (real address mode)

EPROM Memory

Number of sockets—Two 32-pin JEDEC Sites (compatible with 28-pin and 32-pin devices)

Sizes accommodated—64 kb (8k x 8), 128 kb (16k x 8), 256 kb (32k x 8), 512 kb (64k x 8), 1 Mb (128k x 8), 2 Mb (256k x 8)

Device access speeds—130 ns to 320 ns
Maximum memory—512k bytes with 27020 (2M bit) EPROMs

I/O Capability

Serial Channel

Type—One RS232C DTE asynchronous channel using an 8251A device

Data Characteristics—5-8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection; automatic break detect and handling; even/odd parity error generation and detection

Speed—110, 150, 300, 600, 1.2 kb, 2.4 kb, 4.8 kb, 9.6 kb, 19.2 kb

Leads supported—TD, RD, RTS, CTS DSR, RI, CD, SG

Connector Type—10 pin ribbon

Expansion—One 8/16-bit iSBX interface connector for single or double wide iSBX MULTIMODULE board.

Interrupt Capacity

Potential Interrupt Sources—21 (2 fixed, 19 jumper selectable)

Interrupt Levels—16 using two 8259A devices and the 80386 NMI line

Timers

Two programmable timers using one 8274 device.

Output Frequencies/Timing Intervals

Function	Single Counter	
	Min	Max
Real-time interrupt	1.63 μs	53.3 ms
Rate Generator	18.8 Hz	615 kHz
Square-wave rate generator	18.8 Hz	615 kHz
Software triggered strobe	1.63 μs	53.3 ms

Interfaces

MULTIBUS Bus—All signals TTL compatible
iSBX Bus—All signals TTL compatible
Serial I/O—RS 232C, DTE

MULTIBUS® DRIVERS

Function	Type	Sink Current (mA)
Data	Tri-State	64
Address	Tri-State	24
Commands	Tri-State	32
Bus Control	Open Collector	16/32

Power Requirements

iSBC 386/2x and 3x boards
Maximum: +5V, 12.5A
±12V, 35 mA
Typical: +5V, 9A
±12V, 20 mA

NOTE:
Does not include power for iSBX module, EPROM memory, or added iSBCMM0x memory modules.

Add the following power when adding iSBC MM0X memory modules:

iSBC MM01 +5V, 0.71A
MM02 +5V, 0.96A
MM04 +5V, 0.71A
MM08 +5V, 0.96A

Environmental Requirements

Operating Temperature—0°C to 60°C at 300 LFM
Relative Humidity—0% to 85% noncondensing
Storage Temperature—−40°C to +70°C

Physical Characteristics

Dimensions
Width—12.00 in. (30.48 cm)
Height—7.05 in. (17.91 cm)
Depth—0.86 in. (2.18 cm), 1.62 in. (4.11 cm) with added memory module

Recommended Minimum Cardcage Slot Spacing
1.2 in. (3.0 cm), with or without iSBX MULTIMODULE
1.8 in. (4.6 cm), with addded iSBC MM0x memory module

Approximate Weight
26 oz. (738 gm)

Reference Manual

149094—iSBC 386/21/22/24/28 Hardware Reference Manual (order separately)

453652—iSBC 386/31/32/34/38 Single Board Computer Hardware Reference Manual

Ordering Information

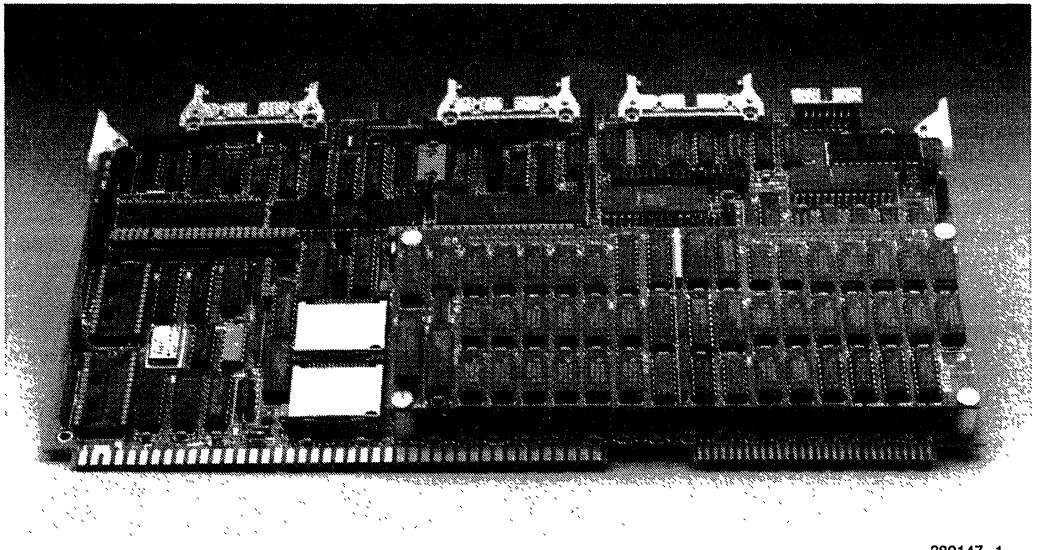
Part Number	Description
CPU Boards	
SBC38621	16 MHz 386 MULTIBUS I CPU Board with 1 MB DRAM Memory
SBC38622	16 MHz 386 MULTIBUS I CPU Board with 2 MB DRAM Memory
SBC38624	16 MHz 386 MULTIBUS I CPU Board with 4 MB DRAM Memory
SBC38628	16 MHz 386 MULTIBUS I CPU Board with 8 MB DRAM Memory
SBC38631	20 MHz 386 MULTIBUS I CPU Board with 1 MB DRAM Memory
SBC38632	20 MHz 386 MULTIBUS I CPU Board with 2 MB DRAM Memory
SBC38634	20 MHz 386 MULTIBUS I CPU Board with 4 MB DRAM Memory
SBC38638	20 MHz 386 MULTIBUS I CPU Board with 8 MB DRAM Memory
Memory Modules	
SBCMM01	1 MB Parity DRAM Memory Expansion Module
SBCMM02	2 MB Parity DRAM Memory Expansion Module
SBCMM04	4 MB Parity DRAM Memory Expansion Module
SBCMM08	8 MB Parity DRAM Memory Expansion Module

iSBC® 286/12, 286/14, 286/16* SINGLE BOARD COMPUTERS

- 8 MHz 80286 Microprocessor
- Two JEDEC 28-Pin Sites for up to 128K Bytes of Local EPROM Memory, Expandable to 256K Bytes Using an iSBC® 341 Expansion Module
- 1, 2, or 4 Megabyte, 0 Wait-State, Dual-Port, Parity Memory
- Supports User Installed 80287 Numeric Data Processor and 82258 Advanced DMA Controller Devices
- Two iSBX™ Bus Interface Connectors for I/O Expansion
- Synchronous High-Speed Interface for 0 Wait-State Read/Write to EX Memory Expansion Boards
- iLBX™ Interface for iLBX Memory Board Expansion
- 16 Levels of Vectored Interrupt Control
- Centronics-Compatible Parallel I/O Printer Interface
- Two Programmable Multiprotocol Synchronous/Asynchronous Serial Interfaces; One RS232C, the Other RS232C or RS422/449 Compatible

The iSBC 286/12, iSBC 286/14, and iSBC 286/16 Single Board Computers are members of Intel's family of 16-bit microcomputers. The boards feature an 80286 microprocessor running at 8 MHz together with 1, 2, or 4 megabytes of dual-ported, 0 wait-state, parity memory. These features make the iSBC 286/12/14/16 boards the ideal single board solution for applications requiring high performance and up to 1, 2, or 4 megabytes of memory. For those applications needing more memory, up to four memory expansion boards may be connected to the iSBC 286/12/14/16 boards over its P2 interface. The P2 interface supports both standard iLBX memory boards and Intel's EX series of synchronous, 0 wait-state, memory boards that provide up to 16 megabytes of system memory. The iSBC 286/12/14/16 boards also feature two sockets for user installed 80287 Numeric Data Processor and 82258 Advanced Direct Memory Access Controller devices. These components further increase board performance by off-loading time intensive tasks from the 80286 microprocessor. The iSBC 286/12/14/16 CPU boards are true single-board solutions that also include two serial I/O channels, one parallel line printer channel, local memory, interrupt controllers and programmable timers all on one board.

*The iSBC® 286/12; iSBC 286/14 and iSBC 286/16 are also manufactured under product code piSBC 286/12; piSBC 286/14 and piSBC 286/16 by Intel Puerto Rico, Inc.



280147-1

*XENIX is a registered trademark of Microsoft Corp.
**UNIX is a trademark of Bell Laboratories.

October 1989
Order Number: 280147-003

FUNCTIONAL DESCRIPTION

Overview

The iSBC 286/12/14/16 boards utilizes the 80286 CPU within the MULTIBUS® system architecture, enhanced by the industry standard iLBX bus and a new, 0 wait-state, synchronous memory interface, to provide a high-performance 16-bit solution. This board features 1, 2, or 4 megabytes of dual-port, 0 wait-state, parity memory, plus interrupt, memory and I/O features facilitating a complete single-board computer system. The iSBC 286/12/14/16 boards can be used in many applications originally designed for Intel's other 16-bit microcomputers.

Central Processing Unit

The central processor for the iSBC 286/12/14/16 board is the 80286 CPU operating at an 8.0 MHz clock rate. The 80286 CPU is upwardly compatible with Intel's 8088 and 8086 CPUs. The 80286 CPU runs 8088 and 8086 code at substantially higher speeds due to its parallel architecture. In addition, the 80286 CPU provides on-chip memory management and protection and virtual memory addressing of up to 1 gigabyte per task. Processing speed and efficiency may be further enhanced by installing an 80287 numerics coprocessor and an 82258 ADMA controller. The clock rates for the 80286 and the 80287 are independent with the 80287 rate jumper selectable at either 5.3 MHz or 8.0 MHz.

Instruction Set

The 80286 instruction repertoire includes variable length instruction format (including double operand instructions), 8-bit and 16-bit signed and unsigned arithmetic operators for binary, BCD and unpacked ASCII data, and iterative word and byte string manipulation functions.

Numeric Data Processor

For enhanced numerics processing capability, the 80287 Numeric Data Processor extends the 80286 architecture and data set. Over 60 numeric instructions offer arithmetic, trigonometric, transcendental, logarithmic and exponential instructions. Supported data types include 16-, 32-, and 64-bit integer, 32- and 64-bit floating point, 18-digit packed BCD and 80-bit temporary.

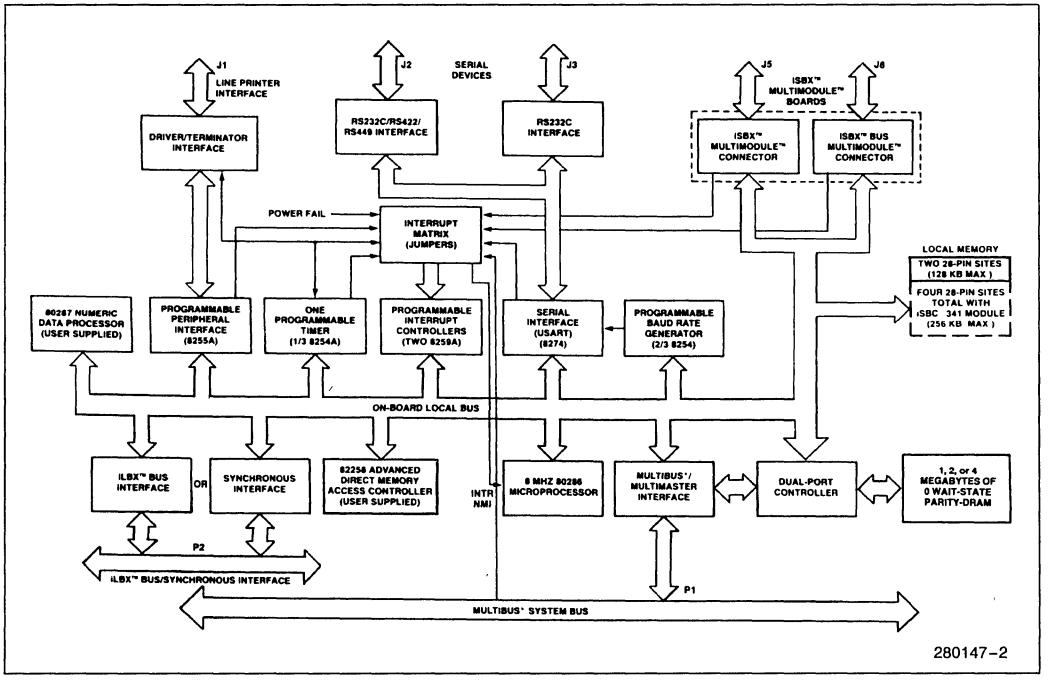


Figure 1. iSBC® 286/12 Block Diagram

Advanced DMA Controller

For those applications that require frequent moving of large blocks of data, the user may install an Intel 82258, 4 channel, advanced DMA (ADMA) controller to further increase system performance. The ADMA Controller supports DMA requests from the 8274 USART (2 channels) and the iSBX interfaces on the board (1 per interface). The ADMA can also perform data transfers over the on-board CPU bus, the MULTIBUS (P1) interface, and the iLBX/synchronous (P2) interface.

ARCHITECTURAL FEATURES

The 8086, 8088, 80186 and 80286 microprocessor family contains the same basic set of registers, instructions, and addressing modes. The 80286 processor is upward compatible with the 8086, 8088, and 80186 CPUs.

The 80286 operates in two modes: 8086 real address mode, and protected virtual address mode. In 8086 real address mode, programs use real address with up to one megabyte of address space. Programs use virtual addresses in protected virtual address mode, also called protected mode. In protected mode, the 80286 CPU automatically maps 1 gigabyte of virtual addresses per task into a 16 megabyte real address space. This mode also provides memory protection to isolate the operating system and ensure privacy of each task's programs and data. Both modes provide the same base instruction set and registers.

Vectored Interrupt Control

Incoming interrupts are handled by two on-board 8259A programmable interrupt controllers (PIC) and by the 80286's NMI line. Interrupts originating from up to 15 sources are prioritized and then sent to the CPU. The 8259A devices support both polled and vectored mode of operation. Further interrupt capability is available through bus vectored interrupts where slave 8259A interrupt controllers resident on separate iSBC Boards supply an interrupt vector to the on-board CPU.

Interrupt Sources

Twenty-six potential interrupt sources are routed to the slave PIC device and to the interrupt jumper matrix where the user can connect the desired interrupt sources to specific interrupt levels.

Memory Capabilities

DUAL-PORT MEMORY

The iSBC 286/12/14/16 boards feature 1, 2, or 4 megabytes of 0 wait-state, parity memory installed on the board. This memory, which is implemented using 256 Kb or 1 Mb DRAMs installed on a daughter board, is dual-ported to the on-board CPU bus and the MULTIBUS (P1) interface. For those applications requiring more memory, the iSBC 286/12/14/16 boards also feature an iLBX and synchronous memory interface to increase physical memory capacity to 16 megabytes.

The iSBC 322/324 memory upgrade modules allow for expansion of on-board dual-port DRAM. The iSBC 322 upgrades an iSBC 286/12 to 2 MBytes. The iSBC 324 upgrades an iSBC 286/12 or iSBC 286/14 to 4 MBytes.

LOCAL MEMORY

Two, 28-pin sites are provided for installing up to 128 KB of EPROM firmware.

By installing an iSBC 341 EPROM expansion module, local memory can be increased to four sites to support up to 256 KB of EPROM. Local memory access time is selectable at one, two, or three wait-states and is a function of the device speed.

Serial I/O

A two-channel serial communications interface using Intel's 8274 Multi-Protocol Serial Controller (MPSC) is contained on the iSBC 286/12/14/16 boards. Two independent software selectable baud rate generators provide the MPSC with all common communication frequencies. The protocol (i.e. asynchronous, bisync, or SDLC/HDLC), data format, control character format, parity and baud rate are all under program control. Software interfacing to the MPSC can be via either a polled or interrupt driven routine. Channel A may be configured for an RS232C or RS422/RS449 interface; channel B is set for RS232C operation only. DMA operation for channel A is available if the optional 82258 (ADMA) is installed.

Programmable Timers

The iSBC 286/12/14/16 boards provide three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8254 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate time intervals under software control. Routing for the outputs of these counters is jumper selectable. The outputs may be independently routed to the 8259A Programmable Interrupt Controller or to the 8274 MPSC to count external events or provide baud rate generation. The third interval timer in the 8254 is dedicated to providing a clock for the programmable baud rate generator in the iSBC 286/12/14/16 boards' MPSC serial controller.

Line Printer Interface/Board ID

An 8255A Programmable Peripheral Interface (PPI) provides a Centronics compatible line printer interface, several on-board functions, and four non-dedicated input bits. Drivers are provided for a complete Centronics compatible line printer interface.

Software Reset

The software reset feature allows the 80286 microprocessor to return to Real Address mode operation from PVAM under software control. The system reset line (INIT*) and the dual-port memory are not affected, and all I/O context is preserved. The software reset is activated by a byte write to I/O location 00E0H. To distinguish the software reset from a true system initialization reset, a flag is provided. Another flag is provided that indicates whether the iSBC 286/12/14/16 board hardware (not the 80286 device) is currently configured for PVAM or Real Address Mode.

Front Panel Connector

A 14-pin connector (J4) is mounted on the top edge of the board and is designed to connect to the front panel and power supply of the system enclosure. Leads supported include Reset and Interrupt input lines from (conditioned) front panel switches, a Run signal to drive a front panel LED, a Power Fail Interrupt line that connects to the power supply, and extra power and ground leads to support miscellaneous front panel circuitry.

System Bus—Expansion Capabilities

Memory and I/O capacity may be expanded and additional functions added using Intel MUTLIBUS compatible expansion boards. Memory may be expanded by adding user specified combinations of RAM boards, EPROM boards, bubble memory boards, or combination boards. Input/output capacity may be added with digital I/O and analog I/O expansion boards. Mass storage capability may be achieved by adding single or double density diskette controllers, or hard disk controllers. Modular expandable backplanes and cardcages are available to support multiboard systems.

Multimaster Capabilities

The iSBC 286/12/14/16 boards provide full system bus arbitration control logic. This control logic allows up to three iSBC 286/12/14/16 boards or other bus masters to share the system bus using a serial (daisy chain) priority scheme and allows up to 16 masters to share the MULTIBUS system bus with an external parallel priority decoder.

Memory Map

The memory map of the iSBC 286/12/14/16 board is shown in Figure 1. The memory maps for the iSBC 286/14 and iSBC 286/16 boards are similar, except the total amount of on-board DRAM memory is 2 or 4 MB, and the dual-port memory space is larger. The memory map, which shows the default configuration of the board, may be easily changed by the user to meet the needs of almost any system design. As a result, the iSBC 286/12/14/16 boards are particularly suited for complex multiple processor and/or multiple intelligent I/O board-based systems.

The memory map can be changed by moving on-board jumpers or by installing user-programmed PALs (programmable array logic devices).

Using only the jumpers on the iSBC 286/12/14/16 board, the MULTIBUS window size can be set at 0 (no window), 64 KB, 128 KB, 256 KB, or 1 MB in real address mode. The MULTIBUS window is normally not available in PVAM, however, a PAL may be programmed to provide this feature. Jumpers are also used to set aside a portion of the dual-port memory so that it may only be accessed by the CPU (single-ported memory). Block sizes of 64 KB, 128 KB, 256 KB, 512 KB or 1 MB may be selected. Finally, jumpers are used to select any of 6 EPROM memory

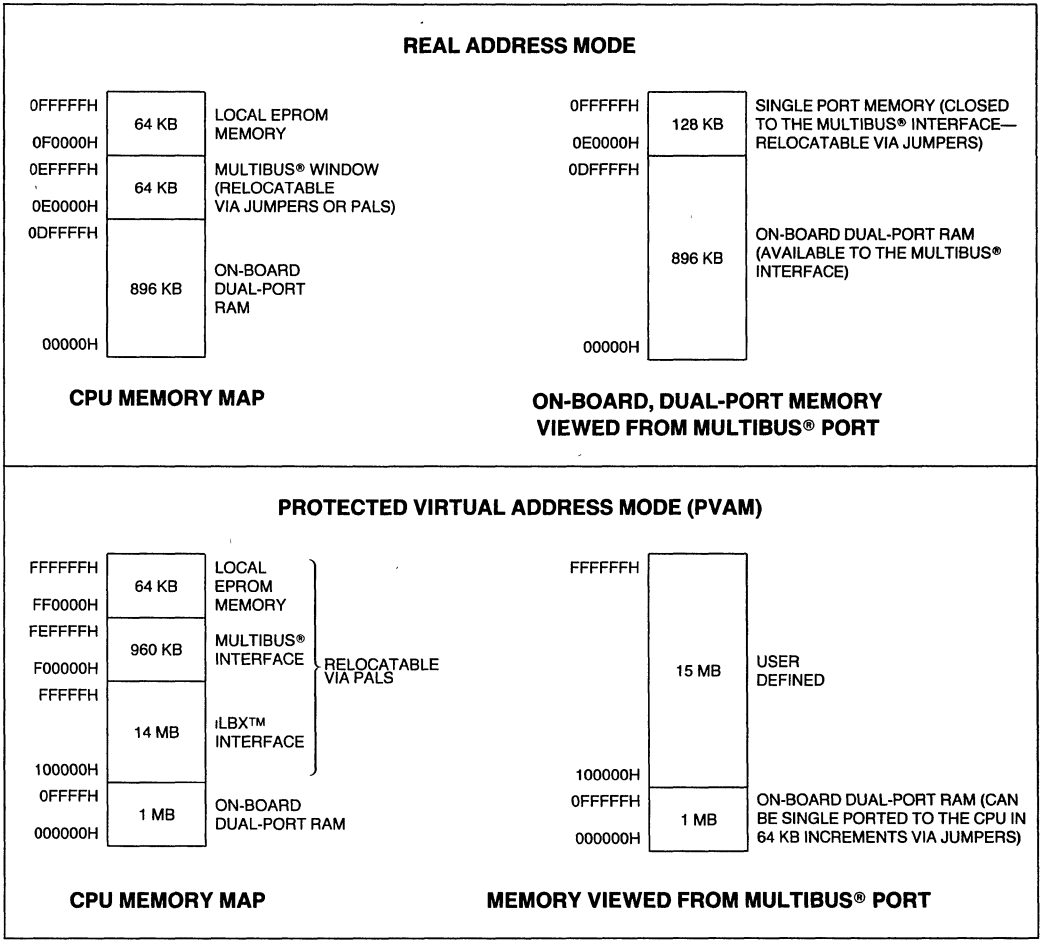


Figure 1. Memory Map for iSBC® 286/12 Board (Default Configuration)

sizes ranging from 4 KB (using 2716 devices) up to 256 KB (using 27512 devices and an iSBC 341 module).

If the user needs to alter the memory map further, five PALs on the baseboard are socketed and may be replaced by custom designed devices. Using programmed PALs, the designer can:

— Set the base DRAM memory starting address (as viewed by the 80286 microprocessor) at 0 (default configuration) or to any ½ megabyte boundary up through 16 MB (0 or 512 KB in real address mode).

— Set the base DRAM memory starting address (as viewed by other boards over the MULTIBUS interface) at 0 (default configuration) or to any megabyte boundary up through 16 MB (fixed at 0 in real address mode).

— Set single or multiple MULTIBUS windows as small as 64 KB or as large as 1 MB within the first megabyte of address space. MULTIBUS windowing can be enabled both in real address mode and PVAM. The window size can also be set at 0 (no window) so that the CPU can only access its own DRAM memory.

The jumper and PAL changes may be used in combination with each other. For example, jumpers can be installed to set EPROM address space and to exclusively allocate (single-port) a portion of the dual-port memory to the CPU. Then, PALs can be installed to establish two MULTIBUS windows of different sizes and to set the DRAM base starting addresses.

High Speed Off-Board Memory

The iSBC 286/12/14/16 boards can access off-board memory either over the MULTIBUS (P1) interface, or over the P2 interface. Memory transfers over the P2 interface are faster because the CPU board doesn't have to arbitrate for access to the MULTIBUS interface.

Using the P2 interface, the iSBC 286/12/14/16 boards can be configured to operate with either a standard iLBX interface or with the high-speed synchronous interface.

The iSBC 286/12/14/16 boards as supplied are configured to operate with a synchronous, P2 interface. This high-performance interface is designed to connect to Intel's EX series of memory expansion boards to yield a CPU to memory read/write time of 0 wait-states. The EX memory expansion boards are available in sizes ranging from 512K bytes up to 4M bytes.

A total of four memory boards can be placed on the iLBX or synchronous interface bus. With 4M byte memory boards, this results in a total of 16M bytes on the memory expansion bus.

iSBX™ Bus MULTIMODULE™ On-Board Expansion

Two 8-, 16-bit iSBX MULTIMODULE connectors are provided on the iSBC 286/12/14/16 boards. The iSBX interface connectors on the iSBC 286/12/14/16 boards provide all signals necessary to interface to the local on-board bus, including 16 data lines for maximum data transfer rates. The iSBX MULTIMODULE Boards designed with 8-bit data paths and using the 8-bit iSBX connector are also supported on the iSBC 286/12/14/16 microcomputer boards. A broad range of iSBX MULTIMODULE Board options are available from Intel. Custom iSBX modules may also be designed.

SPECIFICATIONS

Word Size

Instruction—8, 16, 24, 32 or 40 bits

Data—8 or 16 bits

System Clock

CPU—8.0 MHz

Numeric Processor—5.3 MHz or 8.0 MHz (Jumper Selectable)

Cycle Time

Basic Instruction—8.0 MHz - 250 ns (assumes instruction in queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e. two clock cycles)

Dual-Port Memory

1, 2, or 4 megabyte, 0 wait-state, parity DRAM dual-ported to the on-board CPU bus and the MULTIBUS interface.

Local Memory

Number of sockets—two 28-pin JEDEC sites, expandable to 4 sites using iSBC 341 JEDEC Expansion Module

Maximum Size—128 KB expandable to 256 KB by installing an iSBC 341 EPROM Expansion Module. Memory size is set by jumpers on the iSBC 286/12/14/16 board.

Compatible Devices—EPROM, up to 64K x 8 (Intel 27512)

Interrupt Capacity

26 interrupt sources (total); 5 hard-wired to the 8259A PIC; 21 jumper selectable

Interrupt Levels—16 vectored requests using two 8259A devices and the 80286 microprocessor's NMI line

I/O Capability

Parallel — Line printer interface, on-board functions, and 3-bit board installed options code

Serial — Two programmable channels using one 8274 device

Timers — Three programmable timers using one 8254 device

Expansion— Two 8/16-bit iSBX MULTIMODULE connectors

Interfaces

MULTIBUS Bus—All signals TTL compatible

iSBX Bus—All signals TTL compatible

iLBX Bus—All signals TTL compatible

Synchronous Interface—All signals TTL compatible

Serial I/O— Channel A: RS232C/RS422/RS449 compatible, DCE or DTE

Channel B: RS232C compatible, DCE

Timer—All signals TTL compatible

Interrupt Requests—All TTL compatible

Serial Communications Characteristics

Synchronous—5–8 bit characters; internal or HDLC/SDLC character synchronization; automatic sync insertion; even or odd parity

Asynchronous—5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection; even or odd parity

Physical Characteristics

Width: 12.00 in. (30.48 cm)

Height: 7.05 in. (18.00 cm)

Depth: 0.88 in. (2.24 cm)
1.16 in. (2.95 cm) with iSBX MULTIMODULE board installed

Recommended Slot spacing (without iSBX MULTIMODULE): 1.2 in. (3.0 cm)

Weight: 26 oz. (731 gm)

Electrical Characteristics

DC Power Requirements:

Maximum: +5V, 8.7A; ±12V, 35 mA (for serial I/O)

Typical: +5V, 5.7A; ±12V, 20 mA

NOTE:
Power requirements are for the default configuration. Does not include power for optional EPROM, 80287 or 82258 devices, or installed iSBX MULTIMODULE boards.

Environmental Characteristics

Operating Temperature: 0°C to 60°C with 8 CFM airflow across board (default configuration)

Relative Humidity: to 90% (without condensation)

Reference Manual

147533— iSBC 286/12/14/16 Hardware Reference Manual (order separately)

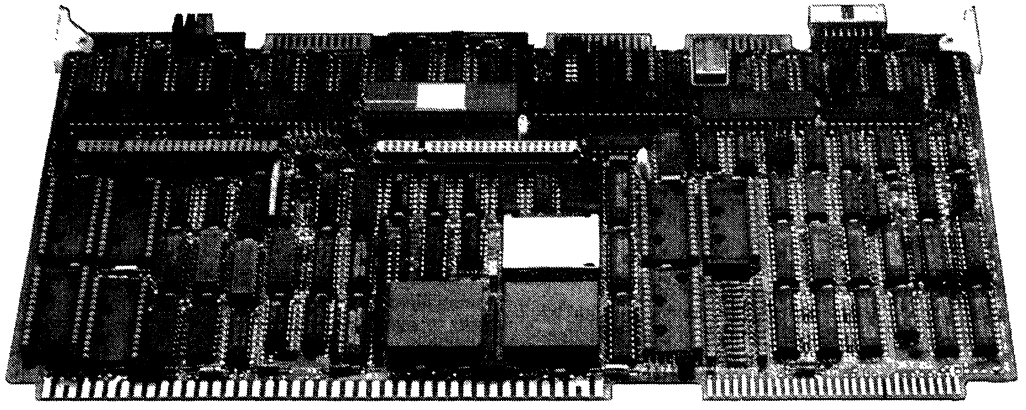
ORDERING INFORMATION

Part Number	Description
SBC 286/12	Single Board Computer with 1 MB of Memory
SBC 286/14	Single Board Computer with 2 MB of Memory
SBC 286/16	Single Board Computer with 4 MB of Memory
SBC 322	2 MB DRAM Module
SBC 324	4 MB DRAM Module

iSBC® 286/10A*
SINGLE BOARD COMPUTER

- 8 MHz 80286 Microprocessor
- Supports User Installed 80287 Numeric Data Processor
- iLBX™ Interface for iLBX Memory Board Expansion
- 0 Wait-State Synchronous Interface to EX Memory Expansion Boards
- Eight JEDEC 28-Pin Sites for Optional SRAM/iRAM/EPROM/E²PROM Components
- Optional Expansion to Sixteen JEDEC 28-Pin Sites with Two iSBE® 341 Boards
- Maximum On-Board Memory Capacity 384 KB
- Two iSBX™ Bus Interface Connectors for I/O Expansion
- 16 Levels of Vectored Interrupt Control
- Centronics-Compatible Parallel I/O Printer Interface
- Two Programmable Multiprotocol Synchronous/Asynchronous Serial Interfaces; One RS232C, the Other RS232C or RS422/449 Compatible

The iSBC® 286/10A Single Board Computer is a member of Intel's complete line of microcomputer modules and systems which take advantage of Intel's VLSI technology to provide economical, off-the-shelf, computer-based solutions for OEM applications. The CPU, system clock, memory sockets, I/O ports and drivers, serial communications interface, priority interrupt logic and programmable timers all reside on the board. The iSBC 286/10A supports both the standard iLBX interface and the synchronous high speed interface (EX-series memory boards).



280079-1

*The iSBC® 286/10A is also manufactured under product code piSBC® 286/10A and siSBC® 286/10A by Intel Puerto Rico, Inc. and Intel Singapore, Ltd.

September 1989
Order Number: 280079-006

FUNCTIONAL DESCRIPTION

Overview

The iSBC 286/10A board utilizes the 80286 CPU within the MULTIBUS® system architecture, enhanced by the industry standard iLBX bus and a new, 0 wait-state, synchronous memory interface, to provide a high performance 16-bit solution. This board also includes on-board interrupt, memory and I/O features facilitating a complete signal board computer system.

Central Processing Unit

The central processor for the iSBC 286/10A board is the 80286 CPU operating at a 8.0 MHz clock rate. The 80286 CPU runs 8088 and 86 code at substantially higher speeds due to it's parallel chip architecture. In addition, the 80286 CPU provides on chip memory management and protection and virtual memory addressing of up to 1 gigabyte per task. Numeric processing power may be enhanced with the user installed 80287 numerics processor. The clock rates for the 80286 and the 80287 are independent with the 80287 rate jumper selectable at either 5.3 or 8.0 MHz.

Instruction Set

The 80286 instruction repertoire includes variable length instruction format (including double operand instructions), 8-bit and 16-bit signed and unsigned arithmetic operators for binary, BCD and unpacked ASCII data, and iterative word and byte string manipulation functions.

For enhanced numerics processing capability, the 80287 Numeric Data Processor extends the 80286 architecture and data set. Over 60 numeric instructions offer arithmetic, trigonometric, transcendental, logarithmic and exponential instructions. Supported data types include 16-, 32-, and 64-bit integer, 32- and 64-bit floating point, 18-digit packed BCD and 80-bit temporary.

Architectural Features

The 8086, 8088, 80186 and the 80286 microprocessor family contains the same basic set of registers, instructions, and addressing modes. The 80286 processor is upward compatible with the 8086, 8088, and 80186 CPUs.

The 80286 operates in two modes: 8086 real address mode, and protected virtual address mode. In 8086 real address mode, programs use real address with up to one megabyte of address space. Programs use virtual addresses in protected virtual address mode, also called protected mode. In protected mode, the 80286 CPU automatically maps 1 gigabyte of virtual addresses per task into a 16 megabyte real address space. This mode also provides memory protection to isolate the operating system and ensure privacy of each task's programs and data. Both modes provide the same base instruction set, registers, and addressing modes.

VECTORED INTERRUPT CONTROL

Incoming interrupts are handled by two on-board 8259A programmable interrupt controllers and by the 80286's NMI line. Interrupts originating from up to 16 sources are prioritized and then sent to the CPU as a vector address. Further interrupt capability is available through bus vectored interrupts where slave 8259A interrupt controllers are resident on separate iSBC boards and are then cascaded into the on-board interrupt control.

INTERRUPT SOURCES

Twenty-three potential interrupt sources are routed to the interrupt jumper matrix where the user can connect the desired interrupt sources to specific interrupt levels.

MEMORY CAPABILITIES

There are a total of eight 28-pin JEDEC sites on board. Four sites are for local memory and can contain up to 256K bytes of EPROM devices. The four other sites are known as the dual-port memory and may be addressed by the MULTIBUS interface and the on-board CPU bus. Up to 128K bytes of either iRAM, SRAM, EPROM, or E^2PROM can reside in these sites. Both the local and dual-port memory can be expanded to eight sites each by using two iSBC 341 JEDEC expansion modules. In this way, smaller size memory devices can be used up to the 256KB (local) and 128KB (dual-port) memory capacities.

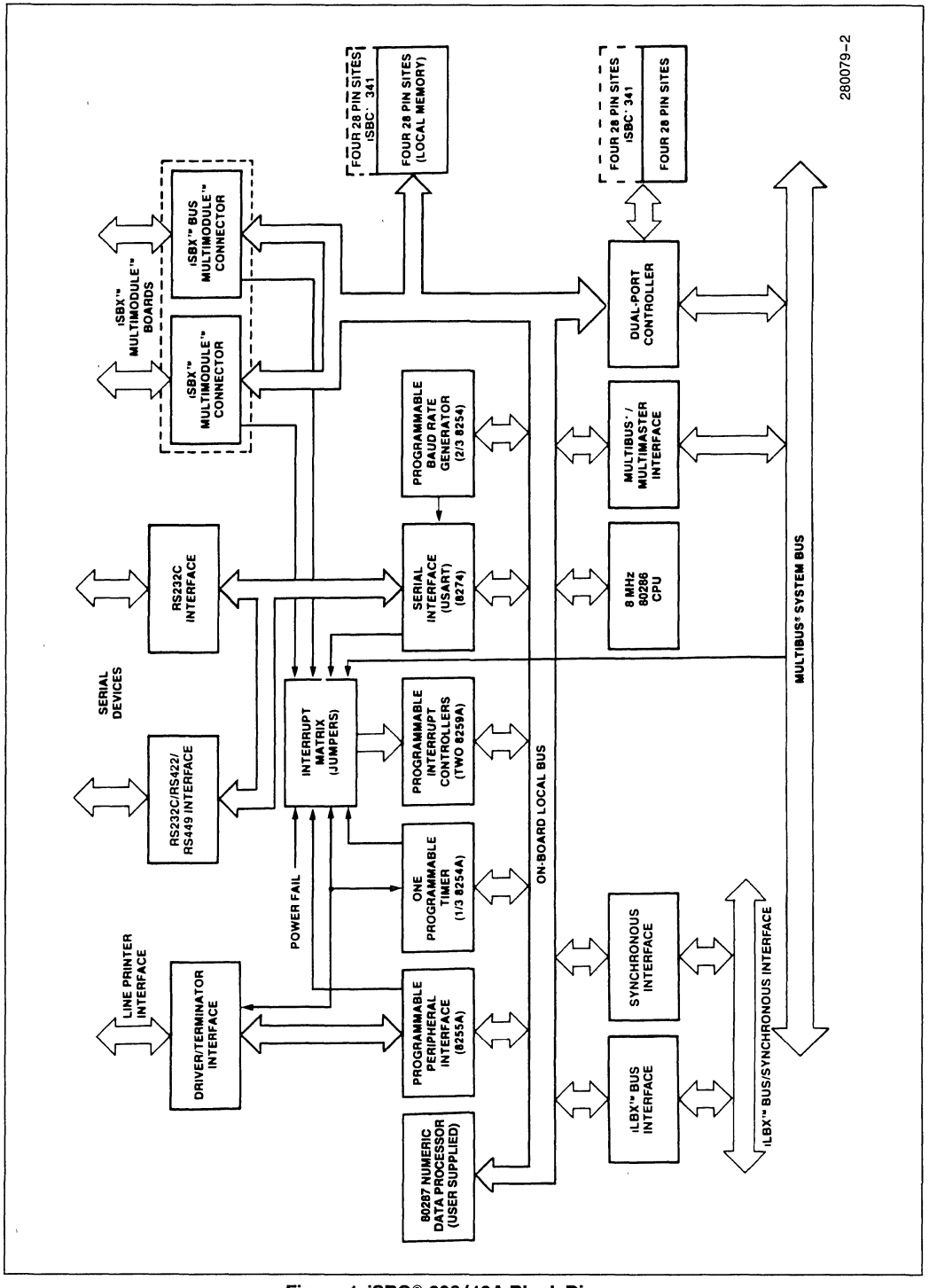


Figure 1. iSBC® 286/10A Block Diagram

SERIAL I/O

A two channel serial communications interface using Intel's 8274 Multi-Protocol Serial Controller (MPSC) is contained on the iSBC 286/10 board. Two independent software selectable baud rate generators provide the MPSC with all common communication frequencies. The protocol (i.e., asynchronous, IBM* bisync, or SDLC/HDLC), data format, control character format, parity and baud rate are all under program control. Software interfacing to the MPSC can be via either a polled or interrupt driven routine. One channel may be configured for an RS232C or RS422/RS449 interface with the other channel RS232C only. .

PROGRAMMABLE TIMERS

The iSBC 286/10A board provides three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8254 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate time intervals under software control. Routing for the outputs of these counters is jumper selectable. The outputs may be independently routed to the 8259A Programmable Interrupt Controller or to the 8274 MPSC to count external events or provide baud rate generation. The third interval timer in the 8254 is dedicated to providing a clock for the programmable baud rate generator in the iSBC 286/10A board's MPSC serial controller.

LINE PRINTER INTERFACE

An 8255A Programmable Peripheral Interface (PPI) provides a line printer interface, several on-board functions, and four non-dedicated input bits. Drivers are provided for a complete Centronics compatible line printer interface.

MULTIMASTER CAPABILITIES

The iSBC 286/10A board provides full system bus arbitration control logic. This control logic allows up to three iSBC 286/10A boards or other bus masters, to share the system bus using a serial (daisy chain) priority scheme and allows up to 16 masters to share the MULTIBUS system bus with an external parallel priority decoder.

HIGH SPEED OFF-BOARD MEMORY

The iSBC 286/10A board can access off-board memory either over the MULTIBUS (P1) interface, or over the P2 interface. Memory transfers over the P2 interface are faster because the CPU board doesn't have to arbitrate for access to the MULTIBUS interface

Using the P2 interface, the iSBC 286/10A Board can be configured to operate with either a standard iLBX interface or with a high-performance, synchronous interface.

The iSBC 286/10A Board as supplied is configured to operate with a synchronous, P2 interface. This high-performance interface is designed to connect to Intel's new EX series of memory expansion boards to yield a CPU to memory read/write time of 0 wait-states. The EX memory expansion boards are available in sizes ranging from 512K bytes up to 4M bytes and available in sizes ranging from 512K bytes up to 2M bytes. Memory expansion boards from other manufacturers that meet the iLBX standard may also be used. CPU to memory access time is usually 1 or more wait-states depending on the speed of the memory used.

A total of four memory boards can be placed on the iLBX or synchronous interface bus. With 4M byte memory boards, this results in a total of 16M bytes on the memory expansion bus.

iSBX™ BUS MULTIMODULE™ ON-BOARD EXPANSION

Two 8/16-bit iSBX MULTIMODULE connectors are provided on the iSBC 286/10A microcomputer board. The iSBX interface connectors on the iSBC 286/10A provide all signals necessary to interface to the local on-board bus, including 16 data lines for maximum data transfer rates. iSBX MULTIMODULE boards designed with 8-bit data paths and using the 8-bit iSBX connector are also supported on the iSBC 286/10A microcomputer board. A broad range of iSBX MULTIMODULE options are available from Intel. Custom iSBX modules may also be designed.

SPECIFICATIONS

Word Size

Instruction—8, 16, 24, 32 or 40 bits
Data—8 or 16 bits

System Clock

CPU—8.0 MHz
Numeric Processor—5.3 or 8.0 MHz (Jumper Selectable)

Cycle Time

Basic Instruction—8.0 MHz—375 ns; 250 ns (assumes instruction in queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles)

Local Memory

Number of sockets—Four 28-pin JEDEC sites, expandable to 8 sites using iSBC 341 JEDEC Expansion Module

Maximum Size—256 KB

Compatible Devices—EPROM, up to 64K x 8 (Intel 27512)

Dual-Port Memory

Number of sockets—Four 28-pin JEDEC sites, expandable to 8 sites using iSBC 341 JEDEC Expansion Module

Maximum Size—128 KB

Compatible Devices—EPROM, up to 32K x 8 (Intel 27256)

SRAM
iRAM, up to 8K x 8 (Intel 2186)
E^2PROM, up to 2K x 8 (Intel 2817A)

I/O Capability

Parallel—Line printer interface, on-board functions, and four non-dedicated input bits

Serial—Two programmable channels using one 8274 device

Timers—Three programmable timers using one 8254 device

Expansion—Two 8/16-bit iSBX MULTIMODULE connectors

Serial Communications Characteristics

Synchronous—5–8 bit characters; internal or HDLC/SDLC character synchronization; automatic sync insertion; even or odd parity

Asynchronous—5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection; even or odd parity

Interrupt Capacity

Potential Interrupt Sources—25, 5 fixed, 20 jumper selectable

Interrupt Levels—16 vectored requests using two 8259As and the 80286's NMI line.

INTERFACES

MULTIBUS Bus—All signals TTL compatible

iSBX Bus—All signals TTL compatible

iLBX Bus—All signals TTL compatible

Synchronous Interface—All signals TTL compatible

Serial I/O—Channel A: RS232C/RS422/RS449 compatible, DCE or DTE; Channel B; RS232C compatible, DCE only

Timer—All signals TTL compatible

Interrupt Requests—All TTL compatible

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.4 in. (1.0 cm)
Minimum Slot Spacing: 0.6 in. (1.5 cm)
Weight: 14 oz. (397 gm)

Electrical Characteristics

DC Power Requirements: +5V, 7.0A; ±12V, 50 mA (serial I/O)

NOTE:
Does not include power for optional EPROM, E^2PROM, or RAM memory devices, or installed MULTIMODULE boards

Environmental Characteristics

Operating Temperature: 0°C to 60°C with 7 CFM airflow across board

Relative Humidity: to 90% (without condensation)

Reference Manual

147532-001—iSBC® 286/10A Hardware Reference Manual (order separately)

ORDERING INFORMATION

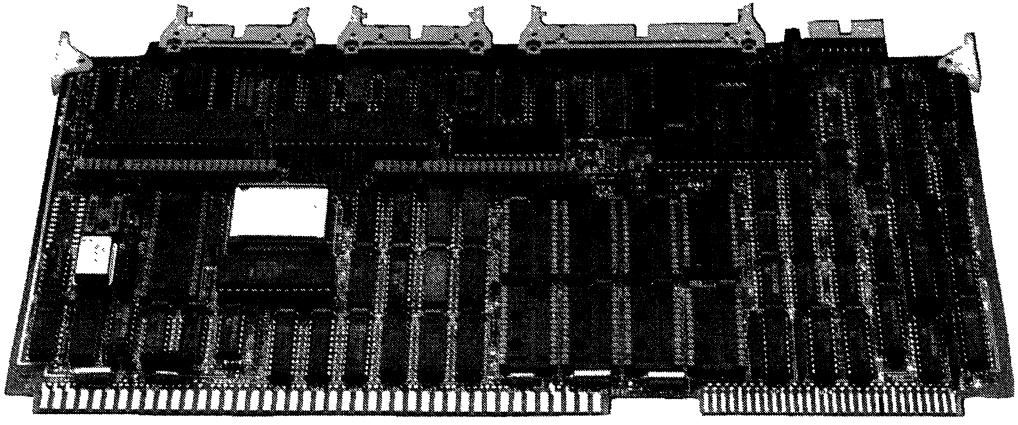
Part Number	Description
SBC 286/10A	Single Board Computer

iSBC® 186/03A*
SINGLE BOARD COMPUTER

- **8.0 MHz 80186 Microprocessor with Optional 8087 Numeric Data Processor**
- **Eight (Expandable to 12) JEDEC 28-Pin Sites**
- **Six Programmable Timers and 27 Levels of Vectored Interrupt Control**
- **MULTIBUS® Interface for System Expansion and Multimaster Configuration**
- **24 Programmable I/O Lines Configurable as a SCSI Interface, Centronics Interface or General Purpose I/O**
- **Two iSBX™ Bus Interface Connectors for Low Cost I/O Expansion**
- **iLBX™ (Local Bus Extension) Interface for High-Speed Memory Expansion**
- **Two Programmable Serial Interfaces; One RS 232C, the Other RS 232C or RS 422 Compatible**

The iSBC 186/03A Single Board Computer is a member of Intel's complete line of microcomputer modules and systems that take advantage of Intel's VLSI technology to provide economical, off-the-shelf, computer-based solutions for OEM applications. The board is a complete microcomputer system on a 7.05 x 12.0 inch printed circuit card. The CPU, system clock, memory, sockets, I/O ports and drivers, serial communications interface, priority interrupt logic and programmable timers, all reside on the board.

The iSBC 186/03A board incorporates the 80186 CPU and SCSI interface on one board. The extensive use of high integration VLSI has produced a high-performance single-board system. For large memory applications, the iLBX local bus expansion maintains this high performance.



230988-1

*The iSBC® 186/03A board is also manufactured under product code piSBC® 186/03A and siSBC® 186/03A by Intel Puerto Rico, Inc. and Intel Singapore, Ltd.

September 1989
Order Number: 230988-006

OVERVIEW

Operating Environment

The iSBC 186/03A single board computer features have been designed to meet the needs of numerous microcomputer applications. Typical applications include:

- Multiprocessing single board computer
- BITBUS master controller
- Stand-alone singel board system

MULTIPROCESSING SINGLE BOARD COMPUTER

High-performance systems often need to divide system functions among multiple processors. A multiprocessing single board computer distributes an applications processing load over multiple processors that communicate over a system bus. Since these applications use the system bus for inter-processor communication, it is required that each processor has local execution memory.

The iSBC 186/03A board supports loosely coupled multiprocessing (where each processor performs a specific function) through its MULTIBUS compatible architecture. The IEEE 796 system bus facilitates processor to processor communication, while the iLBX bus makes high-speed data and execution memory available to each CPU as shown in Figure 1. This architecture allows multiple processors to run in parallel enabling very high-performance applications.

BITBUS™ MASTER CONTROLLER

The BITBUS interconnect environment is a high performance low-cost microcontroller interconnect technology for distributed control of intelligent industrial machines such as robots and process controllers. The BITBUS interconnect is a special purpose serial bus which is ideally suited for the fast transmission of short messages between the microcontroller nodes in a modularly distributed system.

The iSBC 186/03A board can be implemented as the MULTIBUS-based master controller CPU which monitors, processes and updates the control status of the distributed system. The iSBX 344 board is used to interface the iSBC 186/03A board to the BITBUS interconnect. Actual message transfer over the iSBX bus can be accomplished by either software polling by the CPU or by using the on-chip 80186 DMA hardware instead of the CPU. Using DMA, the CPU is only required to start the DMA process and then poll for the completion of the message transfer, thus dramatically improving the data transmission rate and master control processor efficiency. The maximum transfer rates over the iSBX bus for the iSBC 186/03A board are about 900 messages/second in polled mode and 2500 messages/second in DMA mode. An 8 MHz iSBC 186/03A board in DMA mode is 3 times as fast as a typical 5 MHz iSBC 86/30 board running in polled mode. The iSBC 186/03A board in DMA mode provides the highest performance/price solution for BITBUS message transmission out of all of Intel's complete line of 16-bit CPU modules.

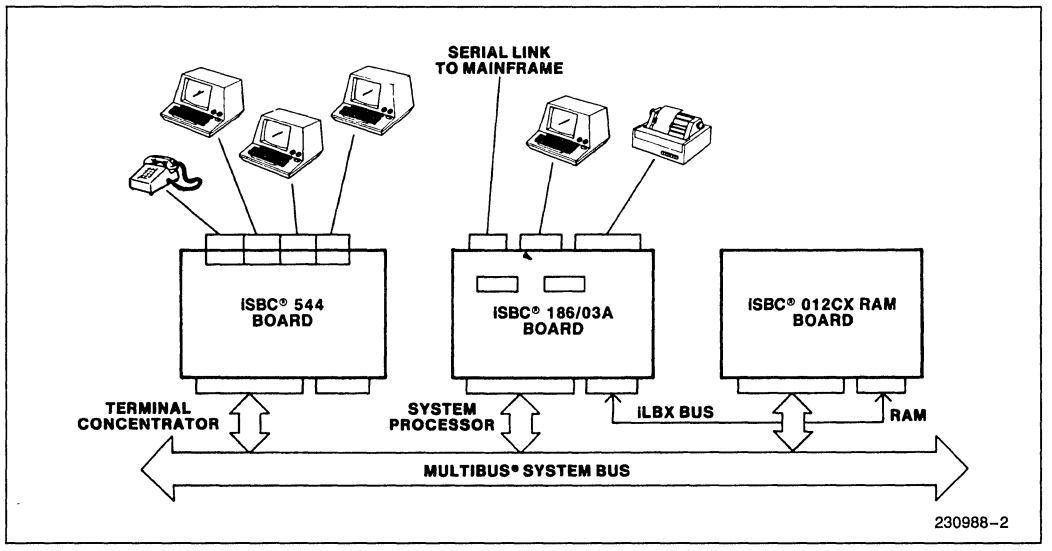


Figure 1. A Multiprocessing Single Board Computer Application

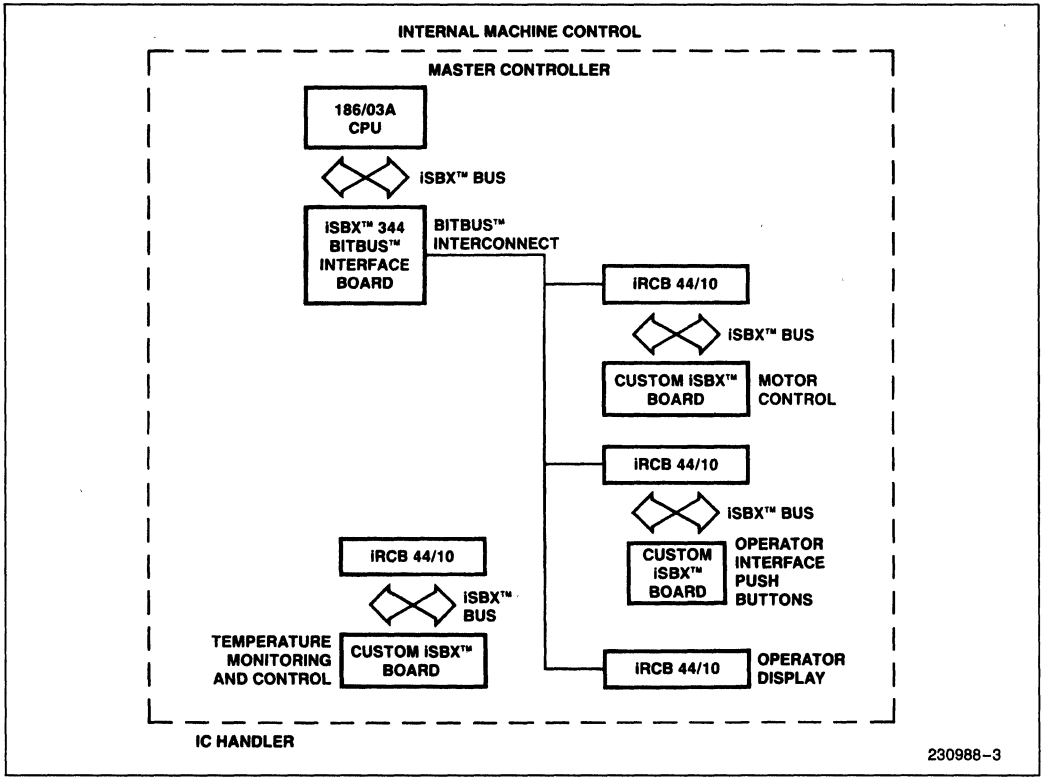


Figure 2. Sample iSBC® 186/03A BITBUS™ Master Application

STAND-ALONE SINGLE BOARD SYSTEM

A stand-alone single board system is a complete computer system on one board. By reducing the system's board count, the single board system saves space, power, and ultimately, costs. The on-board resources need to be capable of performing all of the basic system functions. These applications typically require terminal support, peripheral control, local RAM and program execution. In previous generations of single board computers, these functions could only be obtained with multiple board solutions.

The iSBC 186/03A board integrates all the functions of a general purpose system (CPU, memory, I/O and peripheral control) onto one board. The iSBC 186/03A board can also be customized as a single board system by the selection of memory and iSBX I/O options. The board's 8 JEDEC 28-pin sockets can accommodate a wide variety of byte-wide memory devices.

If more memory is needed, an optional iSBC 341 memory site expansion board can be added to provide an additional four JEDEC sites. Two iSBX MULTIMODULE™ boards can be added to the iSBC 186/03A board to customize the board's I/O capabilities. As shown in Figure 3, the iSBX connectors can support a single-board system with the analog input and output modules needed by machine or process control systems.

FUNCTIONAL DESCRIPTION

Architecture

The iSBC 186/03A board is functionally partitioned into six major sections: central processor, memory, SCSI compatible parallel interface, serial I/O, interrupt control and MULTIBUS bus expansion. These areas are illustrated in Figure 4.

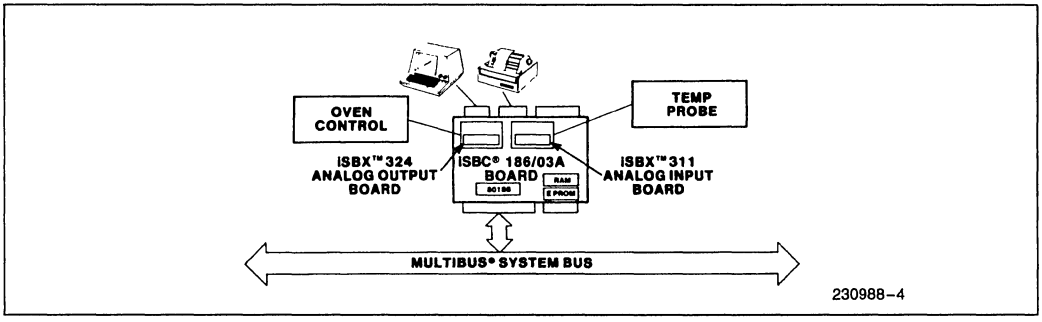


Figure 3. A Stand-Alone Single Board System Application

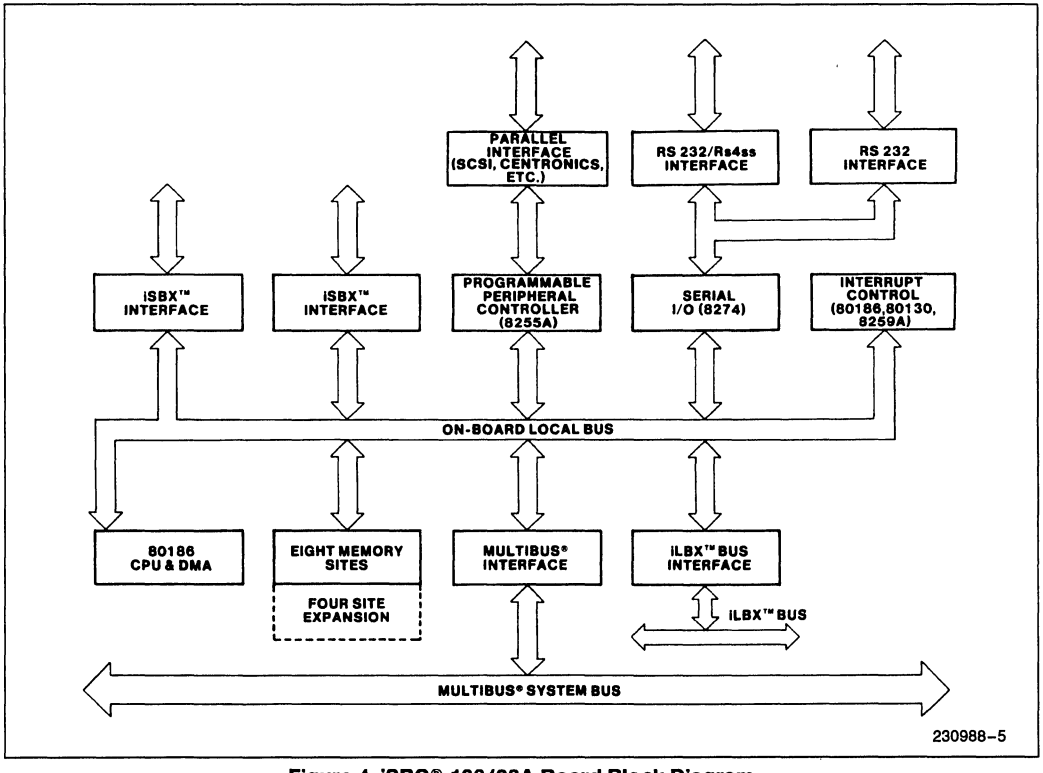


Figure 4. iSBC® 186/03A Board Block Diagram

CENTRAL PROCESSOR

The 80186 component is a high-integration 16-bit microprocessor. It combines several of the most common system components onto a single chip (i.e. Direct Memory Access, Interval Timers, Clock Generator and Programmable Interrupt Controller). The 80186 instruction set is a superset of the 8086. It maintains object code compatability while adding ten new instructions. Added instructions include: Block I/O, Enter and Leave subroutines, Push Immediate, Multiply Quick, Array Bounds Checking, Shift and Rotate by Immediate, and Pop and Push All.

Use of the 80130 component is limited to the 3 timers and 8 levels of interrupts available. Direct processor execution of the 16K bytes of iRMX I Operating System nucleus primitives is not supported.

An optional 8087 Numeric Data Processor may be installed by the user to dramatically improve the 186/03A board's numerical processing power. The interface between the 8087 and 80186 is provided by the factory-installed 82188 Integrated Bus Controller which completes the 80186 numeric data processing system. The 8087 Numeric Data Processor option adds 68 floating-point instructions and eight 80-bit floating point registers to the basic iSBC 186/03A board's programming capabilities. Depending on the application, the 8087 will increase the performance of floating point calculations by 50 to 100 times.

TIMERS

The 80186 provides three internal 16-bit programmable timers. Two of these are highly flexible and are connected to four external pins (two per timer). They can be used to count external events, time external events, generate nonrepetitive waveforms, etc. As shipped on the iSBC 186/03A board, these two timers are connected to the serial interface, and provide baud rate generation. The third timer is not connected to any external pins, and is useful for real-time coding and time-delay applications. In addition, this third timer can be used as a prescaler to the other two, or as a DMA request source. The 80130 provides three more programmable timers. One is a factory default baud rate generator and outputs an 8254 compatible square wave that can be used as an alternate baud rate source to either serial channel. The 80130's second timer is used as a system timer. The third timer is reserved for use by the iRMX Operating System. The system software configures each timer independently to select the desired function. Available functions include: interrupt on terminal count, programmable one-shot, rate generator, square-wave generator, software triggered strobe, hardware triggered strobe and event counter. The contents of each counter may be read at any time during system operation.

MEMORY

There are eight JEDEC 28-pin memory sites on the iSBC 186/03A board providing flexible memory expansion. Four of these sites (EPROM sites) may be used for EPROM or E^2PROM program storage, while the other four (RAM sites) may be used for static RAM or iRAM data storage or used as additional program storage. The eight sites can be extended to twelve by the addition of an iSBC 341 MULTIMODULE board. These additional sites will provide up to 64K bytes of RAM using 8K x 8 SRAM or iRAM devices. The EPROM sites (Bank B) are compatible with 8K x 8 (2764), 16K x 8 (27128A), 32K x 8 (27256), 64K x 8 (27512) as well as 2K x 8 (2817A) and 8K x 8 (2864) E^2PROMs. The RAM sites (Bank A) are compatible with all bytewide SRAM, iRAM or NVRAM devices. NVRAM usage requires additional circuitry in order to guarantee data retention. Bank A can be reassigned to upper memory just below the assigned memory space for Bank B to support additional EPROM or E^2PROMs.

Memory addressing for the JEDEC sites depends on the device type selected. The four EPROM sites are top justified in the 1 MB address space and must contain the power-on instructions. The device size determines the starting address of these devices. The four RAM sites are, by default, located starting at address 0. The addressing of these sites may be relocated to upper memory (immediately below the EPROM site addresses) in applications where these sites will contain additional program storage. The optional iSBC 341 MULTIMODULE sites are addressable immediately above the RAM site addresses.

Power-fail control and auxiliary power are provided for protection of the RAM sites when used with static RAM devices. A memory protect signal is provided through an auxiliary connector (J4) which, when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during system power-down sequences. An auxiliary power bus is also provided to allow separate power to RAM for systems requiring battery back-up of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

SCSI PERIPHERAL INTERFACE

The iSBC 186/03A board includes a parallel peripheral interface that consists of three 8-bit parallel ports. As shipped, these ports are configured for general purpose I/O. The parallel interface may be reconfigured to be compatible with the SCSI disk interface by adding two user-supplied and programmed Programmable Array Logic (PAL) devices, moving jumpers and installing a user-supplied 74LS640-1 device. Alternatively, the parallel interface may be reconfigured as a DMA controlled Centronics compatible line printer interface by adding one PAL and changing jumpers. Refer to the iSBC 186/03A Hardware Reference Manual for PAL equations and a detailed implementation procedure.

The SCSI (Small Computer Systems Interface) interface allows up to 8 mass storage peripherals such as Winchester disk drives, floppy disk drives and tape drives to be connected directly to the iSBC 186/03A board. Intel's iSBC 186/03A board utilizes a single initiator, single target implementation of the SCSI bus specification. Bus arbitration and deselect/reselect SCSI features are not supported. Single host, multiple target configurations can be used. However, the iSBC 186/03A board will stay connected to one target until the transaction is completed before switching to the second target. The iSBC 186/03A board's SCSI interface implements a 5 megabit/second transfer rate. A sample SCSI application is shown in Figure 5.

The Centronics interface requires very little software overhead since a PAL device is used to provide necessary handshake timing. Interrupts are generated for printer fault conditions and a DMA request is issued for every character. The interface supports Centronics type printers compatible with models 702 and 737.

SERIAL I/O

The iSBC 186/03A Single Board Computer contains two programmable communications interfaces using the Intel 8274 Multi-Protocol Serial Controller (MPSC).

Two 80186 timer outputs are used as software selectable baud rate generators capable of supplying the serial channels with common communications frequencies. An 80130 baud rate timer may be jumpered to either serial port to provide higher frequency baud rates. The mode of operation (i.e., asynchronous, byte synchronous or bisynchronous protocols), data format, control character format, parity, and baud rate are all under program control. The 8274 provides full duplex, double buffered transmit and receive capability. Parity, overrun, and framing error detection are all incorporated in the MPSC. The iSBC 186/03A board supports operation in the polled, interrupt and DMA driven interfaces through jumper options. The default configuration is with channel A as RS422A/RS449, channel B as RS232C. Channel A can optionally be configured to support RS232C. Both channels are default configured as data set (DCE). Channel A can be reconfigured as data terminal (DTE) for connection to a modem-type device.

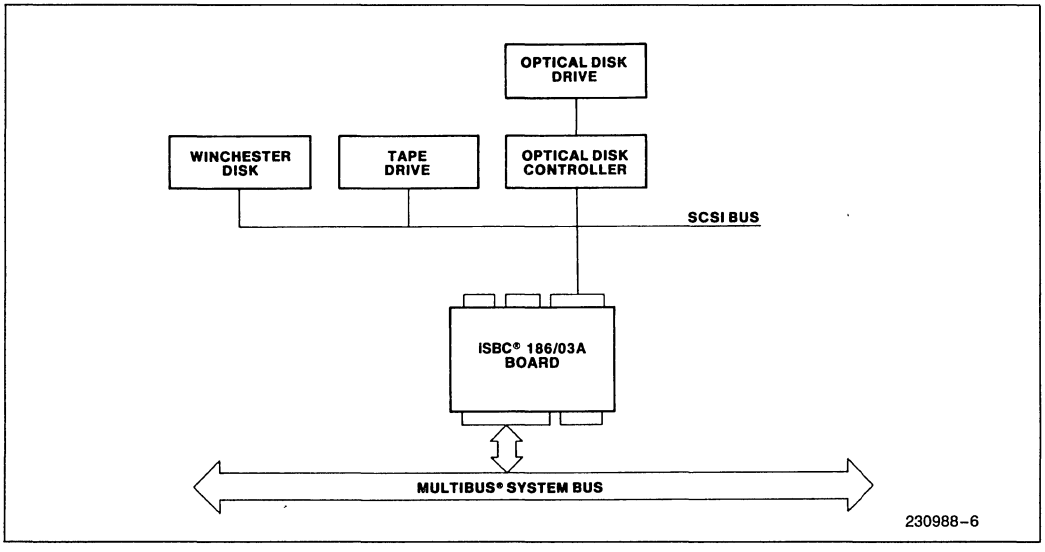


Figure 5. Sample SCSI Application

INTERRUPT CONTROL

The iSBC 186/03A board provides 27 on-board vectored interrupt levels to service interrupts generated from 33 possible sources.

The interrupts are serviced by four programmable interrupt controllers (PICs): one in the 80186 component, one in the 80130 component, one in the 8259A component and one in the 8274 component. The 80186, 8259A and 8274 PICs act as slaves to the 80130 master PIC. The highest priority interrupt is the Non-Maskable Interrupt (NMI) line which is tied directly to the 80186 CPU. This interrupt is typically used to signal catastrophic events (e.g. power failure). The PICs provide prioritization and vectoring for the other 26 interrupt requests from on-board I/O resources and from the MULTIBUS system bus. The PICs then resolve the requests according to the programmable priority resolution mode, and if appropriate, issue an interrupt to the CPU.

MULTIBUS® SYSTEM BUS—IEEE 796

The MULTIBUS system bus is an industry standard (IEEE 796) microcomputer bus structure. Both 8- and 16-bit single board computers are supported on the IEEE 796 structure with 20 or 24 address and 16 data lines. In its simplest application, the system bus allows expansion of functions already contained on a single board computer (e.g., memory and I/O). However, the IEEE 796 bus also allows very powerful distributed processing configurations with multiple processors and intelligent slave, I/O and peripheral boards capable of solving the most demanding microcomputer applications. The MULTIBUS system bus is supported with a broad array of board-level products, LSI interface components, detailed published specifications and application notes.

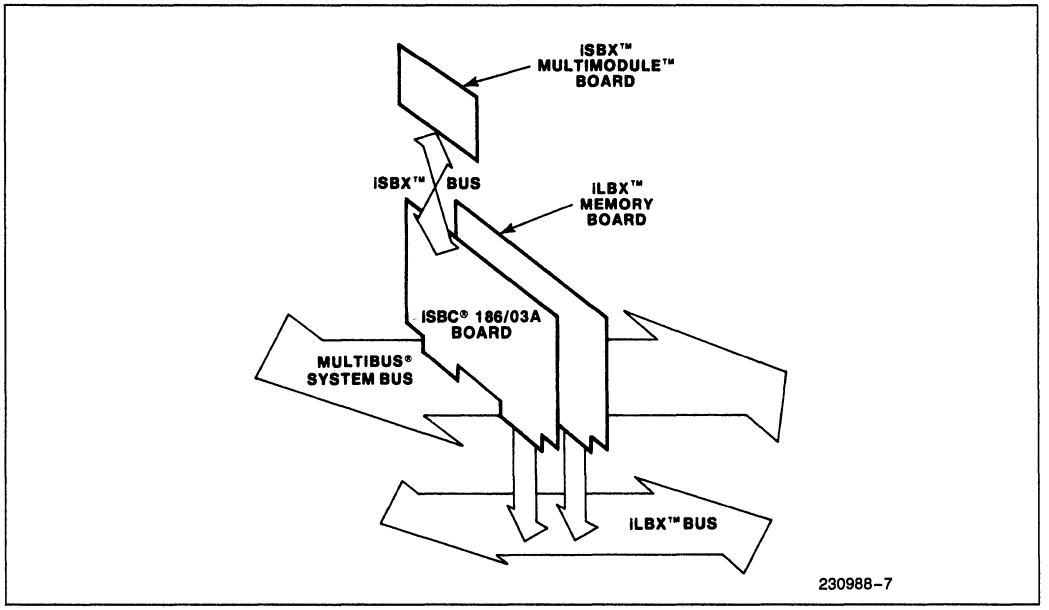


Figure 6. iSBC® 186/03A Board System Architecture

iLBX™ BUS—LOCAL BUS EXTENSION

The iSBC 186/03A board provides a local bus extension (iLBX) interface. This standard extension allows on-board memory performance with physically off-board memory. The combination of a CPU board and iLBX memory boards is architecturally equivalent to a single board computer and thus can be called a "virtual single board computer". The iLBX bus is implemented over the P2 connector and requires independent cabling or backplane connection.

iSBX™ BUS MULTIMODULE™ ON-BOARD EXPANSION

Two iSBX MULTIMODULE board connectors are provided on the iSBC 186/03A microcomputer board. Through these connectors, additional on-board I/O functions may be added. iSBX MULTIMODULE boards optimally support functions provided by VLSI peripheral components such as additional parallel and serial I/O, analog I/O, and graphics control. The iSBX bus connectors on the iSBC 186/03A board provide all signals necessary to interface to the local on-board bus, including 16 data lines for maximum data transfer rates. MULTIMODULE boards designed with 8-bit data paths and using the 8-bit iSBX connector are also supported on the iSBC 186/03A board. A broad range of iSBX MULTIMODULE options are available from Intel. Custom iSBX bus modules may also be designed.

SPECIFICATIONS

Word Size

Instruction—8, 16, 24 or 32 bits
Data—8 or 16 bits

System Clock

8.0 MHz

Numeric Data Processor (Optional)

8087-1

Basic Instruction Cycle Time

750 ns
250 ns (assumes instruction in the queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e. two clock cycles plus instruction fetch). Zero wait-state memory is assumed.

MEMORY CAPACITY/ADDRESSING

Four EPROM Sites		
Device	**Capacity**	**Address Range**
2764 EPROM	32 KB	$F8000_H$–$FFFFF_H$
27128 EPROM	64 KB	$F0000_H$–$FFFFF_H$
27256 EPROM	128 KB	$E0000_H$–$FFFFF_H$
27512 EPROM	256 KB	$C0000_H$–$FFFFF_H$
Four RAM Sites		
Device	**Capacity**	**Address Range**
2K SRAM	8 KB	0–$01FFF_H$
8K SRAM	32 KB	0–$07FFF_H$
32K SRAM	128 KB	0–$1FFFF_H$
2186 RAM	32 KB	0–$07FFF_H$
2817A E^2PROM	8 KB	$F0000_H$–$F7FFF_H$*
2764 EPROM	32 KB	$F0000_H$–$F7FFF_H$ (below EPROM Sites)
27128 EPROM	64 KB	$E0000_H$–$EFFFF_H$ (below EPROM Sites)
27256 EPROM	128 KB	$C0000_H$–$DFFFF_H$ (below EPROM Sites)
Four iSBC® 341 Expansion Sites		
Device	**Capacity**	**Address Range**
2K SRAM	8 KB	02000_H–$03FFF_H$
8K SRAM	32 KB	08000_H–$0FFFF_H$
32K SRAM	128 KB	10000_H–$1FFFF_H$
2186 RAM	32 KB	08000_H–$0FFFF_H$
2817A E^2PROM	8 KB	02000_H–$03FFF_H$**

NOTE:
All on board memory is local to the CPU (i.e. not dual-ported).

*Must use 8k x 8 decode option, there are four copies of the E^2PROM in the 8K x 8 address area.

**(May be mixed with 2K x 8 SRAM)

Serial Communications Characteristics

Synchronous— 5–8 bit characters, internal or external character synchronization; automatic sync insertion; break character generation

Asynchronous— 5–8 bit characters; 1, ½, or 2 stop bit; false start bit detection.

Interface Compliance

MULTIBUS— IEEE 796 compliance: Master D16 M24 116 VO EL

iSBX Bus— Two 8/16 bit iSBX bus connectors allow use of up to 2 single-wide modules or 1 single-wide and 1 double-wide module. Intel iSBX bus compliance: D16/16 DMA

iLBX— Intel iLBX bus compliance: PM D16

Serial— Channel A: Configurable as RS 422A or RS 232C compatible, configurable as a data set or data terminal

Channel B: RS 232C compatible, configured as data set

Parallel I/O— SCSI (ANSI—X3T9, 2/82-s) compatible or Centronics 702 or 737 compatible (requires user supplied PALs and 74LS640-1)

ORDERING INFORMATION

Part Number	Description
SBC 186/03A	186-based Single Board Computer

REFERENCE MANUAL

iSBC® 186/03A Single Board Computer Hardware Reference Manual—Order Number 148060

PHYSICAL CHARACTERISTICS

Width: 12.00 in. (30.48 cm)

Length: 7.05 in (17.90 cm)

Height: 0.50 in. (1.78 cm)

Weight: 13 ounces

ENVIRONMENTAL CHARACTERISTICS

Operating Temperature: 0°C to 60°C at 6 CFM airflow over the board.

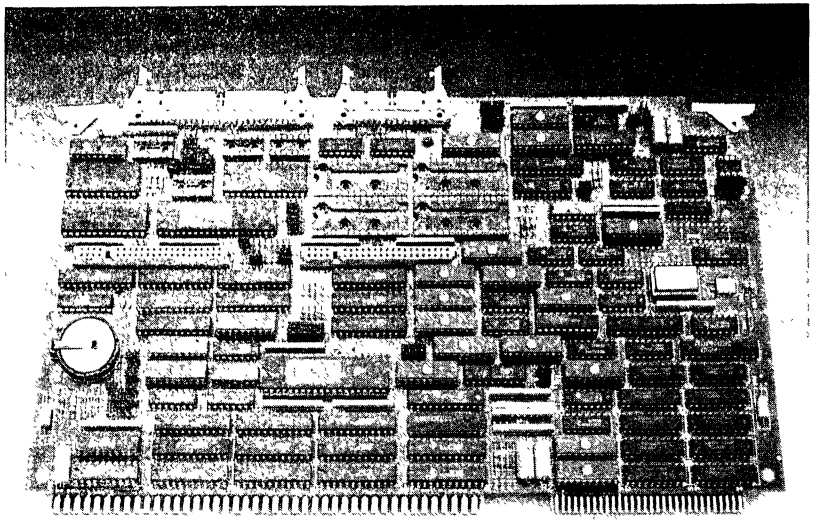
Relative Humidity: to 90% (without condensation)

ELECTRICAL CHARACTERISTICS

The maximum power required per voltage is shown below. These numbers do not include the power required by the optional memory devices, SCSI PALs, battery back-up or expansion modules.

Voltage (volts)	Max. Current (amps)	Max Power (watts)
+5	5.4	27
+12	0.04	0.48
−12	0.04	0.48

iSBC® 86C/38* SINGLE BOARD COMPUTER



CMOS 80C86-BASED MULTIBUS®I SINGLE-BOARD COMPUTER

The iSBC® 86C/38 Single-Board Computer is a high-performance, low-power MULTIBUS®I CPU board based on advanced CMOS (complementary metal oxide semiconductor) technology. The board features Intel's 8 MHz 80C86 microprocessor—which provides the highest performance possible with static CMOS devices—a full megabyte of zero wait state DRAM memory, and power consumption of typically less than 8 watts when operating at full speed. The board's high performance, low power consumption, low heat generation and high reliability make it ideal for embedded real-time applications in harsh industrial environments.

STANDARD FEATURES:

- Advanced CMOS 8 MHz 80C86 microprocessor
- 1 Mbyte of dual-port, zero wait state DRAM with parity
- Sockets for up to 512 Kbytes of standard 32-pin JEDEC EPROM devices
- Real-time clock/calendar with on-board battery backup
- Temperature-sensing device socket
- Optional 8087 numeric data processor with iSBC 337A MULTIMODULE™
- Upward-compatible with iSBC 86/35
- iRMX® Real-Time Operating System support

intel

* The iSBC® 86C/38 Board is also manufactured under product code piSBC® 86C/38 by Intel Puerto Rico, Inc

September, 1989
Order Number 280630-002

FEATURES

CMOS TECHNOLOGY FOR LOW POWER, LOW HEAT

The Intel iSBC® 86C/38 has been implemented entirely in CMOS, from the 80C86 CPU and EPLDs to the descrete logic and peripheral components. CMOS means low power consumption and low heat generation.

When running at full speed (8MHz), the iSBC 86C/38 typically requires less than 8 watts of power. However, a power-saving Slow Mode further reduces power consumption to about 4 watts when operating speed is reduced to 1 MHz.

Slow Mode operation is especially useful during temporary or emergency conditions when battery power is called into use. In a power-fail situation, for instance, Slow Mode operation allows the uninterrupted processing of an application on battery power.

The iSBC 86C/38 generates so little heat that it can operate without any air flow. This allows elimination of fans and other expensive cooling equipment and operation of the iSBC 86C/38 in a sealed enclosure, protected from harsh environments.

HIGH RELIABILITY

The iSBC 86C/38 features improved reliability on several levels. First, CMOS technology is inherently more reliable than NMOS technology: because devices run at lower junction temperatures, they last longer.

Parity error checking in the DRAM circuitry improves system integrity by detecting memory errors.

Finally, improved pin and socket I/O connectors with locking tabs assure secure connections of cables to the board.

MORE MEMORY

The amount of on-board memory has been doubled in the iSBC 86C/38 from earlier iSBC 86/35 board models. The iSBC 86C/38 comes with a full megabyte of zero wait state dynamic RAM, supporting the full 8086 address space. A full megabyte of on-board memory also eliminates the need to add DRAM modules, preserving the economy of a single-slot solution.

UPWARD-COMPABILITY WITH ISBC 86/35 DESIGNS

The iSBC 86C/38 provides complete hardware and software compatibility with Intel iSBC 86/35 designs. All features supported on the iSBC 86/35 board run on the iSBC 86C/38 board with no changes. This includes full access to the MULTIBUS I 16 Mbyte memory address range and support for MULTIBUS I multimaster, 8087 math coprocessor, iSBC 86/35 I/O devices, iSBX connectors and interrupt capability.

PERFECT FOR REAL-TIME EMBEDDED APPLICATIONS

Real-time process control and industrial automation applicaitons frequently require the CPU and control system to be physically located on the factory floor or in the field. These environments are typically harsh, full of dust, dirt, electrical noise and widely fluctuating temperatures.

Because the iSBC 86C/38 generates so little heat and can operate without cooling, it can be placed in a sealed enclosure, protected from harsh factory environments. It also offers excellent noise immunity and tolerance to extreme temperatures.

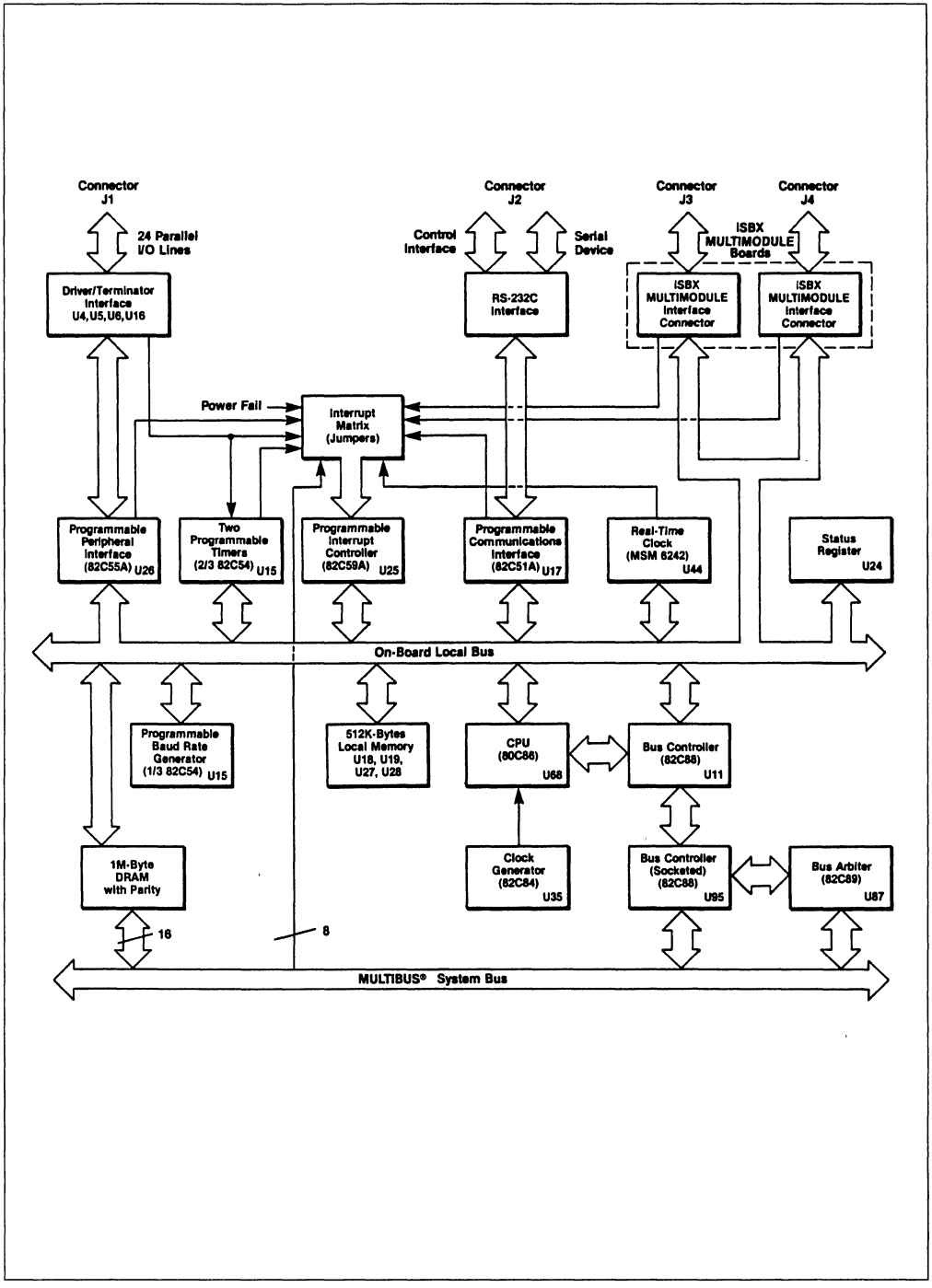


Figure 1: iSBC®86C/38 Block Diagram

SPECIFICATIONS

Central Processor

80C86 CPU	8 MHz
Numberic Processor	8 MHz
iSBC 337A MULTIMODULE	

Cycle Time

Basic Instruction	8 MHz 500ns (assumes instruction in queue)

Note: Basic instruction cycle is defined as the fastest instruction time (i.e. four clock cycles)

DRAM Memory

On-board parity memory	1 Mbyte, 0 Wait States at 8 MHz

Note: Power fail battery backup capability via P2 connector.

EPROM Memory

Number of sockets	Four 32-pin JEDEC Sites (compatible with 28-pin and 32-pin devices)
Device access speeds	265ns (minimum) to 640ns (maximum)
Maximum memory	512 Kb with 27010 (1 M bit) EPROMS

Note: EPROM, E^2PROM (read only), and Static RAM devices are supported.

I/O CAPABILITY

Parallel Channel Three 8-bit parallel ports (50 pin socket connectors) using an 82C55A

Serial Channel One RS-232-C channel using an 82C51 device with speeds from 110 to 19.2 Kb

iSBX Expansion Two 8/16-bit iSBX interface connectors for single or double wide iSBX MULTIMODULE boards

Real Time Clock/Calendar

An OKI MSM6242 provides real time clock/calendar capability with clock operation in either 12 or 24 hour format. The clock/calendar is sustained up to 10,000 hours by an on-board BR2325 lithium battery.

Temperature Sensing

Temperature sensing is an optional capability, allowing system designers to choose the appropriate level of temperature sensing for their application. A socket is on-board which supports four-pin temperature sensor devices.

Interrupt Capacity

Potential Interrupt Sources	37 jumper selectable
Interrupt Levels	9 using the 82C59A device and the 80C86 NMI line

Note: Bus Vetored Interrupt capability is supported.

Timers

Three programmable timer/counters using one 82C54 device

Interfaces

MULTIBUS Bus	All signals TTL compatible
iSBX Bus	All signals TTL compatible
Parallel I/O	All signals TTL compatible
Serial I/O	RS-232-C

POWER REQUIREMENTS/ CONSUMPTION

	8 MHz	*1MHz*
Maximum:		
+5V	1.56 A, 7.8 Watts	.8 A, 4.0 Watts
+12V	.06 A, .72 Watts	.06 A, .72 Watts
−12V	.08 A, .96 Watts	.08 A, .96 Watts
Typical:		
+5V	.82 A, 4.1 Watts	.7 A, 3.5 Watts
+12V	.04 A, .48 Watts	.04 A, .48 Watts
−12V	.06 A, .72 Watts	.06 A, .72 Watts

Note: Does not include power for iSBC modules, iSBX modules or EPROM memory.

ENVIRONMENTAL REQUIREMENTS

Operating Temperature	0° to +60°C at zero LFM air flow
Relative Humidity	0 to 95% noncondensing
Storage Temperature	−40° to +70°C

PHYSICAL CHARACTERISTICS

Dimensions

Width:	12.00 in (30.48 cm)
Depth:	7.05 in. (17.91 cm)
Height:	.375 in. (.96 cm)

Recommended Minimum Cardcage Slot Spacing
.6 in (1.5 cm) without iSBC 337A or iSBX MULTIMODULE
1.2 in (3.0 cm) with iSBC 337A or iSBX MULTIMODULE

Approximate Weight 21.5 oz (609.5 gm)

REFERENCE MANUAL

454554—iSBC 86C/38 Single Board Compuer User's Guide

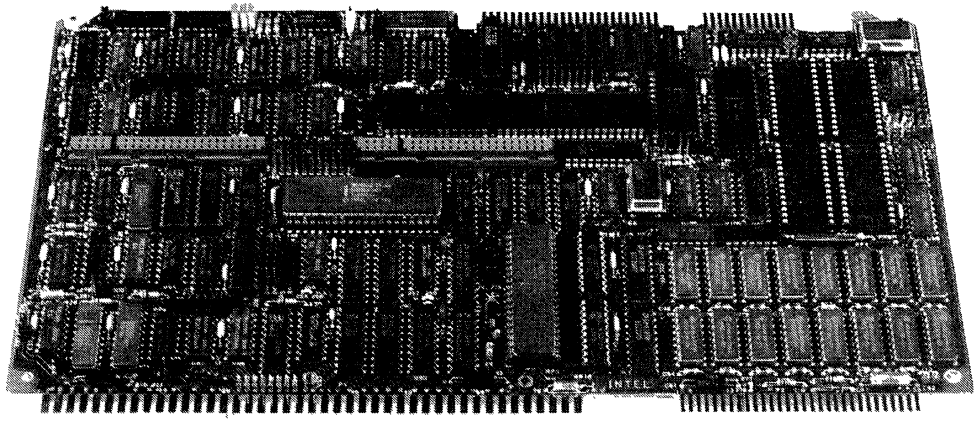
ORDERING INFORMATION

SBC 86C38	Single Board Computer

iSBC® 86/35*
SINGLE BOARD COMPUTER

- 8086 (8086-2) Microprocessor with 5 or 8 MHz CPU Clock
- Optional Numeric Data Processor with iSBC® 337A MULTIMODULE™
- Upward Compatible with iSBC 86/30 Single Board Computer
- 512K Bytes of Dual-Port Read/Write Memory Expandable On-Board to 640K or 1M Bytes
- Sockets for up to 128K Bytes of JEDEC 24/28-Pin Standard Memory Devices
- Two iSBX™ Bus Connectors
- 24 Programmable Parallel I/O Lines
- Programmable Synchronous/ Asynchronous RS232C Compatible Serial Interface with Software Selectable Baud Rates
- Three Programmable 16-Bit BCD or Binary Timers/Event Counters
- 9 Levels of Vectored Interrupt Control, Expandable Off Board to 65 Levels

The iSBC 86/35 Single Board Computer is a member of Intel's complete line of OEM microcomputer systems that take full advantage of Intel's technology to provide economical, self-contained, computer-based solutions for OEM applications. The board is a complete computer system containing the CPU, system clock, dual port read/write memory, nonvolatile read only memory, I/O ports and drivers, serial communications interface, priority interrupt logic and programmable timers.



210219-1

*The iSBC® 86/35 is also manufactured under product code piSBC 86/35 and siSBC 86/35 by Intel Puerto Rico, Inc., and Intel Singapore, Ltd.

November 1989
Order Number: 210219-005

FUNCTIONAL DESCRIPTION

Overview

The iSBC 86/35 board combines the industry standard 8086 CPU with up to a megabyte page of board resident, dual ported system memory. By placing the direct memory addressing capability of the 8086 CPU on board, MULTIBUS® access to system memory can be eliminated, significantly improving system throughput.

Central Processing Unit

The central processor for the iSBC 86/35 board is Intel's 8086-2 CPU. A clock rate of 8 MHz is supported with a jumper selectable option for 5 MHz. The CPU architecture includes four 16-bit byte addressable data registers, two 16-bit index registers, all accessed by a total of 24 operand addressing modes for comprehensive memory addressing and for support of the data structures required for today's structured, high level languages as well as assembly language.

Instruction Set

The 8086 instruction repertoire includes variable length instruction format (including double operand instructions), 8-bit and 16-bit signed and unsigned arithmetic operators for binary, BCD and unpacked ASCII data, and iterative word and byte string manipulation functions.

For enhanced 5 or 8 MHz numerics processing capability, the iSBC 337A MULTIMODULE Numeric Data Processor extends the architecture and data set. Over 60 numeric instructions offer arithmetic, trigonometric, transcendental, logarithmic and exponential instructions. Supported data types include 16-, 32-, and 64-bit integer, and 32- and 64-bit floating point, 18-digit packed BCD and 80-bit temporary.

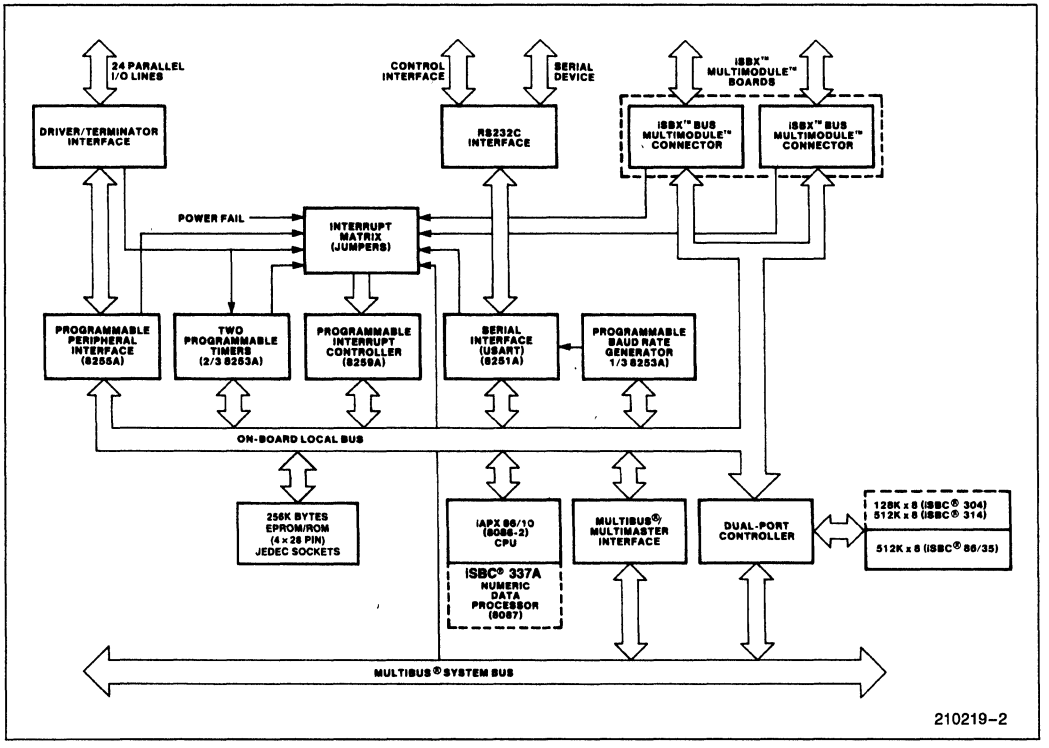


Figure 1. iSBC® 86/35 Block Diagram

Architectural Features

A 6-byte instruction queue provides pre-fetching of sequential instructions and can reduce the 750 ns minimum instruction cycle to 250 ns for queued instructions. The stack-oriented architecture readily supports modular programming by facilitating fast, simple, inter-modular communication, and other programming constructs needed for asynchronous real-time systems. The memory expansion capabilities offer a 1 megabyte addressing range. The dynamic relocation scheme allows ease in segmentation of pure procedure and data for efficient memory utilization. Four segment registers (code, stack, data, extra) contain program loaded offset values which are used to map 16-bit addresses to 20-bit addresses. Each register maps 64K bytes at a time and activation of a specific register is controlled explicitly by program control and is also selected implicitly by specific functions and instructions.

RAM Capabilities

The iSBC 86/35 microcomputer contains 512K bytes of dual-port dynamic RAM which may be expanded on-board by adding a RAM Multimodule board as an option. The on-board RAM may be expanded to 640K bytes with the iSBC 304 MULTIMODULE board mounted onto the iSBC 85/35 board. Likewise, the iSBC 86/35 microcomputer may be expanded to 1 Megabyte with the iSBC 314 MULTIMODULE board option.

The dual-port controller allows access to the on-board RAM (including RAM MULTIMODULE board options) from the iSBC 86/35 board and from any other MULTIBUS master via the system bus. Segments of on-board RAM may be configured as a private resource, protected from MULTIBUS system access.

EPROM Capabilities

Four 28-pin JEDEC sockets are provided for the use of Intel 2764, 27128, 27256, 27512, EPROMs and their respective ROMs. When using 27512, the on-board EPROM capacity is 256K bytes. Other JEDEC standard pinout devices are also supported, including byte-wide static RAMs.

Parallel I/O Interface

The iSBC 86/35 Single Board Computer contains 24 programmable parallel I/O lines implemented using the Intel 8255A Programmable Peripheral Interface. The system software is used to configure the I/O lines in any combination of unidirectional input/output and bidirectional ports indicated in Table 1. In order to take advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators, allowing the selection of the appropriate combination of optional line drivers and terminators with the required drive/termination characteristics. The 24 programmable I/O lines and signal ground lines are brought out to a 50-pin edge connector.

Serial I/O

A programmable communications interface using the Intel 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the iSBC 86/35 board. A software selectable baud rate generator provides the USART with all common communication frequencies. The mode of operation (i.e., synchronous or asynchronous), data format, control character format, parity, and baud rate are all under program control. The 8251A provides full duplex, double buffered transmit and receive capability. Parity, overrun, and framing error detection are all incorporated in the USART. The RS232C command lines, serial data lines and signal ground line are brought out to a 26-pin edge connector.

Programmable Timers

The iSBC 86/35 board provides three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8253 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate timer intervals under software control. Routing for the outputs and gate/trigger inputs of two of these counters is jumper selectable. The outputs may be independently routed to the 8259A Programmable Interrupt Controller and to the I/O terminators associated with the 8255A to allow external devices or an 8255A port to gate the timer or to count external events. The third interval timer in the 8253 provides the programmable baud rate generator for the iSBC 86/35 board's RS232C USART serial port. The system software configures each timer independently to select the desired function. The contents of each counter may be read at any time during system operation.

iSBX™ MULTIMODULE™ On-Board Expansion

Two 8/16-bit iSBX MULTIMODULE connectors are provided on the iSBC 86/35 microcomputer. Through these connectors, additional on-board I/O functions may be added. The iSBX connectors on the iSBC 86/35 board provides all signals necessary to interface to the local on-board bus, including 16 data lines for maximum data transfer rates. iSBX

MULTIMODULE boards designed with 8-bit data paths and using the 8-bit iSBX connector are also supported on the iSBC 86/35 microcomputer. A broad range of iSBX MULTIMODULE options are available from Intel. Custom iSBX modules may also be designed for use on the iSBC 86/35 board.

Multimaster Capabilities

The iSBC 86/35 board provides full MULTIBUS arbitration control logic. This control logic allows both serial (daisy chain) and parallel priority schemes. The serial scheme allows up to three iSBC 86/35 boards/bus masters to share the MULTIBUS system bus; while up to 16 masters may be connected using the parallel scheme and external decode logic.

Interrupt Capability

The iSBC 86/35 board provides 9 vectored interrupt levels. The highest level is the NMI (Non-Maskable Interrupt) line which is directly tied to the 8086-2 CPU. This interrupt is typically used for signaling catastrophic events (e.g., power failure). The Intel 8259A Programmable Interrupt Controller (PIC) provides control and vectoring for the next eight interrupt levels. A selection of four priority processing modes is available for use in designing request processing configurations to match system requirements for efficient interrupt servicing with minimal latencies. Operating mode and priority assignments may be reconfigured dynamically via software at any time during system operation. The PIC accepts interrupt requests from all on-board I/O resources and from the MULTIBUS system bus. The PIC then resolves requests according to the selected mode and, if appropriate, issues an interrupt to the CPU. Any combination of interrupt levels may be masked via software, by storing a single byte in the interrupt mask register of the PIC. In systems requiring additional interrupt levels, slave 8259A PICs may be interfaced via the MULTIBUS system bus, to generate additional vector addresses, yielding a total of 65 unique interrupt levels.

Interrupt requests to be serviced by the iSBC 86/35 board may originate from 28 sources. All interrupt signals are brought to the interrupt jumper matrix where any combination of interrupt sources may be strapped to the desired interrupt request level on the 8259A PIC or the NMI input to the CPU directly.

Power-Fail Control and Auxiliary Power

Control logic is included to accept a power-fail interrupt in conjunction with the AC-low signal from the Power Supply to initiate an orderly shut down of the system in the event of a power failure. Additionally, an active-low TTL compatible memory protect signal is brought out on the auxiliary connector which when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during system powerdown sequences. An auxiliary power bus is also provided to allow separate power to RAM for systems requiring battery backup of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

SPECIFICATIONS

Word Size

INSTRUCTION — 8, 16, 24, or 32 bits

DATA — 8, 16 bits

System Clock

5 MHz or 8 MHz ±0.1% (jumper selectable)

Cycle Time

BASIC INSTRUCTION CYCLE

8 MHz — 250 ns (assumes instruction in the queue)

5 MHz — 400 ns (assumes instruction in the queue)

NOTE:

Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles). Jumper selectable for 1 wait-state on-board memory access.

Memory Capacity/Addressing

ON-BOARD EPROM

Device	Total Capacity	Address Range
2764	32K bytes	$F8000-FFFFF_H$
27128	64K bytes	$F0000-FFFFF_H$
27256	128K bytes	$E0000-FFFFF_H$
27512	256K bytes	$D0000-FFFFF_H$

ON-BOARD RAM

Board	Total Capacity	Address Range
iSBC 86/35	512K bytes	$0-7FFFF_H$

WITH MULTIMODULE™ RAM

Board	Total Capacity	Address Range
iSBC 304	640K bytes	$8-9\ FFFF_H$
iSBC 314	1M bytes	$8-FFFFF_H$

Serial Communications Characteristics

SYNCHRONOUS—5–8 bit characters; internal or external character synchronization; automatic sync insertion

ASYNCHRONOUS—5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection

Interfaces

MULTIBUS®—All signals TTL compatible

iSBX™ BUS—All signals TTL compatible

PARALLEL I/O—All signals TTL compatible

SERIAL I/O—RS232C compatible, configurable as a data set or data terminal

TIMER—All signals TTL compatible

INTERRUPT REQUESTS—All TTL compatible

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.70 in. (1.78 cm)
Weight: 14 oz. (388 gm)

Electrical Characteristics

DC POWER REQUIREMENTS

Configuration	Current Requirements (All Voltages ±5%)		
	+5V	+12V	−12V
Without EPROM(1)	5.1A	25 mA	23 mA
RAM only(2)	660 mA	—	—
With 32K EPROM(3) (using 2764)	5.6A	25 mA	23 mA
With 64K EPROM (using 27128)	5.7A	25 mA	23 mA
With 128K EPROM (using 27256)	5.8A	25 mA	23 mA

NOTES:
1. Does not include power for optional ROM/EPROM, I/O drivers, and I/O terminators.
2. RAM chips powered via auxiliary power bus in power-down mode.
3. Includes power required for 4 ROM/EPROM chips, and I/O terminators installed for 16 I/O lines; all terminator inputs low.

Environmental Characteristics

OPERATING TEMPERATURE — 0°C to 55°C @ 200 linear feet per minute (LFM) air velocity

RELATIVE HUMIDITY — to 90% (without condensation)

Reference Manual

146245-002 — iSBC 86/35 Hardware Reference Manual (NOT SUPPLIED)

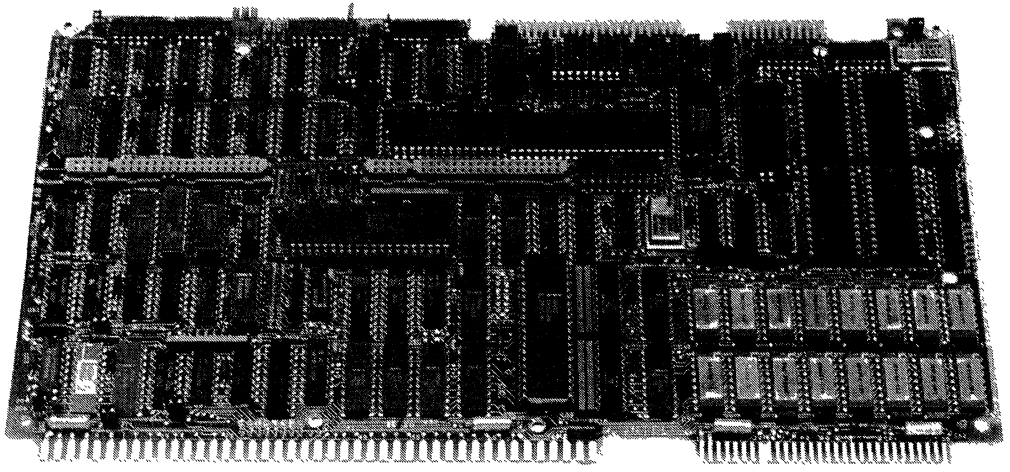
ORDERING INFORMATION

Part Number	Description
SBC 86/35	Single Board Computer

iSBC® 86/14* AND iSBC® 86/30* SINGLE BOARD COMPUTERS

- 8086 Microprocessor with 5 or 8 MHz CPU Clock
- Fully Software Compatible with iSBC® 86/12A Single Board Computer
- Optional Numeric Data Processor with iSBC® 337A MULTIMODULE™
- 32K/128K bytes of Dual-Port Read/Write Memory Expandable On-Board to 256K bytes with On-Board Refresh
- Sockets for up to 64K bytes of JEDEC 24/28-pin Standard Memory Devices
- Two iSBX™ Bus Connectors
- 24 Programmable Parallel I/O Lines
- Programmable Synchronous/Asynchronous RS232C Compatible Serial Interface with Software Selectable Baud Rates
- Two Programmable 16-Bit BCD or Binary Timers/Event Counters
- 9 Levels of Vectored Interrupt Control, Expandable to 65 Levels
- MULTIBUS® Interface for Multimaster Configurations and System Expansion

The iSBC 86/14 and iSBC 86/30 Single Board Computers are members of Intel's complete line of OEM microcomputer systems which take full advantage of Intel's technology to provide economical, self-contained, computer-based solutions for OEM applications. The CPU, system clock, read/write memory, nonvolatile read only memory, I/O ports and drivers, serial communications interface, priority interrupt logic and programmable timers, all reside on the boards.



280007-1

*The iSBC® 8614 and iSBC® 86/30 are also manufactured under product code piSBC 86/14, piSBC 86/30 or siSBC 86/14, siSBC 86/30 by Intel Puerto Rico, Inc. or Intel Singapore, Ltd.

September 1989
Order Number: 280007-005

FUNCTIONAL DESCRIPTION

Central Processing Unit

The central processor for the iSBC 86/XX* boards is Intel's 8086-2 CPU. A clock rate of 8 MHz is supported with a jumper selectable option of 5 MHz. The CPU architecture includes four 16-bit byte addressable data registers, two 16-bit memory base pointer registers and two 16-bit index registers, all accessed by a total of 24 operand addressing modes for comprehensive memory addressing and for support of the data structures required for today's structured, high level languages as well as assembly language.

NOTE:
iSBC 86/XX designates both the iSBC 86/14 and iSBC 86/30 CPU boards.

Instruction Set

The 8086 instruction repertoire includes variable length instruction format (including double operand instructions), 8-bit and 16-bit signed and unsigned arithmetic operators for binary, BCD and unpacked ASCII data, and iterative word and byte string manipulation functions.

For enhanced numerics processing capability, the iSBC 337A MULTIMODULE Numeric Data Processor extends the architecture and data set. Over 60 numeric instructions offer arithmetic, trigonometric, transcendental, logarithmic and exponential instructions. Supported data types include 16-, 32, and 64-bit integer, and 32- and 64-bit floating point, 18-digit packed BCD and 80-bit temporary.

Architectural Features

A 6-byte instruction queue provides pre-fetching of sequential instructions and can reduce the 750 nsec minimum instruction cycle to 250 nsec for queued instructions. The stack-oriented architecture readily supports modular programming by facilitating fast, simple, inter-module communication, and other programming constructs needed for asynchronous real-time systems. The memory expansion capabilities offer a 1 megabyte addressing range. The dynamic relocation scheme allows ease in segmentation of pure procedure and data for efficient memory utilization. Four segment registers (code, stack, data, extra) contain program loaded offset values which are used to map 16-bit addresses to 20-bit addresses. Each register maps 64K bytes at a time and activation of a specific register is controlled explicitly by program control and is also selected implicitly by specific functions and instructions.

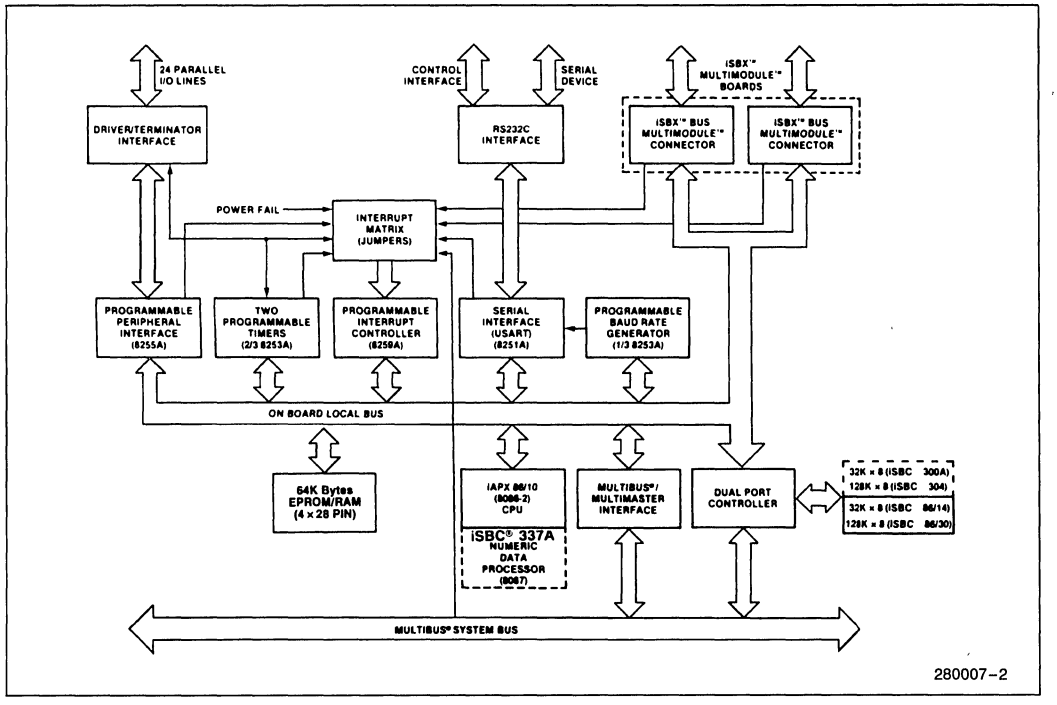


Figure 1. iSBC® 86/XX Block Diagram

RAM Capabilities

The iSBC 86/14 and iSBC 86/30 microcomputers contain 32K bytes and 128K bytes of dual-port dynamic RAM, respectively. In addition, on-board RAM may be doubled on each microcomputer by optionally adding RAM MULTIMODULE boards. The dual-port controller allows access to the on-board RAM (including RAM MULTIMODULE options) from the iSBC 86/XX boards and from any other MULTIBUS master via the system bus. Segments of on-board RAM may be configured as a private resource, protected from MULTIBUS system access.

EPROM Capabilities

Four 28-pin sockets are provided for a maximum on-board EPROM capacity is 64K bytes. Other JEDEC standard pinout devices are also supported, including byte-wide static RAMs.

Parallel I/O Interface

The iSBC 86/XX Single Board Computers contain 24 programmable parallel I/O lines implemented using the Intel 8255A Programmable Peripheral Interface. The system software is used to configure the I/O lines in any combination of unidirectional input/output and bidirectional ports. In order to take advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators, allowing the selection of the appropriate combination of optional line drivers and terminators with the required drive/termination characteristics. The 24 programmable I/O lines and signal ground lines are brought out to a 50-pin edge connector.

Serial I/O

A programmable communications interface using the Intel 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the iSBC 86/XX boards. A software selectable baud rate generator provides the USART with all common communication frequencies. The mode of operation (i.e., synchronous or asynchronous), data format, control character format, parity, and baud rate are all under program control. The 8251A provides full duplex, double buffered transmit and receive capability. Parity, overrun, and framing error detection are all incorporated in the USART. The RS232C command lines, serial data lines and signal ground line are brought out to a 26-pin edge connector.

Programmable Timers

The iSBC 86/XX boards provide three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8253 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate time intervals under software control. Routing for the outputs and gate/trigger inputs of two of these counters is jumper selectable. The outputs may be independently routed to the 8259A Programmable Interrupt Controller and to the I/O terminators associated with the 8255A to allow external devices or an 8255A port to gate the timer or to count external events. The third interval timer in the 8253 provides the programmable baud rate generator for the iSBC 86/XX boards' RS232C USART serial port. The system software configures each timer independently to select the desired function. The contents of each counter may be read at any time during system operation.

iSBX™ MULTIMODULE™ On-Board Expansion

Two 8/16-bit iSBX MULTIMODULE connectors are provided on the iSBC 86/XX microcomputers. Through these connectors, additional on-board I/O functions may be added. iSBX MULTIMODULE boards optimally support functions provided by VLSI peripheral components such as additional parallel and serial I/O, analog I/O, small mass storage device controllers and other custom interfaces to meet specific needs. By mounting directly on the single board computer, less interface logic, less power, simpler packaging, higher performance, and lower cost result when compared to other alternatives such as MULTIBUS form factor compatible boards. The iSBX connectors on the iSBC 86/XX boards provide all signals necessary to interface to the local on-board bus, including 16 data lines for maximum data transfer rates. iSBX MULTIMODULE boards designed with 8-bit data paths and using the 8-bit iSBX connector are also supported on the iSBC 86/XX microcomputers. A broad range of iSBX MULTIMODULE options are available from Intel. Custom iSBX modules may also be designed for use on the iSBC 86/XX boards.

Multimaster Capabilities

For those applications requiring additional processing capacity and the benefits of multiprocessing (i.e., several CPUs and/or controllers logically sharing system tasks through communication of the system

bus), the iSBC 86/XX boards provide full MULTIBUS arbitration control logic. This control logic allows up to three iSBC 86/XX boards or other bus masters, including iSBC 80 family MULTIBUS compatible 8-bit single board computers to share the system bus using a serial (daisy chain) priority scheme and allows up to 16 masters to share the MULTIBUS system bus with an external parallel priority decoder. In addition to the multiprocessing configurations made possible with multimaster capability, it also provides a very efficient mechanism for all forms of DMA (Direct Memory Access) transfers.

Interrupt Capability

The iSBC 86/XX boards provide 9 vectored interrupt levels. The highest level is the NMI (Non-Maskable Interrupt) line which is directly tied to the 8086 CPU. This interrupt is typically used for signaling catastrophic events (e.g., power failure). The Intel 8259A Programmable Interrupt Controller (PIC) provides control and vectoring for the next eight interrupt levels. A selection of four priority processing modes is available for use in designing request processing configurations to match system requirements for efficient interrupt servicing with minimal latencies. Operating mode and priority assignments may be reconfigured dynamically via software at any time during system operation. The PIC accepts interrupt requests from all on-board I/O resources and from the MULTIBUS system bus. The PIC then resolves requests according to the selected mode and, if appropriate, issues an interrupt to the CPU. Any combination of interrupt levels may be masked via software, by storing a single byte in the interrupt mask register of the PIC. In systems requiring additional interrupt levels, slave 8259A PICs may be interfaced via the MULTIBUS system bus, to generate additional vector addresses, yielding a total of 65 unique interrupt levels.

Interrupt requests to be serviced by the iSBC 86/XX boards may originate from 28 sources. All interrupt signals are brought to the interrupt jumper matrix where any combination of interrupt sources may be strapped to the desired interrupt request level on the 8259A PIC or the NMI input to the CPU directly.

Power-Fail Control and Auxiliary Power

Control logic is also included to accept a power-fail interrupt in conjunction with the AC-low signal generation capabilities to initiate an orderly shut down of the system in the event of a power failure. Additionally, an active-low TTL compatible memory protect signal is brought out on the auxiliary connector which, when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during system power-down sequences. An auxiliary power bus is also provided to allow separate power to RAM for systems requiring battery back-up of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

SPECIFICATIONS

Word Size

Instruction: 8, 16, 24, or 32 bits

Data: 8, 16 bits

System Clock

5.00 MHz or 8.00 MHz ±0.1% (jumper selectable)

Cycle Time

BASIC INSTRUCTION CYCLE

8 MHz: 750 ns
250 ns (assumes instruction in the queue)

5 MHz: 1.2 μs
400 ns (assumes instruction in the queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles).

Memory Cycle Time

RAM: 750 ns

EPROM: Jumper selectable from 500 ns to 875 ns

Memory Capacity/Addressing

ON-BOARD EPROM

Device	Total Capacity	Address Range
2716	8K bytes	FE000–FFFFF$_H$
2732A	16K bytes	FC000–FFFFF$_H$
2764	32K bytes	F8000–FFFFF$_H$
27128	64K bytes	F0000–FFFFF$_H$

ON-BOARD RAM

Board	Total Capacity	Address Range
iSBC 86/14	32K bytes	0–07FFF$_H$
iSBC 86/30	128K bytes	0–1FFFF$_H$

I/O Capacity

Parallel: 24 programmable lines using one 8255A

Serial: 1 programmable line using one 8251A

iSBX MULTIMODULE: 2 iSBX boards

Serial Communications Characteristics

Synchronous: 5–8 bits characters; internal or external character synchronization; automatic sync insertion

Asynchronous: 5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit direction

Interfaces

MULTIBUS: All signals TTL compatible

iSBX Bus: All signals TTL compatible

Parallel I/O: All signals TTL compatible

Serial I/O: RS232C compatible, configurable as a data set or data terminal

Timer: All signals TTL compatible

Interrupt Requests: All TTL compatible

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.70 in. (1.78 cm)
Weight: 14 oz (388 gm)

Environmental Characteristics

Operating Temperature: 0°C to 55°C
Relative Humidity: to 90% (without condensation)

Electrical Characteristics

DC POWER REQUIREMENTS

Configuration	Current Requirements (All Voltages ±5%)		
	+5V	+12V	−12V
Without EPROM[1]	5.1A	25 mA	23 mA
RAM only[2]	600 mA	—	—
With 8K EPROM[3] (using 2716)	5.4A	25 mA	23 mA
With 16K EPROM[3] (using 2732A)	5.5A	25 mA	23 mA
With 32K EPROM[3] (using 2764)	5.6A	25 mA	23 mA

NOTES:
1. Does not include power for optional ROM/EPROM, I/O drivers, and I/O terminators.
2. RAM chips powered via auxiliary power bus in power-down mode.
3. Includes power required for 4 ROM/EPROM chips, and I/O terminators installed for 16 I/O lines; all terminator inputs low.

Environmental Characteristics

Operating Temperature: 0°C to 55°C
Relative Humidity: to 90% (without condensation)

Reference Manual

144044-002: iSBC 86/14 and iSBC 86/30 Hardware Reference Manual (NOT SUPPLIED)

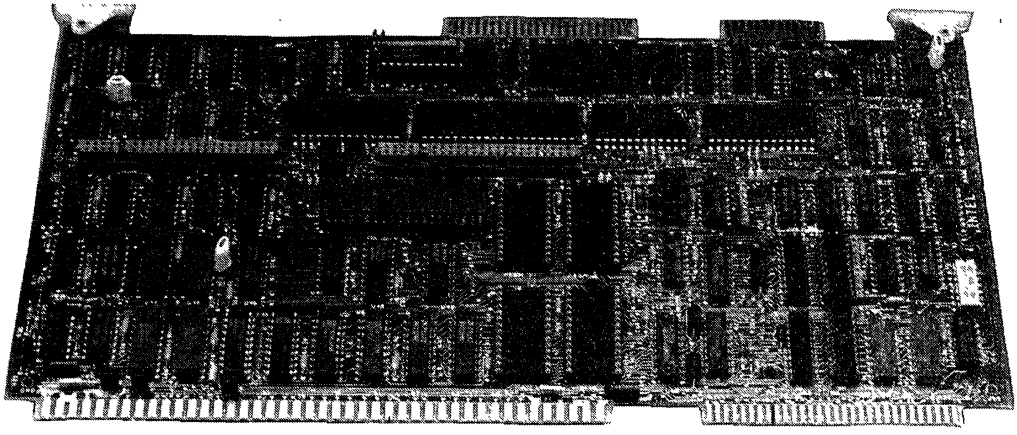
ORDERING INFORMATION

Part Number	Description
SBC 86/14	Single Board Computer
SBC 86/30	Single Board Computer

iSBC® 86/05A*
SINGLE BOARD COMPUTER

- 8086/10 (8086-2) Microprocessor with 5 or 8 MHz CPU Clock
- Software Compatible with 8086, 8088, 80186, 80286 Based 16-bit Single Board Computers
- Optional Numeric Data Processor with iSBC® 337 A MULTIMODULE™
- 8K bytes of Static RAM; Expandable On-Board to 16K Bytes
- Sockets for up to 256K Bytes of JEDEC 24/28-Pin Standard Memory Devices; Expandable On-Board to 512K Bytes
- Two iSBX™ Bus Connectors
- Programmable Synchronous/ Asynchronous RS232C Compatible Serial Interface with Software Selectable Baud Rate
- 24 Programmable Parallel I/O Lines
- Two Programmable 16-Bit BCD or Binary Timers/Event Counters
- 9 Levels of Vectored Interrupt Control, Expandable to 65 Levels
- MULTIBUS® Bus Interface for Multimaster Configurations and System Expansion

The iSBC 86/05A Single Board Computer is a member of Intel's complete line of OEM microcomputer systems which take full advantage of Intel's technology to provide economical, self-contained, computer-based solutions for OEM applications. The CPU, system clock, read/write memory, nonvolatile read only memory, I/O ports and drivers, serial communications interface, priority interrupt logic and programmable timers, all reside on the board. The large control storage capacity makes the iSBC 86/05A board ideally suited for control-oriented applications such as process control, instrumentation and industrial automation.



143325-1

*The iSBC® 86/05A is also manufactured under product code piSBC® 86/05A or siSBC® 86/05A by Intel Puerto Rico, Inc. or Intel Singapore, Ltd.

September 1989
Order Number: 143325-004

FUNCTIONAL DESCRIPTION

Central Processing Unit

The central processor for the iSBC 86/05A board is Intel's 8086-2 CPU. A clock rate of 8 MHz is supported with a jumper selectable option of 5 MHz. The CPU architecture includes four 16-bit byte addressable data registers, two 16-bit memory base pointer registers and two 16-bit index registers. All are accessed by a total of 24 operand addressing modes for comprehensive memory addressing and for support of the data structures required for today's structured, high level languages as well as assembly language.

Instruction Set

The 8086 instruction repertoire includes variable length instruction format (including double operand instructions), 8- and 16-bit signed and unsigned arithmetic operators for binary, BCD and unpacked ASCII data, and iterative word and byte string manipulation functions.

For enhanced numerics processing capability, the iSBC 337A MULTIMODULE Numeric Data Processor extends the architecture and data set. Over 60 numeric instructions offer arithmetic, trigonometric, transcendental, logarithmic and exponential instructions. Supported data types include 16-, 32-, and 64-bit integer, and 32- and 64-bit floating point, 18-digit packed BCD and 80-bit temporary.

Architectural Features

A 6-byte instruction queue provides pre-fetching of sequential instructions and can reduce the 740 ns minimum instruction cycle to 250 ns for queued instructions. The stack-oriented architecture readily supports modular programming by facilitating fast, simple, inter-module communication, and other programming constructs needed for asynchronous real-time systems. The memory expansion capabilities offer a 1 megabyte addressing range. The dynamic relocation scheme allows ease in segmentation of pure procedure and data for efficient memory utilization. Four segment registers (code, stack, data, extra) contain program loaded offset values which are used to map 16-bit addresses to 20-bit addresses. Each register maps 64K bytes at a time with activation of a specific register controlled explicity by program control and selected implicity by specific functions and instructions. All Intel languages support the extended memory capability, relieving the programmer of managing the megabyte memory space yet allowing explicit control when necessary.

Memory Configuration

The iSBC 86/05A microcomputer contains 8K bytes of high-speed 8K x 4 bit static RAM on-board. In addition, the above on-board RAM may be expanded to 16K bytes with the iSBC 302 MULTIMODULE RAM option which mounts on the iSBC 86/05A board. All on-board RAM is accessed by the 8086-2 CPU with no wait states, yielding a memory cycle time of 500 ns.

The iSBC 86/05A board also has four 28-pin, 8-bit wide (byte-wide) sockets, configured to accept JEDEC 24/28-pin standard memory devices. Up to 256K bytes of EPROM are supported in 64K byte increments with Intel 27512 EPROMs. The iSBC 86/05A board also supports 2K x 8, 4K x 8, 8K x 8, 16K x 8 and 32K x 8 EPROM memory devices. These sites also support 2K x 8 and 8K x 8 byte-wide static RAM (SRAM) devices and iRAM devices, yielding up to 32K bytes of SRAM in 8K byte increments on the baseboard.

When the addition of the iSBC 341 MULTIMODULE EPROM option, the on-board capacity for these devices is doubled, providing up to 512K bytes of EPROM and 64K bytes of byte-wide SRAM capacity on-board.

Parallel I/O Interface

The iSBC 86/05A Single Board Computer contains 24 programmable parallel I/O lines implemented using the Intel 8255A Programmable Peripheral Interface. The system software is used to configure the I/O lines in any combination of unidirectional input/output and bidirectional ports indicated in Table 1. In order to take advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators, allowing the selection of the appropriate combination of optional line drivers and terminators with the required drive/termination characteristics. The 24 programmable I/O lines and signal ground lines are brought out to a 50-pin edge connector.

Serial I/O

A programmable communications interface using the Intel 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the iSBC 86/05A board. A software selectable baud rate generator provides the USART with all common communication frequencies. The mode of operation (i.e., synchronous or asynchronous), data format, control character format, parity, and baud rate are all under program control. The 8251A provides full duplex, double buffered transmit and receive capability. Parity, overrun, and framing error detection are all

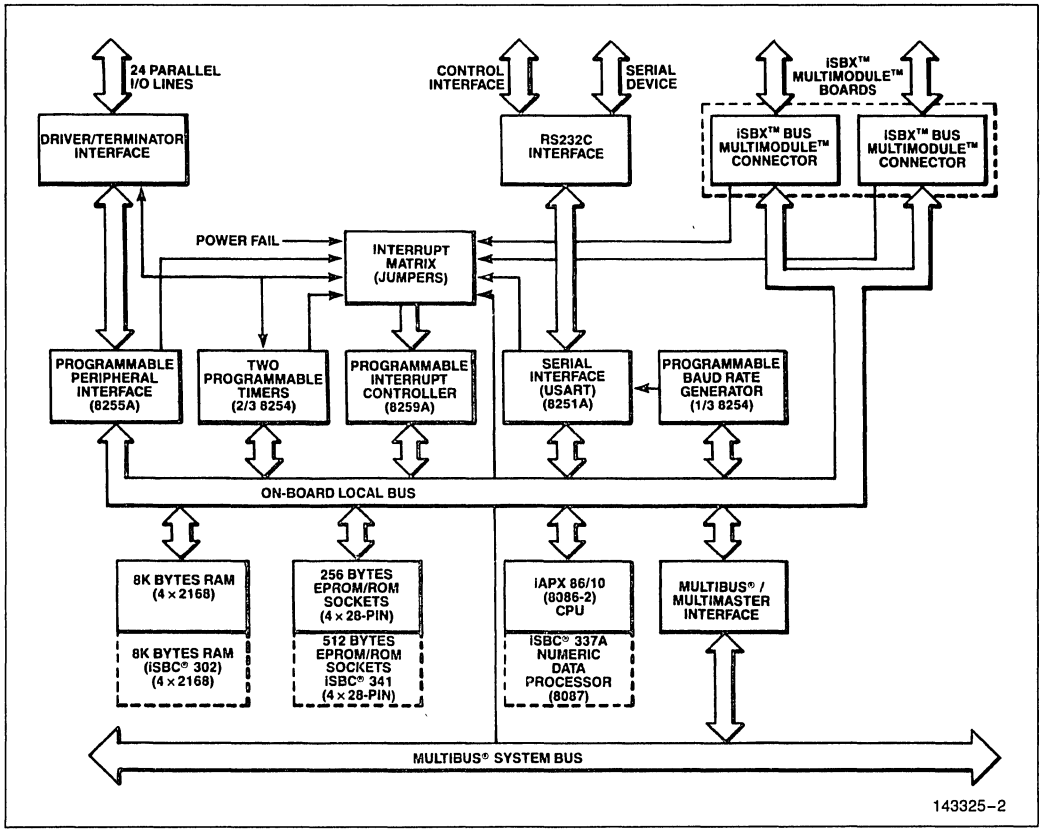


Figure 1. iSBC® 86/05A Block Diagram

incorporated in the USART. The RS232C compatible interface in conjunction with the USART, provides a direct interface to RS232C compatible terminals, cassettes, and asynchronous/synchronous modems. The RS232C command lines, serial data lines and signal ground line are brought out to a 26-pin edge connector.

Programmable Timers

The iSBC 86/05A board provides three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8254 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate time intervals under software control. Routing for the outputs and gate/trigger inputs of two of these counters is jumper selectable. The outputs may be independently routed to the 8259A Programmable Interrupt Controller and to the I/O terminators associated with the 8255A to allow external devices or an 8255A port to gate the timer or to count external events. The third interval timer in the 8254 provides the programmable baud rate generator for the iSBC 86/05A board RS232C USART serial port. The system software configures each timer independently to select the desired function. The contents of each counter may be read at any time during system operation.

iSBX™ MULTIMODULE™ On-Board Expansion

Two 8/16-bit iSBX MULTIMODULE connectors are provided on the iSBC 86/05A microcomputer. Through these connectors, additional on-board I/O and memory functions may be added. iSBX MULTIMODULE boards support functions such as additional parallel and serial I/O, analog I/O, mass storage device controllers BITBUS™ controllers, bubble memory, and other custom interfaces to meet specific needs. By mounting directly on the single board computer, less interface logic, less power, simpler

packaging, higher performance, and lower cost result when compared to other alternatives such as MULTIBUS form factor compatible boards. The iSBX connectors on the iSBC 86/05A board provide all signals necessary to interface to the local on-board bus, including 16 data lines for maximum data transfer rates. iSBX MULTIMODULE boards designed with 8-bit data paths and using the 8-bit iSBX connector are also supported on the iSBC 86/05A microcomputer. A broad range of iSBX MULTIMODULE options are available from Intel. Custom iSBX modules may also be designed for use on the iSBC 86/05A board.

Multimaster Capabilities

For those applications requiring additional processing capacity and the benefits of multiprocessing (i.e., several CPUs and/or controllers logically sharing system tasks through communication of the system bus), the iSBC 86/05A board provides full MULTIBUS arbitration control logic. This control logic allows up to three iSBC 86/05A boards or other bus masters to share the system bus using a serial (daisy chain) priority scheme and allows up to 16 masters to share the MULTIBUS system bus with an external parallel priority decoder. In addition to the multiprocessing configurations made possible with multimaster capability, it also provides a very efficient mechanism for all forms of DMA (Direct Memory Access) transfers.

Interrupt Capability

The iSBC 86/05A board provides 9 vectored interrupt levels. The highest level is the NMI (Non-Maskable Interrupt) line which is directly tied to the 8086 CPU. This interrupt is typically used for signaling catastrophic events (e.g., power failure). The Intel 8259A Programmable Interrupt Controller (PIC) provides control and vectoring for the next eight interrupt levels. A selection of four priority processing modes is available for use in designing request processing configurations to match system requirements for efficient interrupt servicing with minimal latencies. Operating mode and priority assignments may be reconfigured dynamically via software at any time during system operation. The PIC accepts interrupt requests from all on-board I/O resources and from the MULTIBUS system bus. The PIC then resolves requests according to the selected mode and, if appropriate, issues an interrupt to the CPU. Any combination of interrupt levels may be masked via software, by storing a single byte in the interrupt mask register of the PIC. In systems requiring additional interrupt levels, slave 8259A PICs may be interfaced via the MULTIBUS system bus, to generate additional vector addresses, yielding a total of 65 unique interrupt levels.

Interrupt requests to be serviced by the iSBC 86/05A board may originate from 24 sources. All interrupt signals are brought to the interrupt jumper matrix where any combination of interrupt sources may be strapped to the desired interrupt request level on the 8259A PIC or the NMI input to the CPU directly.

Power-Fail Control and Auxiliary Power

Control logic is also included, to accept a power-fail interrupt in conjunction with a power-supply having AC-low signal generation capabilities to initiate an orderly shut down of the system in the event of a power failure. Additionally, an active-low TTL compatible memory protect signal is brought out on the auxiliary connector which, when asserted, disables read/write access to RAM for systems requiring battery backup of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

SPECIFICATIONS

Word Size

Instruction: 8, 16, 24, or 32 bits
Data: 8, 16 bits

System Clock

5.00 MHz or 8.00 MHz ± 0.1% (jumper selectable)

Basic Instruction Cycle

At 8 MHz: 750 ns
250 ns (assumes instruction in the queue)
At 5 MHz: 1.2 sec.
400 ns (assumes instruction in the queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles).

Memory Cycle Time

500 ns cycle time (no wait states requires a memory component access time of 250 ns or less)
RAM: 500 ns
EPROM: Jumper selectable from 500 ns to 875 ns

Memory Capacity/Addressing

JEDEC 24/28 Pin Sites		
Device	**Total Capacity**	**Address Range**
2K × 8	8K bytes	FE000-FFFFF$_H$
4K × 8	16K bytes	FC000-FFFFF$_H$
8K × 8	32K bytes	F8000-FFFFF$_H$
16K × 8	64K bytes	F0000-FFFFF$_H$
32K × 8	128K bytes	E0000-FFFFF$_H$
64K × 8	256K bytes	C0000-FFFFF$_H$
With iSBC® 341 MULTIMODULE™ EPROM/SRAM		
Device	**Total Capacity**	**Address Range**
2K × 8	16K bytes	FC000-FFFFF$_H$
4K × 8	32K bytes	F8000-FFFFF$_H$
8K × 8	64K bytes	F0000-FFFFF$_H$
16K × 8	128K bytes	E0000-FFFFF$_H$
32K × 8	256K bytes	C0000-FFFFF$_H$
64K × 8	512K bytes	80000-FFFFF$_H$

ON-BOARD STATIC RAM

8K bytes — 0–1FFF$_H$

16K bytes— 0–3FFF$_H$ (with iSBC 302 MULTIMODULE Board)

I/O CAPACITY

PARALLEL — 24 programmable lines using one 8255A.

SERIAL — 1 programmable line using one 8251A.

iSBX MULTIMODULE— 2 iSBX single wide MULTIMODULE board or 1 iSBX double-width MULTIMODULE board.

SERIAL COMMUNICATIONS CHARACTERISTICS

SYNCHRONOUS — 5–8 bit characters; internal or external character synchronization; automatic sync insertion.

ASYNCHRONOUS— 5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit direction.

INTERFACES

MULTIBUS Bus: All signals TTL compatible

iSBX BUS Bus: All signals TTL compatible

PARALLEL I/O: All signals TTL compatible

SERIAL I/O: RS232C compatible, configurable as a data set or data terminal

TIMER: All signals TTL compatible

INTERRUPT REQUESTS: All TTL compatible

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.70 in. (1.78 cm)
Weight: 14 oz (388 gm)

ELECTRICAL CHARACTERISTICS

DC Power Requirements

Configuration	Current Requirements (All Voltages ±5%)		
	+5V	+12V	−12V
Without EPROM(1) RAM only(2)	4.7A 120 mA	25 mA	23 mA
With 8K EPROM(3) (using 2716)	5.0A	25 mA	23 mA
With 16K EPROM(3) (using 2732)	4.9A	25 mA	23 mA
With 32K EPROM(3) (using 2764)	4.9A	25 mA	23 mA

NOTES:
1. Does not include power for optional ROM/EPROM, I/O drivers, and I/O terminators.
2. RAM chips powered via auxiliary power bus in power-down mode.
3. Includes power required for 4 ROM/EPROM chips, and I/O terminators installed for 16 I/O lines; all terminator inputs low.

ENVIRONMENTAL CHARACTERISTICS

Operating Temperature: 0°C to 55°C

Relative Humidity: to 90% (without condensation)

REFERENCE MANUAL

Order no. 147162-002—*iSBC 86/05A Hardware Reference Manual* (NOT SUPPLIED)

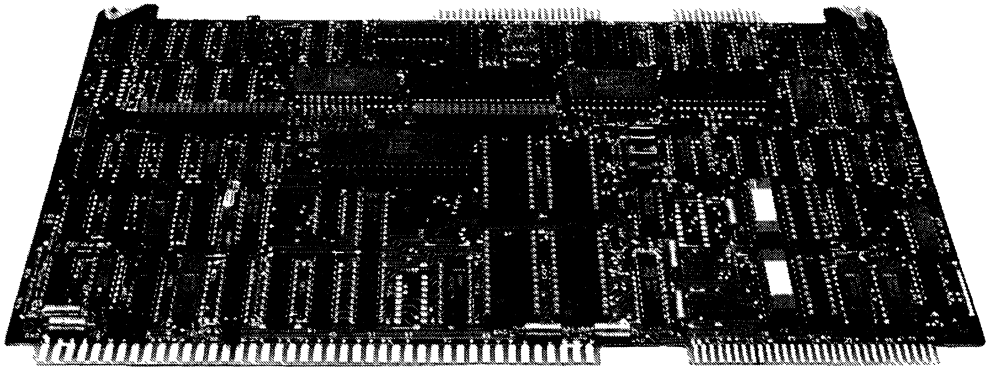
ORDER INFORMATION

Part Number	Description
SBC 86/05A	16-bit Single Board Computer with 8K bytes RAM

iSBC® 88/25*
SINGLE BOARD COMPUTER

- 8-Bit 8088 Microprocessor Operating at 5 MHz
- One Megabyte Addressing Range
- Two iSBX™ Bus Connectors
- Optional Numeric Data Processor with iSBC® 337A MULTIMODULE™
- 4K Bytes of Static RAM; Expandable On-Board to 16K Bytes
- Sockets for up to 64K Bytes of JEDEC 24/28-Pin Standard Memory Devices; Expandable On-Board to 128K Bytes
- Programmable Synchronous/ Asynchronous RS232C Compatible Serial Interface with Software Selectable Baud Rates
- 24 Programmable Parallel I/O Lines
- Two Programmable 16-Bit BCD or Binary Timers/Event Counters
- 9 Levels of Vectored Interrupt Control, Expandable to 65 Levels
- MULTIBUS® Interface for Multimaster Configurations and System Expansion

The iSBC 88/25 Single Board Computer is a member of Intel's complete line of OEM microcomputer systems which take full advantage of Intel's technology to provide economical, self-contained, computer-based solutions for OEM applications. The CPU, system clock, read/write memory, nonvolatile read only memory, I/O ports and drivers, serial communications interface, priority interrupt logic and programmable timers, all reside on the board. The large control storage capacity makes the iSBC 88/25 board ideally suited for control-oriented applications such as process control, instrumentation and industrial automation.



143847-1

The iSBC® 88/25 is also manufactured under product code piSBC® 88/25 or siSBC 88/25 by Intel Puerto Rico, Inc. or Intel Singapore, Ltd.

September 1989
Order Number: 143847-004

FUNCTIONAL DESCRIPTION

Central Processing Unit

The central processor for the iSBC 88/25 board is Intel's 8088 CPU operating at 5 MHz. The CPU architecture includes four 16-bit byte addressable data registers, two 16-bit memory base pointer registers and two 16-bit index registers, all accessed by a total of 24 operand addressing modes for comprehensive memory addressing and for support of the data structures required for today's structured, high level languages, as well as assembly language.

Instruction Set

The 8088 instruction repertoire includes variable length instruction format (including double operand instructions), 8-bit and 16-bit signed and unsigned arithmetic operators for binary, BCD and unpacked ASCII data, and iterative word and byte string manipulation functions.

For enhanced numerics processing capability, the iSBC 337A MULTIMODULE Numeric Data Processor extends the architecture and data set. Over 60 numeric instructions offer arithmetic, trigonometric, transcendental, logarithmic and exponential instructions. Supported data types include 16, 32, and 64-bit integer, and 32 and 64-bit floating point, 18-digit packed BCD and 80-bit temporary.

Architectural Features

A 4-byte instruction queue provides pre-fetching of sequential instructions and can reduce the 750 ns minimum instruction cycle to 250 ns for queued instructions. The stack-oriented architecture readily supports modular programming by facilitating fast, simple, inter-module communication, and other programming constructs needed for asynchronous real-time systems. The memory expansion capabilities

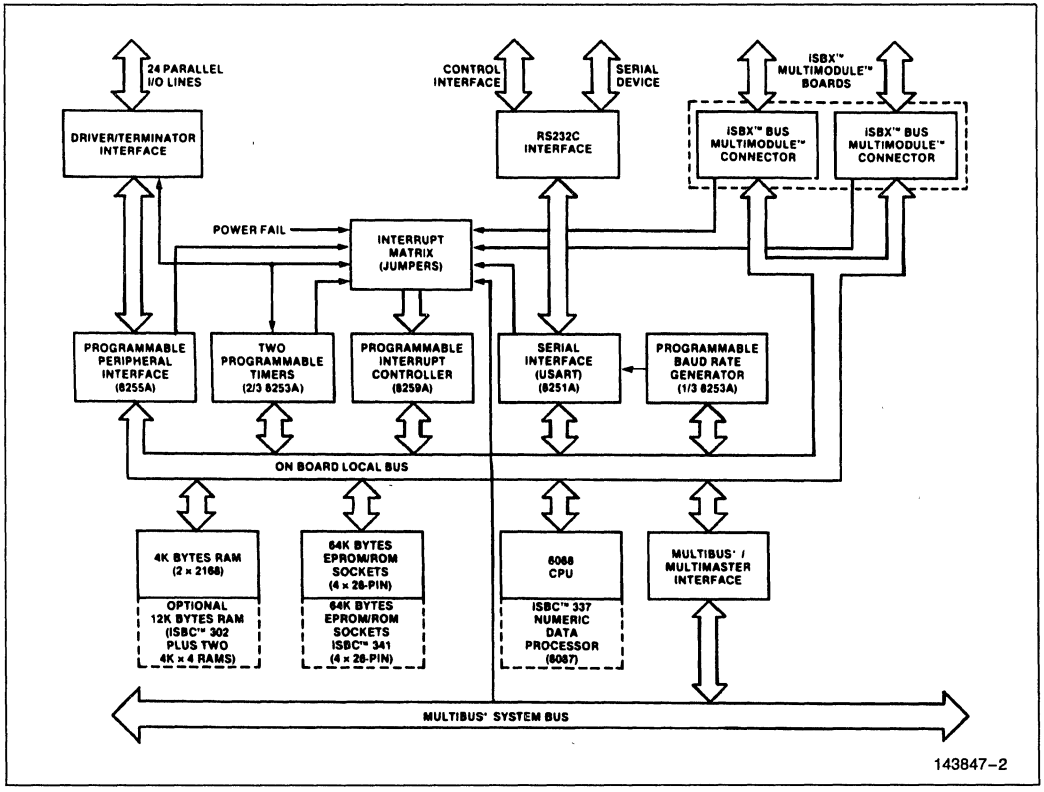


Figure 1. iSBC® 88/25 Block Diagram

offer a 1 megabyte addressing range. The dynamic relocation scheme allows ease in segmentation of pure procedure and data for efficient memory utilization. Four segment registers (code, stack, data, extra) contain program loaded offset values which are used to map 16-bit addresses to 20-bit addresses. Each register maps 64 Kbytes at a time and activation of a specific register is controlled explicitly by program control and is also selected implicitly by specific functions and instructions.

Memory Configuration

The iSBC 88/25 microcomputer contains 4 Kbytes of high-speed static RAM on-board. In addition, the on-board RAM may be expanded to 12 Kbytes via the iSBC 302 8 Kbyte RAM module which mounts on the iSBC 88/25 board and then to 16 Kbytes by adding two 4K × 4 RAM devices in sockets on the iSBC 302 module. All on-board RAM is accessed by the 8088 CPU with no wait states, yielding a memory cycle time of 800 ns.

In addition to the on-board RAM, the iSBC 88/25 board has four 28-pin sockets, configured to accept JEDEC 24/28-pin standard memory devices. Up to 64 Kbytes of EPROM are supported in 16 Kbyte increments with Intel 27128 EPROMs. The iSBC 88/25 board is also compatible with the 2716, 2732 and 2764 EPROMs.

With the addition of the iSBC 341 MULTIMODULE EPROM option, the on-board capacity for these devices is doubled, providing up to 128 Kbytes of EPROM capacity on-board.

Parallel I/O Interface

The iSBC 88/25 Single Board Computer contains 24 programmable parallel I/O lines implemented using the Intel 8255A Programmable Peripheral interface. The system software is used to configure the I/O lines in any combination of unidirectional input/output and bidirectional ports indicated in Table 1. In order to take advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators, allowing the selection of the appropriate combination of optional line drivers and terminators with the required drive/termination characteristics.

The 24 programmable I/O lines and signal ground lines are brought out to a 50-pin edge connector.

Serial I/O

A programmable communications interface using the Intel 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the iSBC 88/25 board. A software selectable baud rate generator provides the USART with all common communication frequencies. The mode of operation (i.e., synchronous or asynchronous), data format, control character format, parity and baud rate are all under program control. The 8251A provides full duplex, double buffered transmit and receive capability. Parity, overrun and framing error detection are all incorporated in the USART. The RS232C compatible interface on each board, in conjunction with the USART, provides a direct interface to RS232C compatible terminals, cassettes and asynchronous and synchronous modems. The RS232C command lines, serial data lines and signal ground line are brought out to a 26-pin edge connector.

Table 1. Input/Output Port Modes of Operation

Port	Lines (qty)	Mode of Operation					
		Unidirectional				Bidirectional	Control
		Input		Output			
		Latched	Latched & Strobed	Latched	Latched & Strobed		
1	8	X	X	X	X	X	
2	8	X	X	X	X		
3	4	X		X			X(1)
	4	X		X			X(1)

NOTE:
1. Part of port 3 must be used as a control port when either port 1 or port 2 are used as a latched and strobed input or a latched and strobed output port or port 1 is used as a bidirectional port.

Programmable Timers

The iSBC 88/25 board provides three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8253 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate time intervals under software control. Routing for the outputs and gate/trigger inputs of two of these counters is jumper selectable.

The outputs may be independently routed to the 8259A Programmable Interrupt Controller and to the I/O terminators associated with the 8255A to allow external devices or an 8255A port to gate the timer or to count external events. The third interval timer in the 8253 provides the programmable baud rate generator for the iSBC 88/25 board RS232C USART serial port. The system software configures each timer independently to select the desired function.

iSBX™ MULTIMODULE™ On-Board Expansion

Two 8-bit iSBX MULTIMODULE connectors are provided on the iSBC 88/25 microcomputer. Through these connectors, additional on-board I/O functions such as additional parallel and serial I/O, analog I/O, mass storage device controllers and other custom interfaces may be added. By mounting directly on the single board computer, less interface logic, less power, simpler packaging, higher performance, and lower cost result when compared to other alternatives such as MULTIBUS form factor compatible boards. The iSBX connectors on the iSBC 88/25 provide all signals necessary to interface to the local on-board bus. A broad range of iSBX MULTIMODULE options are available from Intel. Custom iSBX modules may also be designed for use on the iSBC 88/25 board.

Multimaster Capabilities

For those applications requiring additional processing capacity and the benefits of multiprocessing (i.e., several CPUs and/or controllers logically sharing system tasks through communication of the system bus), the iSBC 88/25 board provides full MULTIBUS arbitration control logic. This control logic allows up to three iSBC 88/25 boards or other bus masters, including iSBC 80 and iSBC 86 family MULTIBUS compatible single board computers to share the system bus using a serial (daisy chain) priority scheme and allows up to 16 masters to share the MULTIBUS system bus with an external parallel priority decoder. In addition to the multiprocessing configurations made possible with multimaster capability, it also provides a very efficient mechanism for all forms of DMA (Direct Memory Access) transfers.

Interrupt Capability

The iSBC 88/25 board provides 9 vectored interrupt levels. The highest level is the NMI (Non-Maskable Interrupt) line which is directly tied to the 8088 CPU. This interrupt is typically used for signaling catastrophic events (e.g., power failure). The Intel 8259A Programmable Interrupt Controller (PIC) provides control and vectoring for the next eight interrupt levels. A selection of four priority processing modes is available for use in designing request processing configurations to match system requirements for efficient interrupt servicing with minimal latencies. Operating mode and priority assignments may be reconfigured dynamically via software at any time during system operation. The PIC accepts interrupt requests from all on-board I/O resources and from the MULTIBUS system bus. The PIC then resolves requests according to the selected mode and, if appropriate, issues an interrupt to the CPU.

Interrupt requests to be serviced by the iSBC 88/25 board may originate from 24 sources. All interrupt signals are brought to the interrupt jumper matrix where any combination of interrupt sources may be strapped to the desired interrupt request level on the 8259A PIC or the NMI input to the CPU directly.

Power-Fail Control and Auxiliary Power

Control logic is also included to accept a power-fail interrupt to initiate an orderly shut down of the system in the event of a power failure. Additionally, an active-low TTL compatible memory protect signal is brought out of the auxiliary connector which, when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during system power-down sequences. An auxiliary power bus is also provided to allow separate power to RAM for systems requiring battery backup of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

SPECIFICATIONS

Word Size

Instruction—8, 16, 24, or 32 bits
Data—8 bits

System Clock

5.00 MHz or 4.17 MHz ±0.1% (jumper selectable)

NOTE:
4.17 MHz required with the optional iSBC 337 module.

Cycle Time

BASIC INSTRUCTION CYCLE

At 5 MHz—1.2 μs
—400 ns (assumes instruction in the queue)

NOTES:
Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles).

Memory Cycle Time

RAM—800 ns (no wait states)
EPROM—Jumper selectable from 800 ns to 1400 ns

Memory Capacity/Addressing

ON-BOARD EPROM

Device	Total Capacity	Address Range
2716	8 Kbytes	$FE000\text{-}FFFFF_H$
2732	16 Kbytes	$FC000\text{-}FFFFF_H$
2764	32 Kbytes	$F8000\text{-}FFFFF_H$
27128	64 Kbytes	$F0000\text{-}FFFFF_H$

WITH iSBC 341 MULTIMODULE EPROM

Device	Total Capacity	Address Range
2716	16 Kbytes	$FC000\text{-}FFFFF_H$
2732	32 Kbytes	$F8000\text{-}FFFFF_H$
2764	64 Kbytes	$F0000\text{-}FFFFF_H$
27128	128 Kbytes	$E0000\text{-}FFFFF_H$

ON-BOARD RAM

4 Kbytes—$0\text{-}0FFF_H$

WITH iSBC 302 MULTIMODULE RAM

12 Kbytes—$0\text{-}2FFF_H$

WITH iSBC 302 MULTIMODULE BOARD AND TWO 4K x 4 RAM CHIPS

16 Kbytes—$0\text{-}3FFF_H$

I/O Capacity

Parallel—24 programmable lines using one 8255A
Serial—1 programmable line using one 8251A
iSBX Multimodule—2 iSBX MULTIMODULE boards

Serial Communications Characteristics

Synchronous—5 8-bit characters; internal or external character synchronization; automatic sync insertion
Asynchronous—5 8-bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection

Interfaces

Multibus: All signals TTL compatible

iSBX Bus: All signals TTL compatible

Parallel I/O: All signals TTL compatible

Serial I/O: RS232C compatible, configurable as a data set or data terminal

Timer: All signals TTL compatible

Interrupt Requests: All TTL compatible

Physical Characteristics

Width: 12.00 in. (30.48 cm)

Height: 6.75 in. (17.15 cm)

Depth: 0.70 in. (1.78 cm)

Weight: 14 oz. (388 gm)

Electrical Characteristics

DC POWER REQUIREMENTS

Configuration	Current Requirements (All Voltages ±5%)		
	+5V	+12V	−12V
Without EPROM(1)	3.8A	25 mA	23 mA
RAM only(2)	104 mA		
With 8K EPROM(3) (using 2716)	4.3A	25 mA	23 mA
With 16K EPROM(3) (using 2732)	4.4A	25 mA	23 mA
With 32K EPROM(3) (using 2764)	4.4A	25 mA	23 mA

NOTES:
1. Does not include power for optional ROM/EPROM, I/O drivers and I/O terminators.
2. RAM chips powered via auxiliary power bus in power-down mode. Does not include power for optional RAM.
3. Includes power required for 4 ROM/EPROM chips, and I/O terminators installed for 16 I/O lines; all terminator inputs low.

Environmental Characteristics

Operating Temperature: 0°C to 55°C

Relative Humidity: to 90% (without condensation)

Reference Manual

143825-001—iSBC 88/25 Hardware Reference Manual (NOT SUPPLIED)

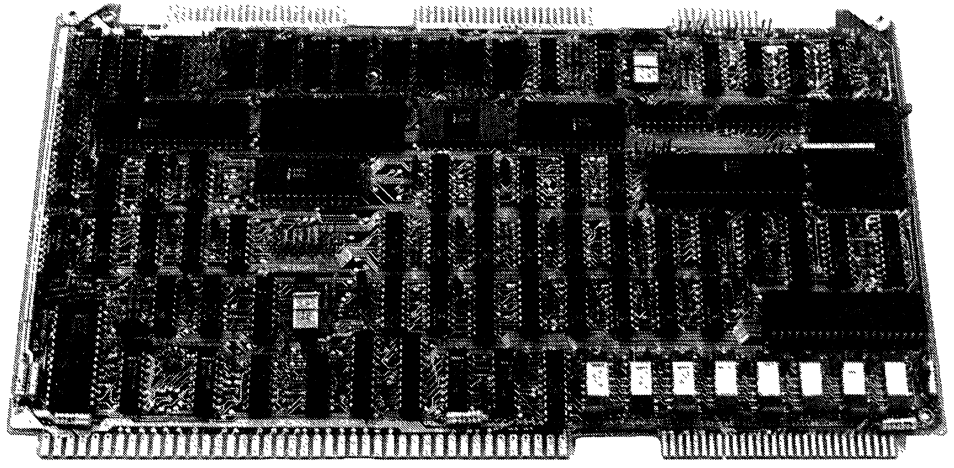
ORDERING INFORMATION

Part Number	Description
SBC 88/25	8-bit Single Board Computer with 4 Kbytes RAM

iSBC® 80/30*
SINGLE BOARD COMPUTER

- 8085A Central Processing Unit
- 16K Bytes of Dual Port Dynamic Read/ Write Memory with On-Board Refresh
- Sockets for up to 8K Bytes of Read Only Memory
- Sockets for 8041A/8741A Universal Peripheral Interface and Interchangeable Line Drivers and Line Terminators
- 24 Programmable Parallel I/O Lines with Sockets for Interchangeable Line Drivers and Terminators
- Full MULTIBUS® Control Logic
- Programmable Synchronous/ Asynchronous RS232C Compatible Serial Interface with Fully Software Selectable Baud Rate Generation
- 12 Levels of Programmable Interrupt Control
- Two Programmable 16-Bit BCD or Binary Counters
- Auxiliary Power Bus, Memory Protect, and Power-Fail Interrupt Control Logic for RAM Battery Backup

The iSBC 80/30 Single Board Computer is a member of Intel's complete line of OEM computer systems which take full advantage of Intel's LSI technology to provide economical self-contained computer-based solutions for OEM applications. The CPU, system clock, read/write memory, nonvolatile read only memory, universal peripheral interface capability, I/O ports and drivers, serial communications interface, priority interrupt logic, programmable timers, MULTIBUS control logic, and bus expansion drivers all reside on the board.



280219-1

*The iSBC® 80/30 board is also manufactured under product code piSBC® 80/30 and siSBC® 80/30 by Intel Puerto Rico, Inc. and Intel Singapore, Ltd.

September 1989
Order Number: 280219-003

FUNCTIONAL DESCRIPTION

Central Processing Unit

Intel's 8-bit n-channel 8085A CPU, fabricated on a single LSI chip, is the central processor for the iSBC 80/30. The 8085A CPU is directly software compatible with the Intel 8080A CPU. The 8085A contains six 8-bit general purpose registers and an accumulator. The six general purpose registers may be addressed individually or in pairs, providing both single and double precision operators. The minimum instruction execution time is 1.45 microseconds. The 8085A CPU has a 16-bit program counter. An external stack, located within any portion of iSBC 80/30 read/write memory, may be used as a last-in/first-out storage area for the contents of the program counter, flags, accumulator, and all of the six general purpose registers. A 16-bit stack pointer controls the addressing of this eternal stack. This stack provides subroutine nesting bounded only by memory size.

Bus Structure

The iSBC 80/30 has an internal bus for all on-board memory and I/O operations and a system bus (i.e., the MULTIBUS) for all external memory and I/O operations. Local (on-board) operations do not tie up the system bus, and allow true parallel processing when several bus masters (i.e., DMA devices, other single board computers) are used in a multimaster scheme. A block diagram of the iSBC 80/30 functional components is shown in Figure 1.

RAM Capacity

The iSBC 80/30 contains 16K bytes of dynamic read/write memory. All RAM read and write operations are performed at maximum processor speed. Power for the on-board RAM may be provided on an auxiliary power bus, and memory protect logic is included for RAM battery backup requirements. The iSBC 80/30 contains a dual port controller, which provides dual port capability for the on-board RAM memory. RAM accesses may occur from either the iSBC 80/30 or from any other bus master interfaced via the MULTIBUS. Since on-board RAM accesses do not require the MULTIBUS, the bus is available for any other concurrent operations (e.g., DMA data transfers) requiring the use of the MULTIBUS. Dynamic RAM refresh is accomplished automatically ther the CPU or via the MULTIBUS. Memory space assignment can be selected independently for on-board and MULTIBUS RAM accesses. The on-board RAM, as seen by the 8085A CPU, may be placed anywhere within the 0- to 64K-address space. The iSBC 80/30 provides extended addressing jumpers to allow the on-board RAM to reside within a one megabyte address space when accessed via the MULTIBUS. In addition, jumper options are provided which allow the user to reserve 8K- and 16K-byte segments of on-board RAM for use by the 8085A CPU only. This reserved RAM space is not accessible via the MULTIBUS and does not occupy any system address space.

EPROM/ROM Capacity

Sockets for up to 8K bytes of nonvolatile read only memory and provided on the iSBC 80/30 board.

Parallel I/O Interface

The iSBC 80/30 contains 24 programmable parallel I/O lines implemented using the Intel 8255A Programmable Peripharal Interface. The system software is used to configure the I/O lines in any combination of unidirectional input/output and bidirectional ports. Therefore, the I/O interface may be customized to meet specific peripheral requirements. In order to take full advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators. Hence, the flexibility of the I/O interface is further enhanced by the capability of selecting the appropriate combination of optional line drivers and terminators to provide the required sink current, polarity, and drive/termination characteristics for each application. The 24 programmable I/O lines and signal ground lines are brought out to a 50-pin edge connector that mates with flat, woven, or round cable.

Universal Peripheral Interface (UPI)

The iSBC 80/30 provides sockets for a user supplied Intel 8041A/8741A Universal Peripheral Interface (UPI) chip and the associated line drivers and terminators for the UPI's I/O ports. The 8041A/8741A is a single chip microcomputer containing a CPU, 1K bytes of ROM (8041A) or EPROM (8741A), 64 bytes of RAM, 18 programmable I/O lines, and an 8-bit timer. Special interface registers included in the chip allow the 8041A to function as a

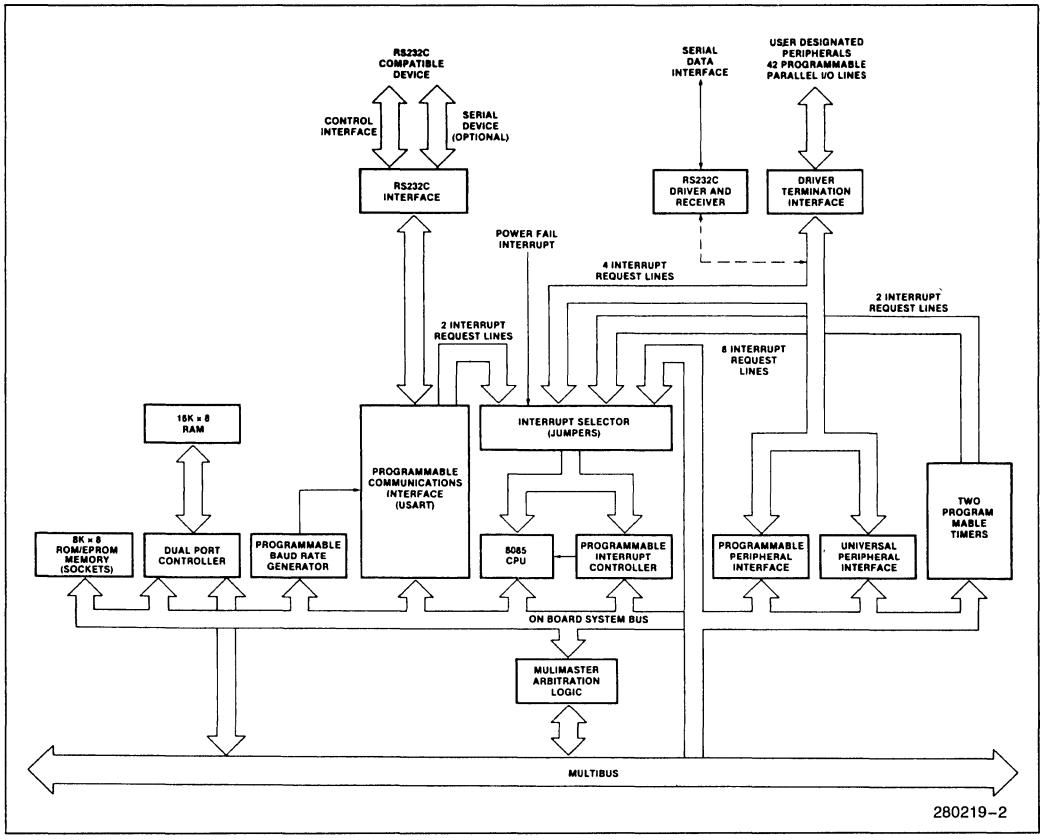


Figure 1. iSBC® 80/30 Single Board Computer Block Diagram

slave processor to the iSBC 80/30's 8085A CPU. The UPI allows the user to specifiy algorithms for controlling user peripherals directly in the chip, thereby relieving the 8085A for other system functions. The iSBC 80/30 provides an RS232C driver and an RS232C receiver for optional connection to the 8041A/8741A in applications where the UPI is programmed to handle simple serial interfaces.

Serial I/O

A programmable communications interface using the Intel 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the iSBC 80/30. A software selectable baud rate generator provides the USART with all common communication frequencies. The USART can be programmed by the system software to select the desired asynchronous or synchronous serial data transmission technique (including IBM By-Sync). The mode of operation (i.e., synchronous or asynchronous), data format, control character format, parity, and baud rate are all under program control. The 8251A provides full duplex, double buffered transmit and receive capability. Parity, overrun, and framing error detection are all incorporated in the USART. The RS232C compatible interface on each board, in conjunction with the USART, provides a direct interface to RS232C compatible terminals, cassettes, and asynchronous and synchronous modems. The RS232C command lines, serial data lines, and signal ground line are brought out to a 26-pin edge connector that mates with RS232C compatible flat or round cable.

Multimaster Capability

The iSBC 80/30 is a full computer on a single board with resources capable of supporting a great variety of OEM system requirements. For those applications requiring additional processing capacity and the benefits of multiprocessing (i.e., several CPUs and/or controllers logically sharing system tasks through communication over the system bus), the iSBC 80/30 provides full MULTIBUS arbitration control logic. This control logic allows up to three iSBC 80/

30's or other bus masters to share the system bus in serial (daisy chain) priority fashion, and up to 16 masters to share the MULTIBUS with the addition of an external priority network. The MULTIBUS arbitration logic operates synchronously with a MULTIBUS clock (provided by the iSBC 80/30 or optionally connected directly to the MULTIBUS clock) while data is transferred via a handshake between the master and slave modules. This allows different speed controllers to share resources on the same bus, and transfer via the bus proceed asynchronously. Thus, transfer speed is dependent on transmitting and receiving devices only. This design prevents slow master modules from being handicapped in their attempts to gain control of the bus, but does not restrict the speed at which faster modules can transfer data via the same bus. The most obvious applications for the master-slave capabilities of the bus are multiprocessor configurations, high speed direct memory access (DMA) operations, and high speed peripheral control, but are by no means limited to these three.

Programmable Timers

The iSBC 80/30 provides three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8253 Programmable Interval Timer. Each counter is capabile of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate time intervals under software control. Routing for the outputs and gate/trigger inputs of two of these counters is jumper selectable. The outputs may be independently routed to the 8259A Programmable Interrupt Controller, to the I/O line drivers associated with the 8255A Programmable Peripheral Interface, and to the 8041A/8741A Universal Programmable Interface, or may be routed as inputs to the 8255A and 8041A/8741A chips. The gate/trigger inputs may be routed to I/O terminators associated with the 8255A or as output connections from the 8255A. The third interval timer in the 8253 provides the programmable baud rate generator for the iSBC 80/30 RS232C USART serial port. In utilizing the iSBC 80/30, the systems designer simply configures, via software, each timer independently to meet system requirements.

Interrupt Capability

The iSBC 80/30 provides vectoring for 12 interrupt levels. Four of these levels are handled directly by the interrupt processing capability of the 8085A CPU and represent the four highest priority interrupts of the iSBC 80/30. Requests are routed to the 8085A interrupt inputs, TRAP, RST 7.5, RST 6.5, and RST 5.5 (in decreasing order of priority) and each input generates a unique memory address (TRAP: 24H; RST 7.5: 3CH; RST 6.5: 34H; and RST 5.5: 2CH). An 8085A jump instruction at each of these addresses then provides linkage to interrupt service routines located independently anywhere in memory. All interrupt inputs with the exception of the trap interrupt may be masked via software. The trap interrupt should be used for conditions such as power-down sequences which require immediate attention by the 8085A CPU. The Intel 8259A Programmable Interrupt Controller (PIC) provides vectoring for the next eight interrupt levels. Operating mode and priority assignments may be reconfigured dynamically via software at any time during system operation. The PIC accepts interrupt requests from the programmable parallel and serial I/O interfaces, the programmable timers, the system bus, or directly from peripheral equipment.

Interrupt requests may originate from 18 sources. Two jumper selectable interrupt requests can be automatically generated by the programmable peripheral interface when a byte of information is ready to be transferred to the CPU (i.e., input buffer is full) or a byte of information has been transferred to a peripheral device (i.e., output buffer is empty). Two jumper selectable interrupt requests can be automatically generated by the USART when a character is ready to be transferred to the CPU (i.e., receive channel buffer is full), or a character is ready to be transmitted (i.e., transmit channel data buffer is empty). A jumper selectable request can be generated by each of the programmable timers and by the universal peripheral interface, eight additional interrupt request lines are available to the user for direct interface to user designated peripheral devices via the system bus, and two interrupt request lines may be jumper routed directly from peripherals via the parallel I/O driver/terminator section.

Power-Fail Control

Control logic is also included to accept a power-fail interrupt in conjunction with the AC-low signal from the iSBC 635 Power Supply or equivalent.

SPECIFICATIONS

Word Size

Instruction: 8, 16, or 24 bits
Data: 8 bits

Cycle Time

Basic Instruction Cycle: 1.45 μs

NOTE:
Basic instruction cycle is defined as the fastest instruction (i.e., four clock cycles).

Memory Addressing

On-Board ROM/EPROM: 0-07FF (using 2708 or 2758 EPROMs); 0-0FFF (using 2716 EPROMs); 0-1FFF (using 2716 EPROMs; 0-1FFF (using 2732 EPROMs).

On-Board RAM: 16K bytes of dual port RAM starting on a 16K boundary. One or two 8 K-byte segments may be reserved for CPU use only.

Memory Capacity

On-Board Read Only Memory: 8K bytes (sockets only)
On-Board RAM: 16K bytes

I/O Capacity

Parallel: 42 programmable lines using one 8255A (24 I/O lines) and an optional 8041A/8741A (18 I/O lines)

Serial: 2 programmable lines using one 8251A and an optional 8041A/8741A programmed for serial operation

Serial Communications Characteristics

Synchronous: 5–8 bit characters; internal or external character synchronization; automatic sync insertion.

Asynchronous: 5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection.

Interfaces

MULTIBUS: All signals TTL compatible
Parallel I/O: All signals TTL compatible
Interrupt Requests: All TTL compatible
Timer: All signals TTL compatible
Serial I/O: RS232C compatible, data set configuration

System Clock (8085A CPU)

2.76 MHz ±0.1%

Auxiliary Power

An auxiliary power bus is provided to allow separate power to RAM for systems requiring battery backup of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

Memory Protect

An active-low TTL compatible memory protect signal is brought out on the auxiliary connector which, when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during system power-down sequences.

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.50 in. (1.27 cm)
Weight: 18 oz. (509.6 gm)

Electrical Characteristics

DC POWER REQUIREMENTS

Configuration	Current Requirements			
	V_{CC} = +5V ±5% (max)	V_{DD} = +12V ±5% (max)	V_{BB} = −5V ±5% (max)	V_{AA} = −12V ±5% (max)
Without EPROM(1)	I_{CC} = 3.5A	I_{DD} = 220 mA	I_{BB} = −	I_{AA} = 50 mA
With 8041/8741(2)	3.6A	220 mA	—	50 mA
RAM only(3)	350 mA	20 mA	2.5 mA	—

NOTES:
1. Does not include power required for optional EPROM/ROM, 8041A/8741A I/O drivers, and I/O terminators.
2. Does not include power required for optional EPROM/ROM. I/O drivers and I/O terminators.
3. RAM chips powered via auxiliary power bus.

Environmental Characteristics

Operating Temperature: 0°C to 55°C

Reference Manual

9800611B— iSBC 80/30 Single Board Computer Hardware Reference Manual (NOT SUPPLIED)

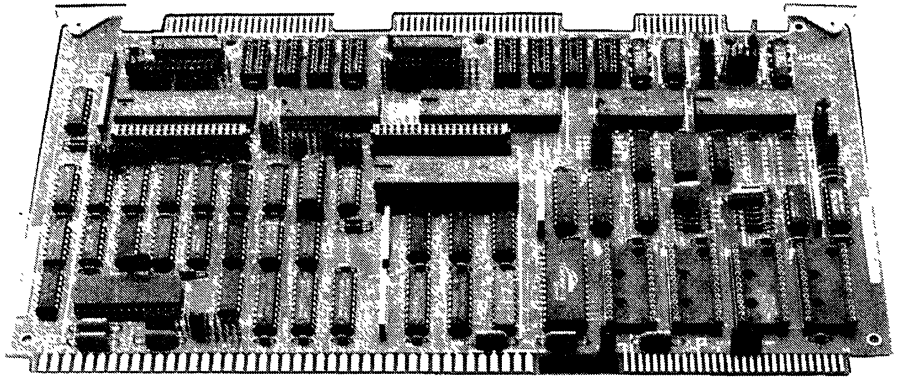
ORDERING INFORMATION

Part Number	Description
SBC 80/30	Single Board Computer with 16K bytes RAM

iSBC® 80/24A*
SINGLE BOARD COMPUTER

- **Upward Compatible Replacement for iSBC 80/20-4 Single Board Computer**
- **8085A-2 CPU Operating at 4.8 or 2.4 MHz**
- **Two iSBX™ Bus Connectors for iSBX MULTIMODULE™ Board Expansion**
- **8K Bytes of Static Read/Write Memory**
- **Sockets for Up to 32K Bytes of Read Only Memory**
- **48 Programmable Parallel I/O Lines with Sockets for Interchangeable Line Drivers and Terminators**
- **Programmable Synchronous/ Asynchronous RS232C Compatible Serial Interface with Software Selectable Baud Rates**
- **Full MULTIBUS® Control Logic for Multimaster Configurations and System Expansion**
- **Two Programmable 16-Bit BCD or Binary Timers/Event Counters**
- **12 Levels of Programmable Interrupt Control**
- **Auxiliary Power Bus, Memory Protect, and Power-Fail Interrupt Control Logic Provided for Battery Backup RAM Requirements**

The Intel 80/24A Single Board Computer is a member of Intel's complete line of OEM microcomputer systems which take full advantage of Intel's LSI technology to provide economical, self-contained computer-based solutions for OEM applications. The CPU, system clock, iSBX bus interface, read/write memory, read only memory sockets, I/O ports and drivers, serial communications interface, priority interrupt logic, and programmable timers all reside on the board. Full MULTIBUS interface logic is included to offer compatibility with the Intel OEM Microcomputer Systems family of Single Board Computers, expansion memory options, digital and analog I/O expansion boards, and peripheral and communications controllers.



142927-1

*The iSBC® 80/24A Board is also manufactured under product code piSBC® 80/24A and siSBC® 80/24A by Intel Puerto Rico, Inc, and Intel Singapore, Ltd.

October 1989
Order Number: 142927-005

FUNCTIONAL DESCRIPTION

Central Processing Unit

Intel's 8-bit 8085A-2 CPU is the central processor for the iSBC 80/24A board operating at either 4.8 or 2.4 MHz. The 8085A-2 CPU is directly software compatible with the Intel 8080A CPU. The 8085A-2 contains six 8-bit general purpose registers and an accumulator. The six general purpose registers may be addressed individually or in pairs, providing single and double precision operators. Minimum instruction execution time is 826 nanoseconds.

iSBX™ MULTIMODULE™ On-Board Expansion

Two 8-bit iSBX bus MULTIMODULE connectors are provided for plug-in expansion of iSBX MULTIMODULE boards.

Memory Addressing

The 8085A-2 has a 16-bit program counter which allows direct addressing of up to 64K bytes of memory. An external stack, located within any portion of read/write memory, may be used as a last-in/first-out storage area for the contents of the program counter, flags, accumulator, and all of the six general purpose registers. A 16-bit stack pointer controls the addressing of this external stack. This stack provides subroutine nesting bounded only by memory size.

Memory Capacity

The iSBC 80/24A board contains 8K bytes of static read/write memory using an 8K × 8 SRAMs. All RAM read and write operations are performed at maximum processor speed. Power for the on-board RAM may be provided on an auxiliary power bus, and memory protect logic is included for RAM battery backup requirements.

Four sockets are provided for up to 32K bytes of nonvolatile read only memory on the iSBC 80/24A board.

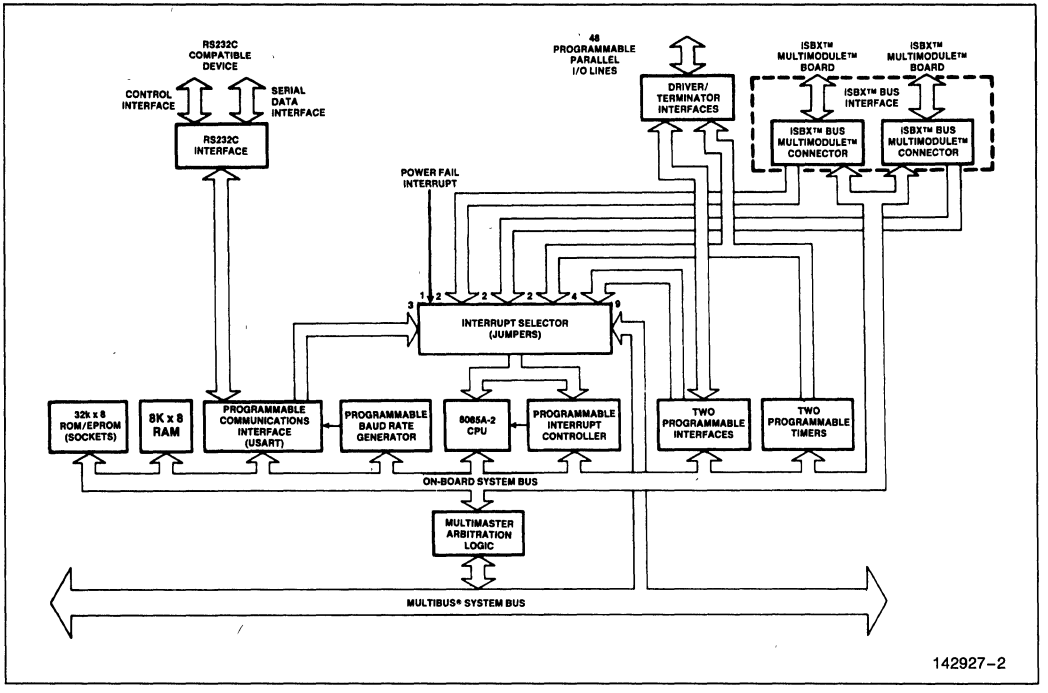


Figure 1. iSBC® 80/24A Single Board Computer Block Diagram

Parallel I/O Interface

The iSBC 80/24A board contains 48 programmable parallel I/O lines implemented using two Intel 8255A Programmable Peripheral Interfaces. The system software is used to configure the I/O lines in any combination of unidirectional input/output and bidirectional ports. Therefore, the I/O interface may be customized to meet specific peripheral requirements. In order to take full advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators. Hence, the flexibility of the I/O interface is further enhanced by the capability of selecting the appropriate combination of optional line drivers and terminators to provide the required sink current, polarity, and drive/termination characteristics for each application. The 48 programmable I/O lines and signal ground lines are brought out to two 50-pin edge connectors that mate with flat, woven, or round cables.

Serial I/O Interface

A programmable communications interface using the Intel 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the iSBC 80/24A board. A software selectable baud rate generator provides the USART with all common communication frequencies. The USART can be programmed by the system software to select the desired asynchronous or synchronous serial data transmission technique (including IBM Bi-Sync). The mode of operation (i.e. synchronous or asynchronous), data format, control character format, parity, and baud rate are all under program control. The 8251A provides full duplex, double buffered transmit and receive capability. Parity, overrun, and framing error detection are all incorporated in the USART. The RS232C compatible interface, in conjunction with the USART, provides a direct interface to RS232C compatible terminals, cassettes, and asynchronous and synchronous modems. The RS232C command lines serial data lines, and signal ground line are brought out to a 26-pin edge connector that mates with RS232C compatible flat or round cable.

Multimaster Capability

For those applications requiring additional processing capacity and the benefits of multiprocessing, the iSBC 80/24A board provides full MULTIBUS arbitration control logic. This control logic allows up to three iSBC 80/24A boards or other bus masters to share the system bus in serial (daisy chain) priority fashion, and up to 16 masters to share the MULTIBUS system bus with the addition of an external priority network. The MULTIBUS arbitration logic operates synchronously with a MULTIBUS clock (provided by the iSBC 80/24A board or optionally connected directly to the MULTIBUS clock) while data is transferred via a handshake between the master and slave modules. This allows different speed controllers to share resources on the same bus since transfers via the bus proceed asynchronously. Thus, transfer speed is dependent on transmitting and receiving devices only. This design provides slow master modules from being handicapped in their attempts to gain control of the bus, but does not restrict the speed at which faster modules can transfer data via the same bus. The most obvious applications for the master-slave capabilities of the bus are multiprocessor configurations, high speed direct memory access (DMA) operations, and high speed peripheral control, but are by no means limited to these three.

Programmable Timers

The iSBC 80/24A board provides three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8254 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate time intervals under software control. Routing for the outputs and gate/trigger inputs of two of these counters is jumper selectable. The outputs may be independently routed to the 8259A Programmable Interrupt Controller, to the I/O line drivers associated with the 8255A Programmable Peripheral Interface, or may be routed as inputs to the 8255A chip. The gate/trigger inputs may be routed to I/O terminators associated with the 8255A or as output connections from the 8255A. The third interval timer in the 8254 provides the programmable baud rate generator for the RS232C USART serial port. In utilizing the iSBC 80/24A board, the systems designer simply configures, via software, each timer independently to meet system requirements. Whenever a given time delay or count is needed, software commands to the programmable timers/event counters select the desired function. The contents of each counter may be read at any time during system operation with simple read operations for event counting applications, and special commands are included so that the contents of each counter can be read "on the fly".

Interrupt Capability

The iSBC 80/24A board provides vectoring for 12 interrupt levels. Four of these levels are handled directly by the interrupt processing capability of the 8085A-2 CPU and represent the four highest priority interrupts of the iSBC 80/24A board. Requests are routed to the 8085A-2 interrupt inputs—TRAP, RST 7.5, RST 6.5, and RST 5.5 (in decreasing order of priority), each of which generates a call instruction to a unique address (TRAP: 24H; RST 7.5: 3CH; RST 6.5: 34H; and RST 5.5: 2CH). An 8085A-2 JMP instruction at each of these addresses then provides linkage to interrupt service routines located independently anywhere in memory. All interrupt inputs with the exception of the trap interrupt may be masked via software. The trap interrupt should be used for conditions such as power-down sequences which require immediate attention by the 8085A-2 CPU. The Intel 8259A Programmable Interrupt Controller (PIC) provides vectoring for the next eight interrupt levels. As shown in Table 3, a selection of four priority processing modes is available to the systems designer for use in designing request processing configurations to match system requirements. Operating mode and priority assignments may be reconfigured dynamically via software at any time during system operation. The PIC accepts interrupt requests from the programmable parallel and serial I/O interfaces, the programmable timers, the system bus, iSBX bus, or directly from peripheral equipment. The PIC then determines which of the incoming requests is of the highest priority, determines whether this request is of higher priority than the level currently being serviced, and, if appropriate, issues an interrupt to the CPU. Any combination of interrupt levels may be masked, via software, by storing a single byte in the interrupt mask register of the PIC. The PIC generates a unique memory address for each interrupt level. These addresses are equally spaced at intervals of 4 or 8 (software selectable) bytes. This 32 or 64-byte block may be located to begin at any 32 or 64-byte boundary in the 65,536-byte memory space. A single 8085A-2 JMP instruction at each of these addresses then provides linkage to locate each interrupt service routine independently anywhere in memory.

Programmable Interrupt Modes

Mode	Operation
Fully nested	Interrupt request line priorities fixed at 0 as highest, 7 as lowest.
Autorotating	Equal priority. Each level, after receiving service, becomes the lowest priority level until next interrupt occurs.
Specific priority	System software assigns lowest priority level. Priority of all other levels based in sequence numerically on this assignment.
Polled	System software examines priority-encoded system interrupt status via interrupt status register.

Interrupt Request Generation

Interrupt requests may originiate from 23 sources. Two jumper selectable interrupt requests can be generated by each iSBX MULTIMODULE board. Two jumper selectable interrupt requests can be automatically generated by each programmable peripheral interface when a byte of information is ready to be transferred to the CPU (i.e., input buffer is full) or a byte of information has been transferred to a peripheral device (i.e., output buffer is empty). Three jumper selectable interrupt requests can be automatically generated by the USART when a character is ready to be transferred to the CPU (i.e., receiver channel buffer is full), a character is ready to be transmitted (i.e., the USART is ready to accept a character from the CPU), or when the transmitter is empty (i.e., the USART has no character to transmit). A jumper selectable request can be generated by each of the programmable timers. Nine interrupt request lines are available to the user for direct interface to user designated peripheral devices via the MULTIBUS system bus. A power-fail signal can also be selected as an interrupt source.

Power-Fail Control

A power-fail interrupt may be detected through the AC-low signal generated by the power supply. This signal may be configured to interrupt the 8085A-2 CPU to initiate an orderly power down instruction sequence.

SPECIFICATIONS

Word Size

Instruction— 8, 16 or 24 bits
Data — 8 bits

Cycle Time

BASIC INSTRUCTION CYCLE

826 ns (4.84 MHz operating frequency)
1.65 μs (2.42 MHz operating frequency)

NOTE:
Basic instruction cycle is defined as the fastest instruction (i.e., four clock cycles).

Memory Addressing

ON-BOARD EPROM

0-0FFF using 2708, 2758 (1 wait state)
0-1FFF using 2716 (1 wait state)
0-3FFF using 2732 (1 wait state)
using 2732A (no wait states)
0-7FFF using 2764A (no wait states)

ON-BOARD RAM

E000-FFFF

NOTE:
Default configuration—may be reconfigured to top end of any 16K boundary.

Memory Capacity

ON-BOARD EPROM

32K bytes (sockets only)

ON-BOARD RAM

8K bytes

Serial Communications Characteristics

Synchronous — 5–8 bit characters; internal or external character synchronization; automatic sync insertion
Asynchronous— 5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detectors

Interfaces

MULTIBUS — All signals TTL compatible
iSBX Bus — All signals TTL compatible
Parallel I/O — All signals TTL compatible
Serial I/O — RS232C compatible, configurable as a data set or data terminal
Timer — All signals TTL compatible
Interrupt Requests— All TTL compatible

System Clock (8085A-2 CPU)

4.84 or 2.42 MHz ±0.1% (jumper selectable)

Auxiliary Power

An auxiliary power bus is provided to allow separate power to RAM for systems requiring battery backup of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

Memory Protect

An active-low TTL compatible memory protect signal is brought out on the auxiliary connector which, when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during system power-down sequences.

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.50 in. (1.27 cm)
Weight: 12.64 oz. (354 gm)

Electrical Characteristics

DC POWER REQUIREMENTS

Configuration	Current Requirements			
	V_{CC} = +5V ±5% (max)	V_{DD} = +12V ±5% (max)	V_{BB} = −5V ±5% (max)	V_{AA} = −12V ±5% (max)
Without EPROM(1)	2.66A	40 mA	—	20 mA
RAM Only(2)	0.01A	—	—	—
With iSBC 530(3)	2.66A	140 mA	—	120 mA
With 4K EPROM(4) (using 2708)	3.28A	300 mA	180 mA	20 mA
With 4K EPROM(4) (using 2758)	3.44A	40 mA	—	20 mA
With 8K EPROM(4) (using 2716)	3.44A	40 mA	—	20 mA
With 16K EPROM(4) (using 2732A)	3.46A	40 mA	—	20 mA
With 32K EPROM(4) (using 2764A)	3.42A	40 mA	—	20 mA

NOTES:
1. Does not include power for optional EPROM, I/O drivers, and I/O terminators.
2. RAM chips powered via auxiliary power bus.
3. Does not include power for optional EPROM, I/O drivers, I/O terminators. Power for iSBC 530 Adapter is supplied via serial port connector.
4. Includes power required for four EPROM chips, and I/O terminators installed for 16 I/O lines; all terminators inputs low.

Environmental Characteristics

Operating Temperature: 0°C to 55°C

ORDERING INFORMATION

Part Number	Description
SBC 80/24A	Single Board Computer

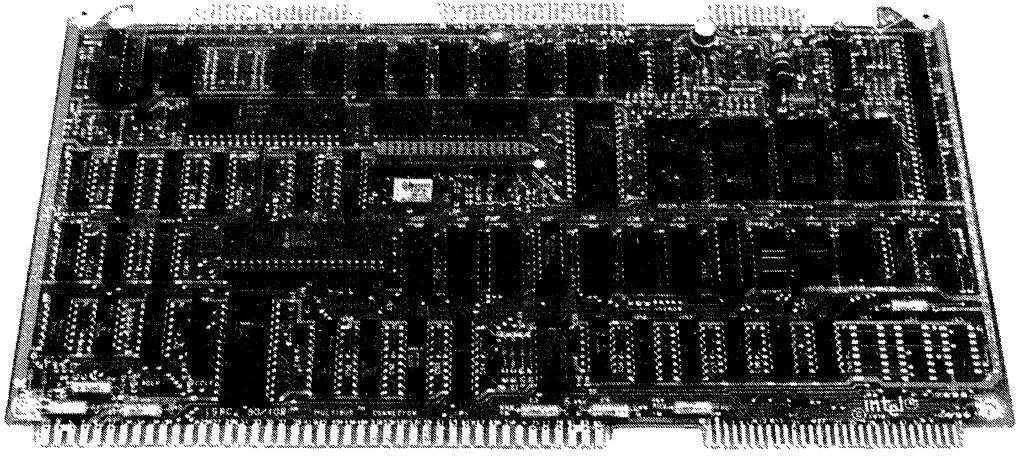
Reference Manual

148437-001— iSBC 80/24A Single Board Computer Hardware Reference Manual (NOT SUPPLIED)

iSBC® 80/10B*
SINGLE BOARD COMPUTER

- 8080A Central Processing Unit
- One iSBX™ Bus Connector for iSBX™ MULTIMODULE™ Board Expansion
- 1K Byte of Read/Write Memory with Sockets for Expansion up to 4K Bytes
- Sockets for up to 16K Bytes of Read Only Memory
- 48 Programmable Parallel I/O Lines with Sockets for Interchangeable Line Drivers and Terminators
- Programmable Synchronous/ Asynchronous Communications Interface with Selectable RS232C or Teletypewriter Compatiblity
- Single Level Interrupt with 11 Interrupt Sources
- Auxiliary Power Bus and Power-Fail Interrupt Control Logic for RAM Battery Backup
- 1.04 Millisecond Interval Timer
- Limited Master MULTIBUS® Interface

The Intel iSBC 80/10B board is a member of Intel's complete line of OEM microcomputer systems which take full advantage of Intel's LSI technology to provide economical, self-contained computer-based solutions for OEM applications. The CPU, system clock, iSBX bus interface, read/write memory, read only memory sockets, I/O ports and drivers, serial communications interface, bus control logic, and drivers all reside on the board.



280217-1

*The iSBC® 80/10B is also manufactured under product code piSBC® 80/10B by Intel Puerto Rico, Inc.

September 1989
Order Number: 280217-003

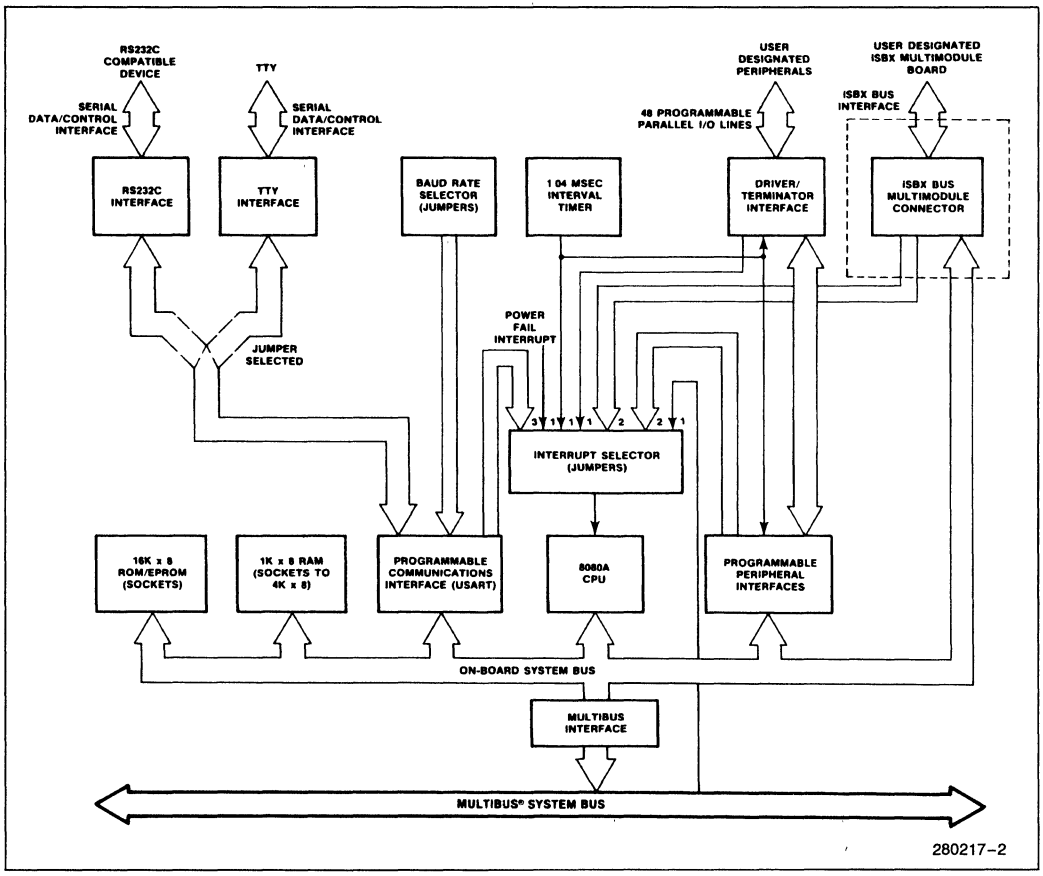


Figure 1. iSBC® 80/10B Single Board Computer Block Diagram

FUNCTIONAL DESCRIPTION

Intel's 8-bit n-channel MOS 8080A CPU, fabricated on a single LSI chip, is the central processor for the iSBC 80/10B board. The 8080A contains six 8-bit general purpose registers and an accumulator. The six general purpose registers may be addressed individually or in pairs, providing both single and double precision operators. A block diagram of iSBC 80/10B board functional components is shown in Figure 1.

iSBX™ Bus MULTIMODULE™ Board Expansion

One iSBX bus connector interface is provided to accomplish plug-in expansion with any iSBX MULTIMODULE board.

Memory Addressing

The 8080A has a 16-bit program counter which allows direct addressing of up to 64K bytes of memory. An external stack, located within any portion of read/write memory, may be used as a last-in/first-out storage area for the contents of the program counter, flags, accumulator, and all of the six general purpose registers. A 16-bit stack pointer controls the addressing of this external stack. This stack provides subroutine nesting bounded only by memory size.

Memory Capacity

The iSBC 80/10B board contains 1K bytes of read/write static memory. In addition, sockets for up to 4K bytes of RAM memory are provided on board. Read/write memory may be added in 1K byte increments. Sockets for up to 16K bytes of nonvolatile read-only-memory are provided on the board. All on-board RAM, ROM or EPROM read operations are performed at maximum processor speed.

Parallel I/O Interface

The iSBC 80/10B board contains 48 programmable parallel I/O lines implemented using two Intel 8255A programmable peripheral interfaces. The system software is used to configure the I/O lines in any combination of unidirectional input/output, and bidirectional ports. In order to take full advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators. The flexibility of the I/O interface is further enhanced by the capability of selecting the appropriate combination of optional line drivers and terminators to provide the required sink current, polarity, and drive/termination characteristics for each application. The 48 programmable I/O lines and signal ground lines are brought out to two 50-pin edge connectors that mate with flat cable or round cable.

Serial I/O Interface

A programmable communications interface using the Intel 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the board. A jumper selectable baud rate generator provides the USART with all common communications frequencies. The USART can be programmed by the system software to select the desired synchronous or asynchronous serial data transmission technique (including IBM Bi-Sync). The mode of operation (i.e., synchronous or asynchronous), data format, control character format and parity are all under program control. The 8251A provides full duplex, double-buffered transmit and receive capability. Parity, overrun, and framing error detection are all incorporated in the USART. The inclusion of jumper selectable TTY or RS232C compatible interfaces on the board, in conjunction with the USART, provides a direct interface to teletypes, CRTs, RS232C compatible cassettes, and asynchronous and synchronous modems. The RS232C or TTY command lines, serial data lines, and signal ground lines are brought out to a 26-pin edge connector that mates with RS232C compatible flat or round cable.

Interrupt Capability

Interrupt requests may originate from 11 sources. Two jumper selectable interrupt requests can be automatically generated by the programmable peripheral interface when a byte of information is ready to be transferred to the CPU (i.e., input buffer is full) or a byte of information has been transferred to a peripheral device (i.e., output buffer is empty). Three jumper selectable interrupt requests can be automatically generated by the USART when a character is ready to be transferred to the CPU (i.e., receive channel buffer is full), a character is ready to be transmitted (i.e., the USART is ready to accept a character from the CPU), or when the transmitter is empty (i.e., the USART has no character to transmit). These five interrupt request lines are all maskable under program control. Two interrupt request lines may be interfaced directly to user designated peripheral devices; one via the MULTIBUS system bus and the other via the I/O edge connector. One jumper selectable interrupt request may be interfaced to the power-fail interrupt control logic. One jumper selectable interrupt request may originate from the interval timer. Two general purpose interrupt requests are jumper selectable from the iSBX interface. These two signals permit a user installed MULTIMODULE board to interrupt to 8080A CPU. The eleven interrupt request lines share a single CPU interrupt level. When an interrupt request is recognized, a restart instruction (RESTART 7) is generated. The processor responds by suspending program execution and executing a user defined interrupt service routine.

Power-Fail Control

A power-fail interrupt may be detected through the AC-low signal generated by the power supply. This signal may be configured to interrupt the 8080A CPU to initiate an orderly power down instruction sequence.

Interval Timer

A 1.04 millisecond timer is available for interval interrupts or as a clock output to the parallel I/O connector. The timer output is jumper selectable to the programmable parallel interface, the parallel I/O connector (J1), or directly to the 8080A CPU.

SPECIFICATIONS

Word Size

Instruction: 8, 16, or 24 bits
Data: 8 bits

Cycle Time

Basic Instruction Cycle: 1.95 μs

NOTE:
Basic instruction cycle is defined as the fastest instruction (i.e., four clock cycles).

Memory Addressing

On-Board ROM/EPROM
0–0FFF using 2708, 2758
0–1FFF using 2716
0–3FFF using 2732

On-Board RAM
3C00–3FFF with no RAM expansion
3000–3FFF with 2114A-5 expansion

Memory Capacity

On-Board ROM/EPROM
16K bytes (sockets only)

On-Board RAM
1K byte with user expansion in 1K increments to 4K byte using Intel 2114A-5 RAMs.

Off-Board Expansion
Up to 64K bytes using user specified combinations of RAM, ROM, and EPROM.

I/O Capacity

Parallel:	48 programmable lines
Serial:	1 transmit, 1 receive
MULTIMODULE:	1 iSBX Bus MULTIMODULE Board

Serial Communications Characteristics

Synchronous:	5–8 bit characters; internal or external character synchronization; automatic sync insertion
Asynchronous:	5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detectors

Interrupts

Single-level with on-board logic that automatically vectors the processor to location 38H using a restart instruction (RESTART 7). Interrupt requests may originate from user specified I/O (2); the programmable peripheral interface (2); the iSBX MULTIMODULE board (2); the programmable communications interface (3); the power fail interrupt (1); or the interval timer (1).

Interfaces

MULTIBUS:	All signals TTL compatible
iSBX Bus:	All signals TTL compatible
Parallel I/O:	All signals TTL compatible
Serial I/O:	RS232C or a 20 mil current loop TTY interface (jumper selectable)
Interrupt Requests:	All TTL compatible (active-low)

Clocks

System Clock: 2.048 MHz ±0.1%
Interval Timer: 1.042 ms ±0.1% (959.5 Hz)

Physical Characteristics

Width: 12.00 in (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.05 in. (1.27 cm)
Weight: 14 oz. (397.3 gm)

Electrical Characteristics

DC Power Requirements

Voltage	Without EPROM(1)	With 2708 EPROM(2)	With 2758, 2716, or 2732 EPROM(3)	Power Down Requirements (RAM and Support Circuit)
V_{CC} = +5V ±5%	I_{CC} = 2.0A(4)	3.1A	3.46A	84 mA + 140 mA/K (2114A-5)
V_{DD} = +12V ±5%	I_{DD} = 150 mA	400 mA	150 mA	Not Required
V_{BB} = −5V ±5%	I_{BB} = 2 mA	200 mA	2 mA	Not Required
V_{AA} = −12V ±5%	I_{AA} = 175 mA	175 mA	175 mA	Not Required

NOTES:
1. Does not include power required for optional ROM/EPROM, I/O drivers, or I/O terminators.
2. With four Intel 2708 EPROMS and 220Ω/330Ω for terminators, installed for 48 input lines. All terminator inputs low.
3. Same as #2 except with four 2758s, 2716s, or 2732s installed.
4. I_{CC} shown without RAM supply current. For 2114-5 add 140 mA per K byte to a maximum of 560 mA.

Environmental Characteristics

Operating Temperature: 0°C to 55°C

Reference Manual

9803119-01— iSBC 80/10B Single Board Computer Hardware Reference Manual (NOT SUPPLIED).

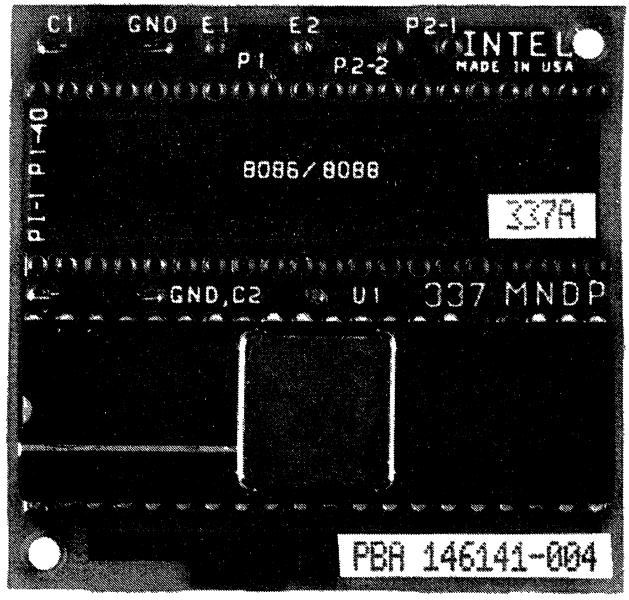
ORDERING INFORMATION

Part Number	Description
iSBC80/10B	Single Board Computer

iSBC® 337A MULTIMODULE™ NUMERIC DATA PROCESSOR

- **High Speed Fixed and Floating Point Functions for iSBC® Boards**
- **Extends Host CPU Instruction Set with Arithmetic, Logarithmic, Transcendental and Trigonometric Instructions**
- **MULTIMODULE™ Option Containing 8087 Numeric Data Processor**
- **Supports Seven Data Types Including Single and Double Precision Integer and Floating Point**
- **Fully Supported in the Multi-Tasking Environment of the iRMX™ I Operating System**

The Intel iSBC® 337A MULTIMODULE™ Numeric Data Processor offers high performance numerics support for iSBC 86 and iSBC 88 Single Board Computer users, for applications including simulation, instrument automation, graphics, signal processing and business systems. The coprocessor interface between the 8087 and the host CPU provides a simple means of extending the instruction set with over 60 additional numeric instructions supporting seven additional data types.



280077-1

September 1989
Order Number: 280077-002

OVERVIEW

The iSBC 337A MULTIMODULE Numeric Data Processor (NDP) provides arithmetic and logical instruction extensions to the 86/88 families. The instruction set consists of arithmetic, transcendental, logical, trigonometric and exponential instructions which can all operate on seven different data types. The data types are 16-, 32-, and 64-bit integer, 32- and 64-bit floating point, 18 digit packed BCD and 80-bit temporary.

Coprocessor Interface

The coprocessor interface between the host CPU and the iSBC 337A MULTIMODULE provides easy to use and high performance math processing. Installation of the iSBC 337A is simply a matter of removing the host CPU from its socket, installing the iSBC 337A MULTIMODULE into the host's CPU socket, and reinstalling the host CPU chip into the socket provided for it on the iSBC 337A MULTIMODULE (see Figure 1).

All synchronization and timing signals are provided via the coprocessor interface with the host CPU. The two processors also share a common address/data bus. The NDP component is capable of recognizing and executing NDP numeric instructions as they are fetched by the host CPU. This interface allows concurrent processing by the host CPU and the NDP. It also allows NDP and host CPU instructions to be intermixed in any fashion to provide the maximum overlapped operation and the highest aggregate performance.

High Performance and Accuracy

The 80-bit wide internal registers and data paths contribute significantly to high performance and minimize the execution time difference between single and double precision floating point formats. This 80-bit architecture provides very high resolution and accuracy.

This precision is complemented by extensive exception detection and handling. Six different types of exceptions can be reported and handled by the NDP. The user also has control over internal precision, infinity control and rounding control.

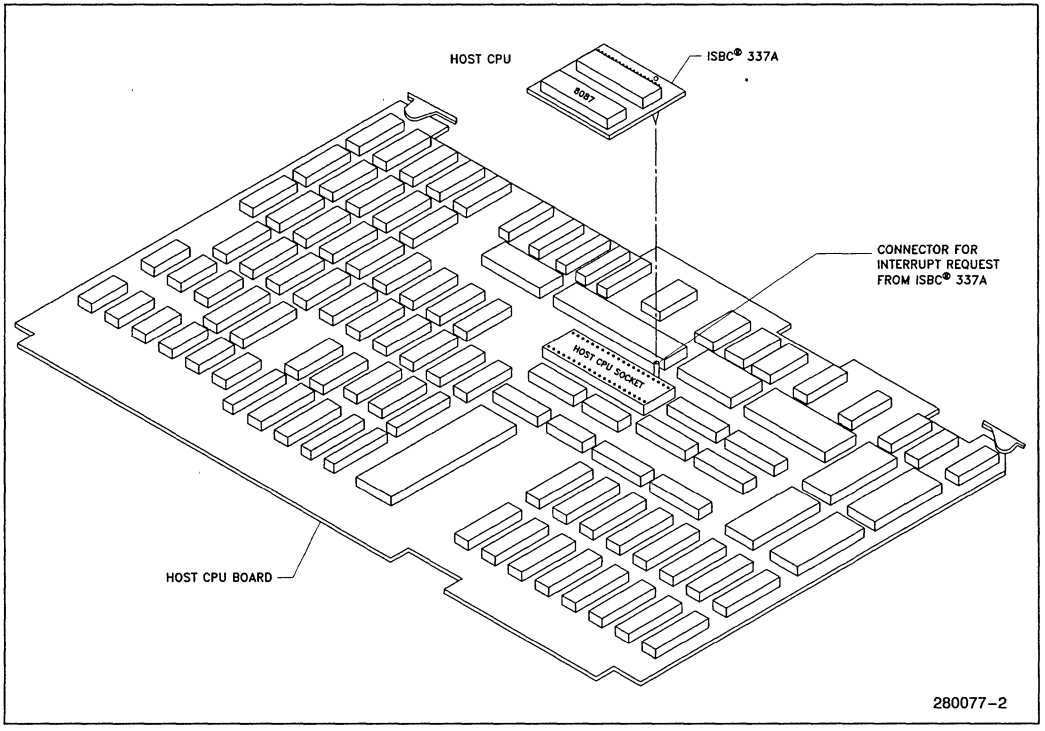


Figure 1. iSBC® 337A MULTIMODULE Installation

SYSTEM CONFIGURATION

As a coprocessor to the Host CPU, the NDP is wired in parallel with the CPU. The CPU's status and queue status lines enable the NDP to monitor and decode instructions in synchronization with the CPU and without any CPU overhead. Once started, the NDP can process in parallel with and independent of the host CPU. For resynchronization, the NDP's BUSY signal informs the CPU that the NDP is executing an instruction and the CPU WAIT instruction tests this signal to insure that the NDP is ready to execute subsequent instructions.

The NDP can interrupt the CPU when it detects an error or exception. The interrupt request line is routed to the CPU through an 8259A Programmable Interrupt Controller. This interrupt request signal is brought down from the iSBC 337A MULTIMODULE to the single board computer through a single pin connector (see Figure 1). The signal is then routed to the interrupt matrix for jumper connection to the 8259A Interrupt Controller. Other iAPX designs may use a similar arrangement, or by masking off the CPU "READ" pin from the iSBC 337A socket, provisions are made to allow the now vacated pin of the host's CPU socket to be used to bring down the interrupt request signal for connection to the base board and then to the 8259A.

FUNCTIONAL DESCRIPTION

The NDP is internally divided into two processing elements, the control unit (CU) and the numeric execution unit (NEU), providing concurrent operation of the two units. The NEU executes all numeric instructions, while the CU receives and decodes instructions, reads and writes memory operands and executes processor control instructions.

Control Unit

The CU keeps the NDP operating in synchronization with its host CPU. NDP instructions are intermixed with CPU instructions in a single instruction stream. The CPU fetches all instructions from memory; by monitoring the status signals emitted by the CPU, the NDP control unit determines when a 8086-2 instruction is being fetched. The CU taps the bus in parallel with the CPU and obtains that portion of the data stream.

After decoding the instruction, the host executes all opcodes but ESCAPE (ESC), while the NDP executes only the ESCAPE class instructions. (The first five bits of all ESCAPE instructions are identical). The CPU does provide addressing for ESC instructions however.

An NDP instruction either will not reference memory, will require loading one or more operands from memory into the NDP, or will require storing one or more operands from the NDP into memory. In the first case, a non-memory reference escape is used to start NDP operation. In the last two cases, the CU makes use of a "dummy read" cycle initiated by the CPU, in which the CPU calculates the operand address and initiates a bus cycle, but does not capture the data. Instead, the CPU captures and saves the address which the CPU places on the bus. If the instruction is a load, the CU additionally captures the data word when it becomes available on the local data bus. If data required is longer than one word, the CU immediately obtains the bus from the CPU using the request/grant protocol and reads the rest of the information in consecutive bus cycles. In a store operation, the CU captures and saves the store address as in a load, and ignores the data word that follows in the "dummy read" cycle. When the NDP is ready to perform the store, the CU obtains the bus from the CPU and writes the operand starting at the specified address.

Numeric Execution Unit

The NEU executes all instructions that involve the register stack. These include arithmetic, logical, transcendental, constant and data transfer instructions. The data path in the NEU is 80 bits wide (64 fraction bits, 15 exponent bits and a sign bit) which allows internal operand transfers to be performed at very high speeds.

When the NEU begins executing an instruction, it activates the NDP BUSY signal. This signal is used in conjunction with the CPU WAIT instruction to resynchronize both processors when the NEU has completed its current instruction.

SPECIFICATIONS

Physical Characteristics

Width— 5.33 cm (2.100″)
Length— 5.08 cm (2.000″)
Height— 1.82 cm (0.718″)
iSBC 337A board + host board
Weight— 17.33 grams (0.576 oz.)

Electrical Characteristics

DC Power Requirements

V_{CC} = 5V ±5%
I_{CC} = 475 mA max.
I_{CC} = 350 mA typ.

Environmental Characteristics

Operating Temperature—0°C to 55°C with 200 linear feet/minute airflow

Relative Humidity—Up to 90% R.H. without condensation.

Reference Manual

147163-001—iSBC 337A MULTIMODULE Numeric Data Processor Hardware Reference Manual (NOT SUPPLIED WITH MULTIMODULE BOARD).

ORDERING INFORMATION

Part Number	Description
SBC 337A	MULTIMODLE Numeric Data Processor

MULTIBUS® I
Memory Expansion Boards

9

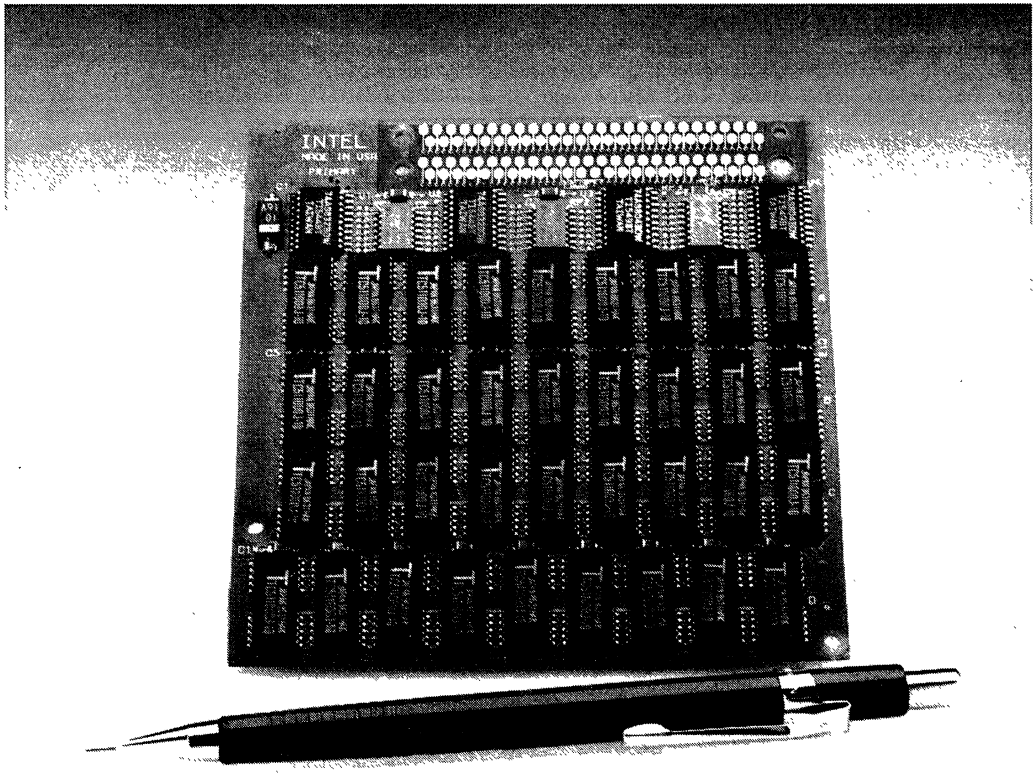
iSBC® MM01, MM02, MM04, MM08* HIGH PERFORMANCE MEMORY MODULES

- **Provides High Speed Parity Memory Expansion for Intel's iSBC® 386/2X, iSBC 386/3X and iSBC 386/1XX CPU Boards**
- **Available in 1M, 2M, 4M, and 8M Byte Sizes**
- **32 Bits Wide with Byte Parity**
- **Stackable to Provide up to 16M Bytes of High Speed Memory for MULTIBUS I and MULTIBUS II CPU Boards**
- **Supports 32-Bit, 16-Bit and 8-Bit Data Paths**
- **Supports Independent Read/Writes**
- **Easily Installed**

The iSBC MM01, iSBC MM02, iSBC MM04, and iSBC MM08 DRAM memory modules are members of Intel's complete line of iSBC memory and I/O expansion boards. The MM-Series of memory modules use a dedicated interface to maximize CPU/memory performance. The iSBC MM series of memory modules have been designed to provide both the on-board and expansion memory for the iSBC 386/2X, the iSBC 386/3X and the iSBC 386/1XX CPU Boards.

The modules contain (respectively) 1M byte, 2M, 4M, and 8M bytes of read/write memory using surface mounted DRAM components (see Figure 1).

Due to the high speed interface of the memory modules, they are ideally suited in applications where memory performance is critical.



280346-1

Figure 1. iSBC® MM08 Memory Module

*The iSBC® MM01, MM02, MM04, MM08 Memory Modules are also manufactured under product code piSBC® MM01, MM02, MM04, MM08 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280346-002

FUNCTIONAL DESCRIPTION

The iSBC MMxx memory modules provide high performance, 32-bit parity DRAM memory for the MULTIBUS I and MULTIBUS II CPU boards. These CPU boards come standard with one MMxx module installed, with memory expansion available through the addition of a second stackable iSBC MMxx module.

Memory Access Capabilities

The dynamic RAM memory of the memory modules is accessed through the dedicated memory module interface.

The MM memory module is designed for direct transfer of data between the CPU and the memory module without accessing the MULTIBUS interface.

MM01/MM02/MM04/MM08 Memory Size

The iSBC MM01, iSBC MM02, iSBC MM04, and iSBC MM08 modules can be stacked on the CPU baseboard in any combination.

Data Bus Structure

The MMxx-series memory modules use a 32-bit wide data path with storage for byte parity that can accommodate 8-bit byte, 16-bit or 32-bit word data transfers. In addition, the data path is capable of independent byte operations. This means that one byte can be written while the other three bytes (or any other combination) can be read.

Parity

One parity bit is provided for each of the four, 8-bit bytes in the 32-bit wide data path. For special applications, the parity bits can serve as data bits making possible 9-, 18-, or 36-bit data transfers.

Memory Function

The module protocol supports standard dynamic RAM READ, WRITE, RAS* only REFRESH cycles, and CAS* before RAS* REFRESH.

Installation

The iSBC MMxx memory modules are easily installed by the user. Each module includes all necessary connectors, screws, and other hardware for installation, either as a second stacked module or as a replacement for a module with less memory.

SPECIFICATIONS

Word Size Supported

8-, 16-, or 32-bits

Memory Size

iSBC MM01	1,048,576 bytes
iSBC MM02	2,097,152 bytes
iSBC MM04	4,194,304 bytes
iSBC MM08	8,388,608 bytes

Access Time (All Densities)

Read/Write — 107 ns (max)

The MMxx-series memory modules run with the iSBC 386/2X and iSBC 386/116 Boards at 16 MHz, and with the iSBC 386/3X and iSBC 386/120 Boards at 20 MHz. Wait state performance information with each of these CPU baseboards is contained in the Hardware Reference Manual for the specific CPU baseboard.

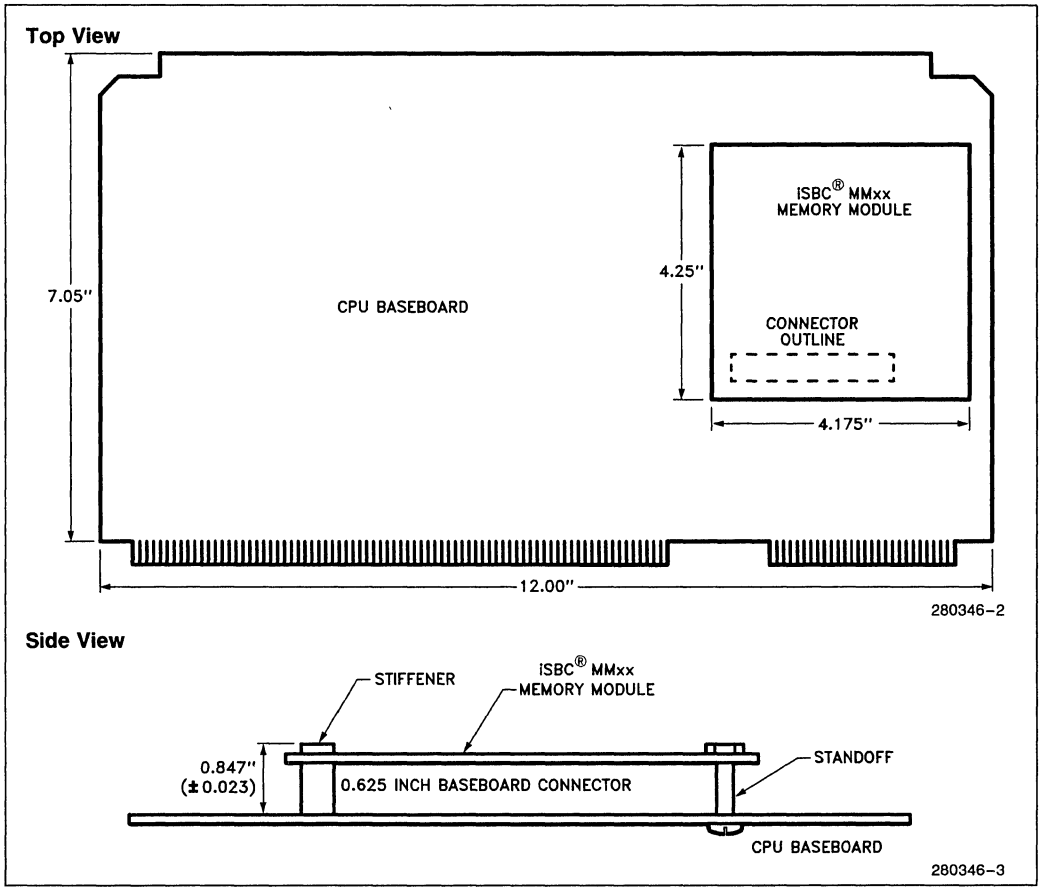
Cycle Time (All Densities)

Read/Write — 200 ns (min)

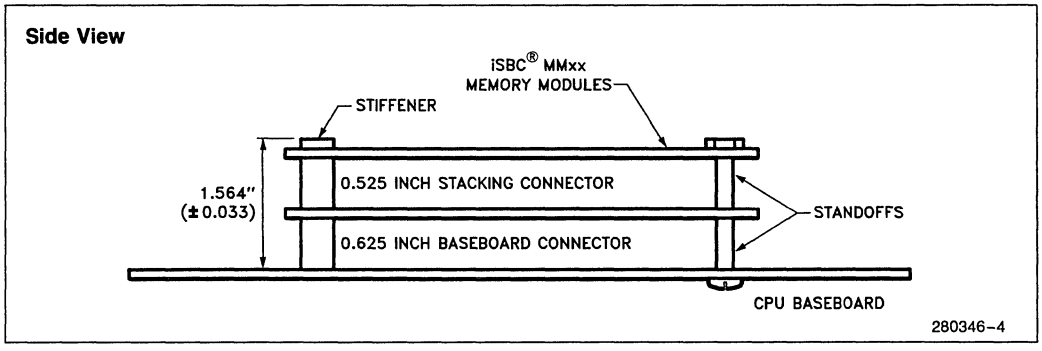
Power Requirements

Voltage −5 VDC ±5%

Memory addressing for the iSBC MMxx memory modules is controlled by the host CPU board over the memory module interface. The maximum system RAM size is 16M Bytes.



Single iSBC® MMxx Memory Module



Stacked iSBC® MMxx Memory Modules

Environmental Requirements

Operating Temperature — 0°C to 60°C

Storage Temperature — 40°C to +75°C

Cooling Requirement — 3 cubic feet per minute of airflow at an ambient temperature of 0°C to 60°C

Operating Humidity — To 95% relative humidity without condensation

Physical Dimensions

Module Alone:

Width — 4.250 inches (10,795 cm)

Length — 4.175 inches (10,604 cm)

Height — 0.500 inches (1,270 cm)

Weight — iSBC MM01/MM04: 2.5 ounces (70.0 gm)
iSBC MM02/MM08: 3.5 ounces (110.0 gm)

ORDERING INFORMATION

Part Number	Description
iSBC MM01	1M Byte RAM Memory Module
iSBC MM02	2M Byte RAM Memory Module
iSBC MM04	4M Byte RAM Memory Module
iSBC MM08	8M Byte RAM Memory Module

The Memory Modules ship with the required hardware (connectors, mounting screws, stand-offs, etc.) to stack a second module on the module already mounted on the base CPU board.

iSBC® MM01FP, MM02FP, MM04FP, MM08FP* HIGH PERFORMANCE MEMORY MODULES

- **Provides High Speed Parity Memory Expansion for Intel's iSBC® 386/2X, iSBC 386/3X and iSBC 386/1XX CPU Boards**
- **Available in 1M, 2M, 4M, and 8M Byte Sizes**
- **32 Bits Wide with Byte Parity**
- **Stackable to Provide up to 16M Bytes of High Speed Memory for MULTIBUS I and MULTIBUS II Boards**
- **Supports 32-Bit, 16-Bit and 8-Bit Data Paths**
- **Supports Independent Read/Writes**
- **Easily Installed**

The iSBC MM0X and iSBC MM0XFP DRAM memory modules are members of Intel's complete line of iSBC memory and I/O expansion boards. The MM-Series of memory modules use a dedicated memory interface to maximize CPU/memory performance.

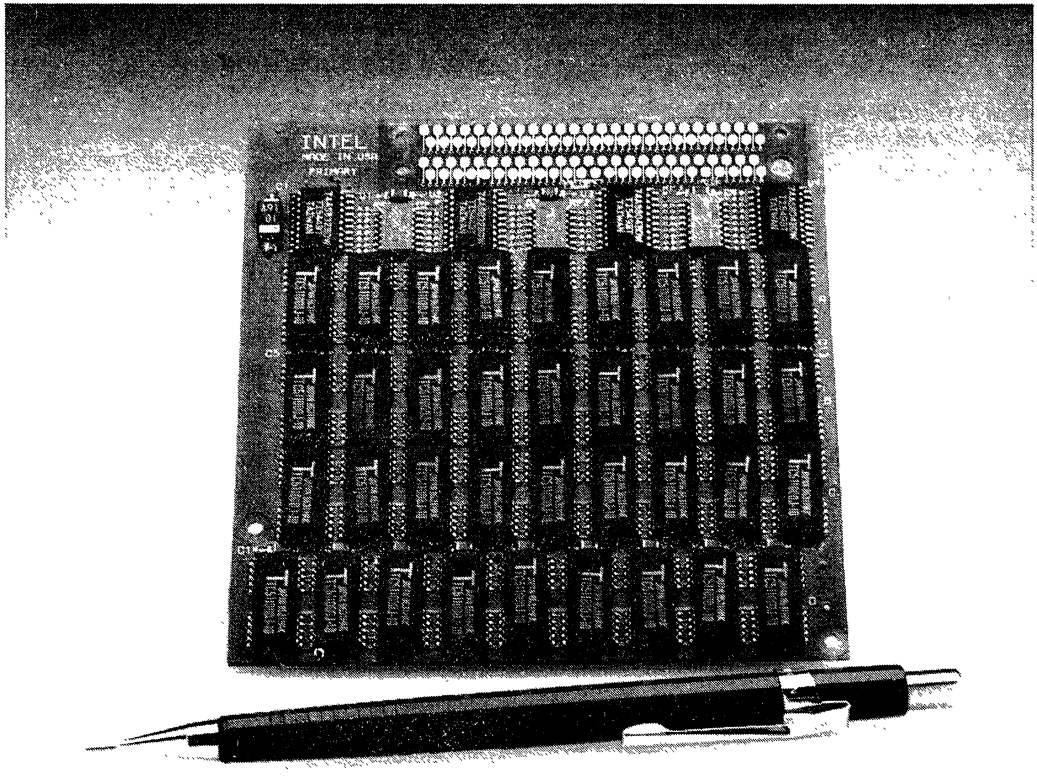


Figure 1. iSBC® MM08FP Memory Module 281010-1

The iSBC® MM01FP, MM02FP, MM04FP, MM08FP memory modules are also manufactured under product code piSBC® MM01FP, MM02FP, MM04FP, MM08FP by Intel Puerto Rico, Inc.

September 1989
Order Number: 281010-001

FUNCTIONAL DESCRIPTION

The iSBC MM-Series provide high performance, 32-bit parity DRAM memory for the MULTIBUS I and MULTIBUS II boards. These CPU boards come standard with one MM-Series module installed, with memory expansion available through the addition of a second stackable iSBC MM-Series module.

Memory Access Capabilities

The dynamic RAM memory of the memory modules is accessed through the dedicated memory module interface.

The MM memory module is designed for direct transfer of data between the CPU and the memory module without accessing the MULTIBUS interface.

MM01/MM02/MM04/MM08 Memory Size

The iSBC MM01, iSBC MM02, iSBC MM04, and iSBC MM08 modules can be stacked on the CPU baseboard in any combination.

Data Bus Structure

The MM-Series memory modules use a 32-bit wide data path with storage for byte parity that can accommodate 8-bit byte, 16-bit or 32-bit word data transfers. In addition, the data path is capable of independent byte operations. This means that one byte can be written while the other three bytes (or any other combination) can be read.

Parity

One parity bit is provided for each of the four, 8-bit bytes in the 32-bit wide data path. For special applications, the parity bits can serve as data bits making possible 9-, 18-, or 36-bit data transfers.

Memory Function

The module protocol supports standard dynamic RAM READ, WRITE, RAS* only REFRESH cycles, and CAS* before RAS* REFRESH.

Installation

The iSBC MM-Series memory modules are easily installed by the user. Each module includes all necessary connectors, screws, and other hardware for installation, either as a second stacked module or as a replacement for a module with less memory.

SPECIFICATIONS

Word Size Supported

8-, 16-, or 32-bits

Memory Size

iSBC MM01	1,048,576 bytes
iSBC MM02	2,097,152 bytes
iSBC MM04	4,194,304 bytes
iSBC MM08	8,388,608 bytes

Access Time (All Densities)

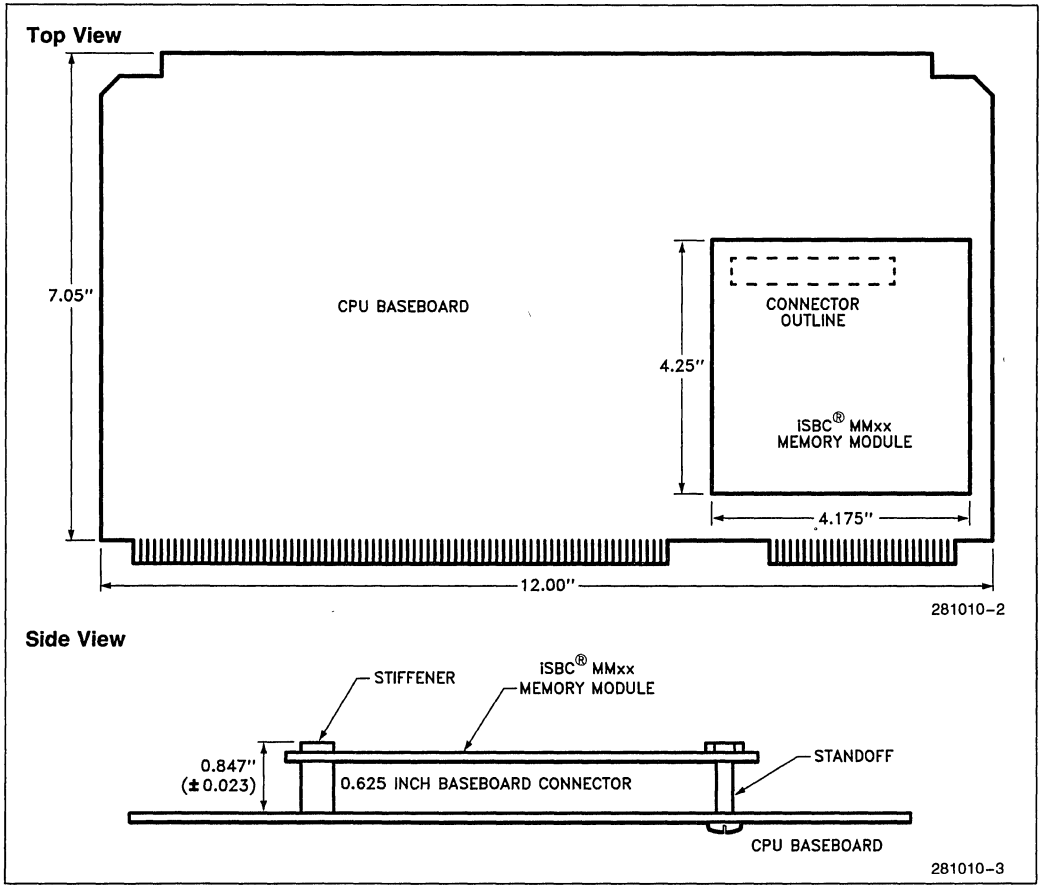
Read/Write — 107 ns (max)-MM0X

Read/Write — 88 ns (max)-MM0XFP

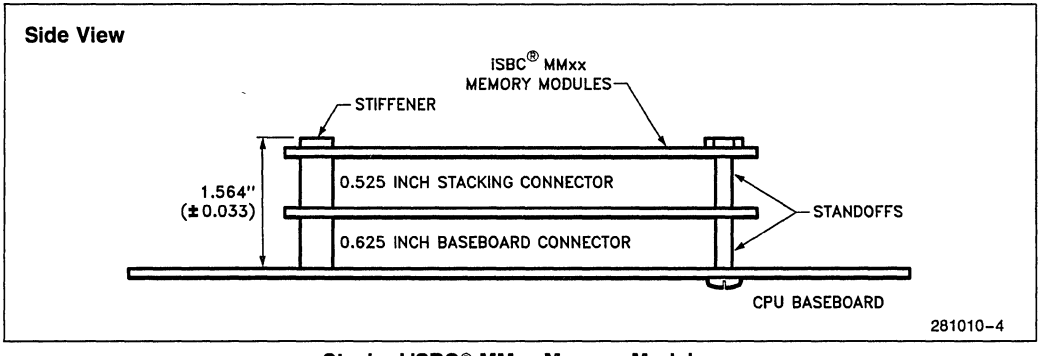
Power Requirements

Voltage −5 VDC ±5%

Memory addressing for the iSBC MM-Series memory modules is controlled by the host CPU board over the memory module interface. The maximum system RAM size is 16M Bytes.



Single iSBC® MMxx Memory Module



Stacked iSBC® MMxx Memory Modules

Environmental Requirements

Operating Temperature — 0°C to 60°C

Storage Temperature — 40°C to +75°C

Operating Humidity — To 95% relative humidity without condensation

Physical Dimensions

Module Alone:

Width — 4.250 inches (10,795 cm)

Length — 4.175 inches (10,604 cm)

Height — 0.500 inches (1,270 cm)

Weight — iSBC MM01/MM04: 2.5 ounces (70.0 gm)
iSBC MM02/MM08: 3.5 ounces (110.0 gm)

ORDERING INFORMATION

Part Number	Description
iSBC MM01FP	1M Byte Fast Page Memory Module
iSBC MM02FP	2M Byte Fast Page Memory Module
iSBC MM03FP	4M Byte Fast Page Memory Module
iSBC MM04FP	8M Byte Fast Page Memory Module

The Memory Modules ship with the required hardware (connectors, mounting screws, stand-offs, etc.) to stack a second module on the module already mounted on the base CPU board.

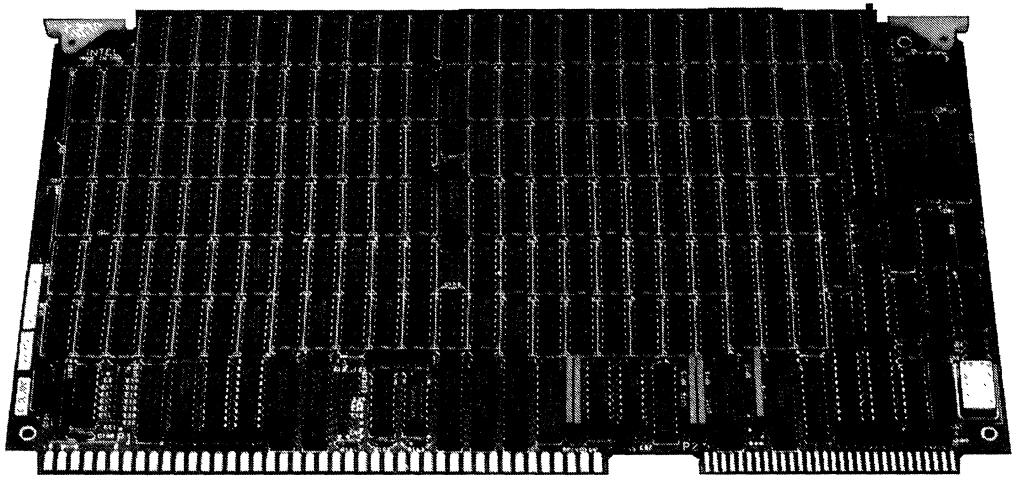
iSBC® 012EX, 010EX, 020EX, and 040EX* HIGH PERFORMANCE RAM BOARDS

- **0 Wait States at 8 MHz Performance with the iSBC® 286/10A, iSBC 286/12 Board**
- **Dual Port Capability Via MULTIBUS® and High Speed Synchronous Interface**
- **Configurable to Function Over iLBX™ Bus**
- **On-Board Parity Generator/Checker**
- **Independently Selectable Starting and Ending Addresses**
- **16 Megabyte Addressing Capability**
- **512K Byte, 1024K Byte, 2048K Byte, and 4096K Byte Densities Available**

The iSBC 012EX, iSBC 010EX, iSBC 020EX, and iSBC 040EX RAM memory boards are members of Intel's complete line of iSBC memory and I/O expansion boards. The EX boards are dual ported between the MULTIBUS interface and one of two types of dedicated memory buses. The dedicated buses are the iLBX bus and a high speed interface. The EX series of RAM-boards can be configured to be accessed over the iLBX bus, as well as MULTIBUS bus, to provide memory support for the iSBC 286/10 board, iSBC 186/03A, or iSBC 386/12. The EX boards are default configured to run over the MULTIBUS interface and the high speed interface. This provides 0 wait state 8 MHz memory support for the iSBC 286/10A and iSBC 286/12 boards.

The EX RAM-boards generate byte oriented parity during all write operations and perform parity checking during all read operations. An on-board LED provides a visual indication that a parity error has occurred.

The iSBC 012EX, iSBC 010EX, iSBC 020EX, and iSBC 040EX boards contain 512K bytes, 1M byte, 2M bytes, and 4M bytes of read/write memory using 256K dynamic RAM components.



280142-1

*The iSBC® 012EX, 010EX, 020EX and 040EX Boards are also manufactured under product code piSBC® or siSBC® 012EX, 010EX, 020EX and 040EX by Intel Puerto Rico, Inc. and Intel Singapore, Ltd.

September 1989
Order Number: 280142-002

FUNCTIONAL DESCRIPTION

General

The iSBC 012EX, 010EX, 020EX, and 040EX RAM boards are physically and electrically compatible with the MULTIBUS interface standard, IEEE-796, as outlined in the Intel MULTIBUS architecture specification.

Dual Port Capabilities

The "EX" series of RAM-Boards can be accessed by the MULTIBUS interface, and either the iLBX Bus, or the high speed synchronous interface (see Figures 1 and 2). The EX series require jumper and PAL configuration to be accessed over iLBX Bus.

Intel's iLBX interface is an unarbitrated bus architecture which allows direct transfer of data between the CPU and the memory boards without accessing the MULTIBUS bus. Due to the unarbitrated nature of the iLBX interface, significant improvements in memory access times compared to the MULTIBUS bus accesses result. The EX Boards provide 1 wait state performance at 6 MHz and 2 wait states at 8 MHz over the iLBX board. The EX Memory Board Hardware Reference Manual should be consulted for details.

The high speed synchronous interface, like the iLBX Bus, is a bus architecture which allows direct transfer of data between the CPU and the memory boards without accessing the MULTIBUS bus. This high speed interface runs synchronously with the iSBC 286/10A and iSBC 286/12 to provide 0 wait state performance at 8 MHz.

System Memory Size

Maximum system memory size with this series of boards is 16 megabytes. Memory partitioning is independent for the MULTIBUS interface and the iLBX interface.

Address Selection/Memory

SELECTABLE STARTING ADDRESS

A 256K boundary select is implemented on the iSBC 012EX board. A 512K boundary select is implemented on the iSBC 010EX board. A 1M boundary is implemented on the iSBC 020EX and iSBC 040EX boards.

SELECTABLE ENDING ADDRESS

The ending address is selectable as memory size minus select options of 0, 128K, 256K, or 512K on all of the EX boards.

PARITY INTERRUPT CLEAR

The I/O address of the Parity Interrupt Clear circuitry is jumperable to any one of 256 addresses.

SPECIFICATIONS

Word Size Supported

8- or 16-bits.

Memory Size

524,288 bytes (iSBC 012EX board)
1,048,576 bytes (iSBC 010EX board)
2,097,152 bytes (iSBC 020EX board)
4,194,304 bytes (iSBC 040EX board)

Access Times (All densities)

MULTIBUS® SYSTEM BUS

Read/Full Write— 375 ns (max)
Write Byte— 375 ns (max)

HIGH SPEED SYNCHRONOUS INTERFACE

Read/Full Write— 167 ns (max)
Write Byte— 132 ns (max)

iLBX™ BUS

Read/Full Write— 295 ns (max)
Write Byte— 116 ns (max)

Cycle Times (All densities)

MULTIBUS® SYSTEM BUS

Read/Full Write— 625 ns (max)
Write Byte— 625 ns (max)

HIGH SPEED SYNCHRONOUS INTERFACE

Read/Full Write— 250 ns (max)
Write Byte — 250 ns (max)

iLBX™ BUS

Read/Full Write— 437.5 ns (max)
Write Byte — 437.5 ns (max)

Memory Partitioning

Maximum System memory size is 16M Bytes for the MULTIBUS, iLBX bus and the high speed interface.

BASE ADDRESS

Board	Base Address
iSBC 012EX Board	any 256K boundary in first 4 megabytes
iSBC 010EX Board	any 512K boundary in first 8 megabytes
iSBC 020EX Board	any 1M boundary
iSBC 040EX Board	any 1M boundary

Power Requirements

Voltage—5 VDC ±5%

Product	Current
iSBC 012EX Board	3.2A (typ) 4.9A (max)
iSBC 010EX Board	3.4A (typ) 5.0A (max)
iSBC 020EX Board	3.7A (typ) 5.2A (max)
iSBC 040EX Board	3.9A (typ) 5.5A (max)

ENVIRONMENTAL REQUIREMENTS

Operating Temperature: 0°C to 60°C airflow of 5 cubic feet per minute

Storage Temperature: −40°C to +75°C

Operating Humidity: To 90% without condensation

PHYSICAL DIMENSIONS

Width: 12 inches (30.48 cm)

Height: 6.75 inches (17.15 cm)

Thickness: 0.50 inches (1.27 cm)

Weight: iSBC 012EX board: 6.8 ounces (1910 gm)
iSBC 010EX board: 9.0 ounces (2550 gm)
iSBC 020EX board: 13.5 ounces (3830 gm)
iSBC 040EX board: 18.0 ounces (5100 gm)

REFERENCE MANUALS

147783-001— iSBC 012EX/iSBC 010EX/iSBC 020EX/iSBC 040EX Hardware Reference Manual

144456-001— Intel iLBX Specification

ORDERING INFORMATION

Part Number	Description
iSBC 012EX	512K byte RAM board with parity
iSBC 010EX	1M byte RAM board with parity
iSBC 020EX	2M byte RAM board with parity
iSBC 040EX	4M byte RAM board with parity
EXASYNCX86	PALs and jumper configuration for iLBX mode

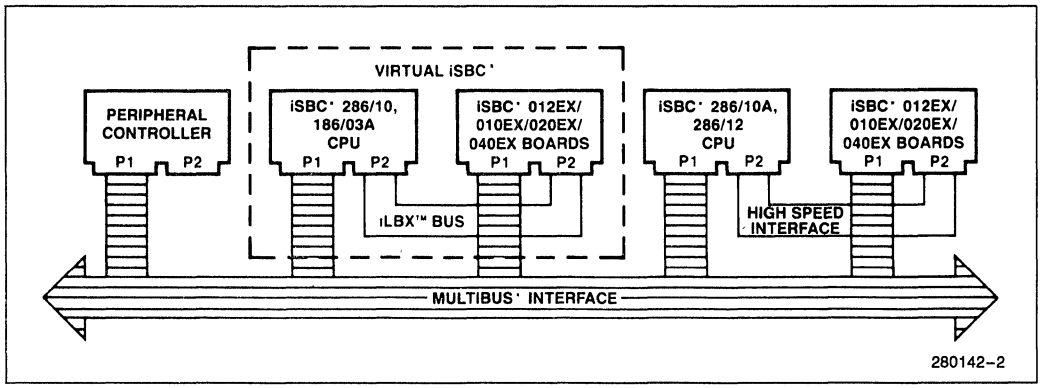


Figure 1. Typical iLBX™ System Configuration

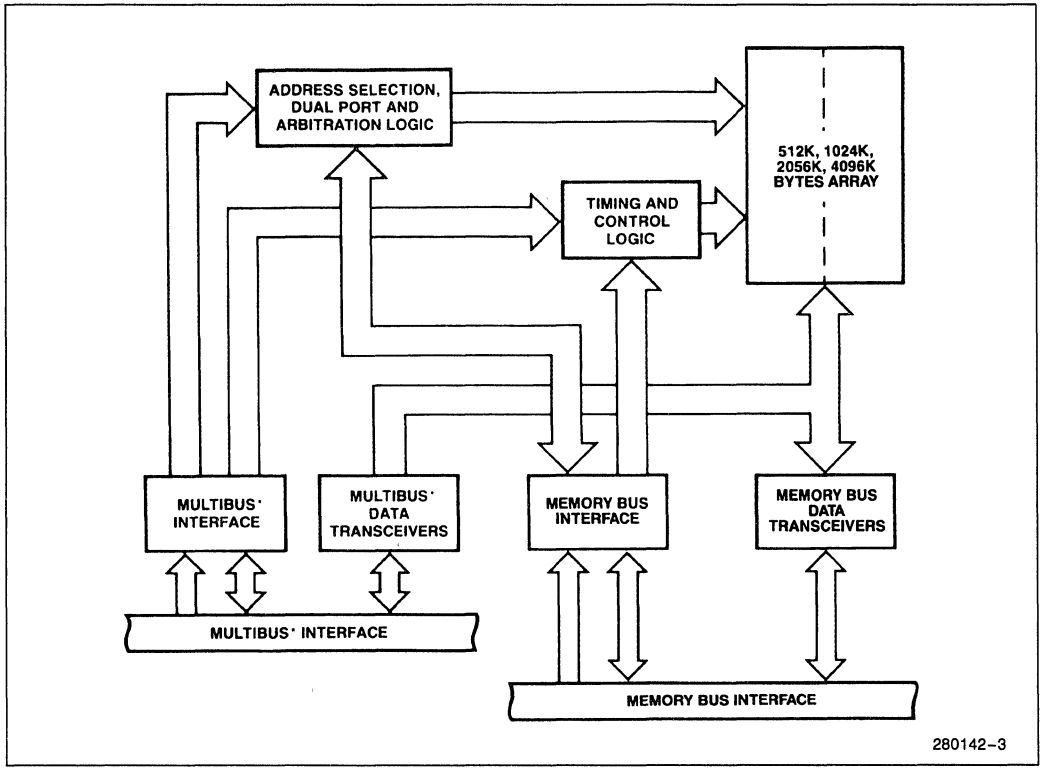


Figure 2. iSBC® EX Memory Board Block Diagram

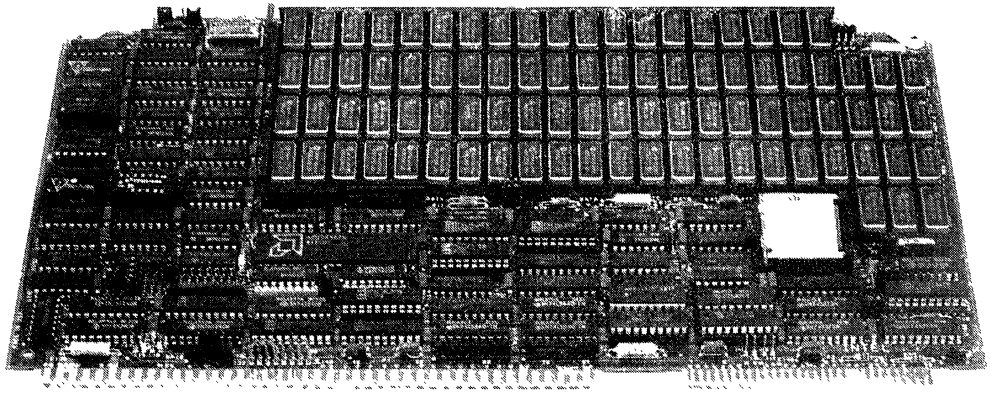
iSBC® 012CX, 010CX, AND 020CX*
iLBX™ RAM BOARDS

- **Dual Port Capability via MULTIBUS® and iLBX Interfaces**
- **Single Bit Error Correction and Double Bit Error Detection Utilizing Intel 8206 ECC Device**
- **512K Byte, 1024K Byte, and 2048K Byte Versions Available**
- **Control Status Register Supports Multiple ECC Operating Modes**
- **Error Status Register Provides Error Logging by Host CPU Board**
- **16 Megabyte Addressing Capability**
- **Supports 8- or 16-bit Data Transfer and 24-bit Addressing**
- **Auxiliary Power Bus and Memory Protect Logic for Battery Back-Up RAM Requirements**

The iSBC 012CX, iSBC 010CX and iSBC 020CX RAM memory boards are members of Intel's complete line of iSBC memory and I/O expansion boards. The dual port feature of the CX series of RAM-boards allow access to the memory of both the MULTIBUS and iLBX bus interfaces.

In addition to the dual port features the "CX" series of RAM-boards provide Error Checking and Corrections Circuitry (ECC) which can detect and correct single bit errors and detect, but not correct, double and most multiple bit errors.

The iSBC 012CX board contains 512K bytes of read/write memory using 64K dynamic RAM components. The iSBC 010 CX and iSBC 020 CX boards contain 1024K and 2048K bytes of read/write memory using 256K dynamic RAM components.



231023-1

*The iSBC® 012CX, 010CX, and 020CX Boards are also manufactured under product code piSBC® and siSBC 012CX, 010CX, and 020CX by Intel Puerto Rico, Inc. and Intel Singapore, Ltd.

September 1989
Order Number: 231023-003

FUNCTIONAL DESCRIPTION

General

The iSBC 012CX, 010CX, and 020CX RAM boards are physically and electrically compatible with the MULTIBUS interface standard, IEEE-796, as outlined in the Intel MULTIBUS specification. In addition the CX series of RAM-boards are physically and electrically compatible with the iLBX bus (Local Bus Extension) interface as outlined in the Intel iLBX Specification (see Figure 1).

Dual Port Capabilities

The "CX" series of RAM-boards can be accessed by either the MULTIBUS interface or the iLBX interface (see Figure 2). Intel's iLBX interface is an unarbitrated bus architecture which allows direct transfer of data between the CPU and the memory boards

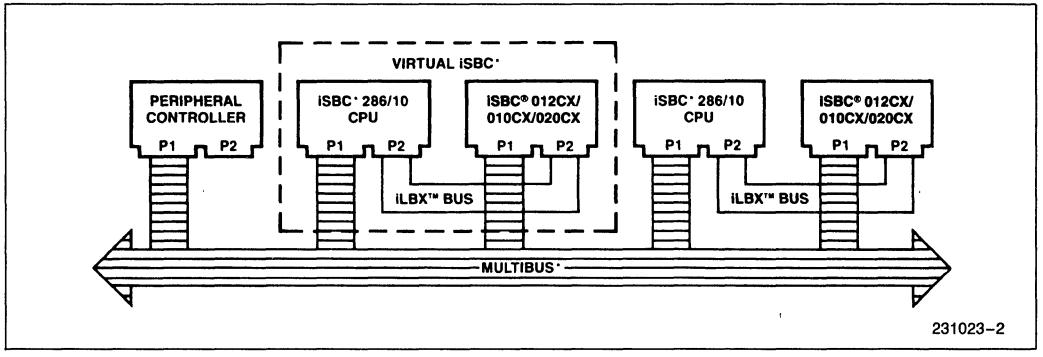


Figure 1. Typical iLBX™ System Configuration

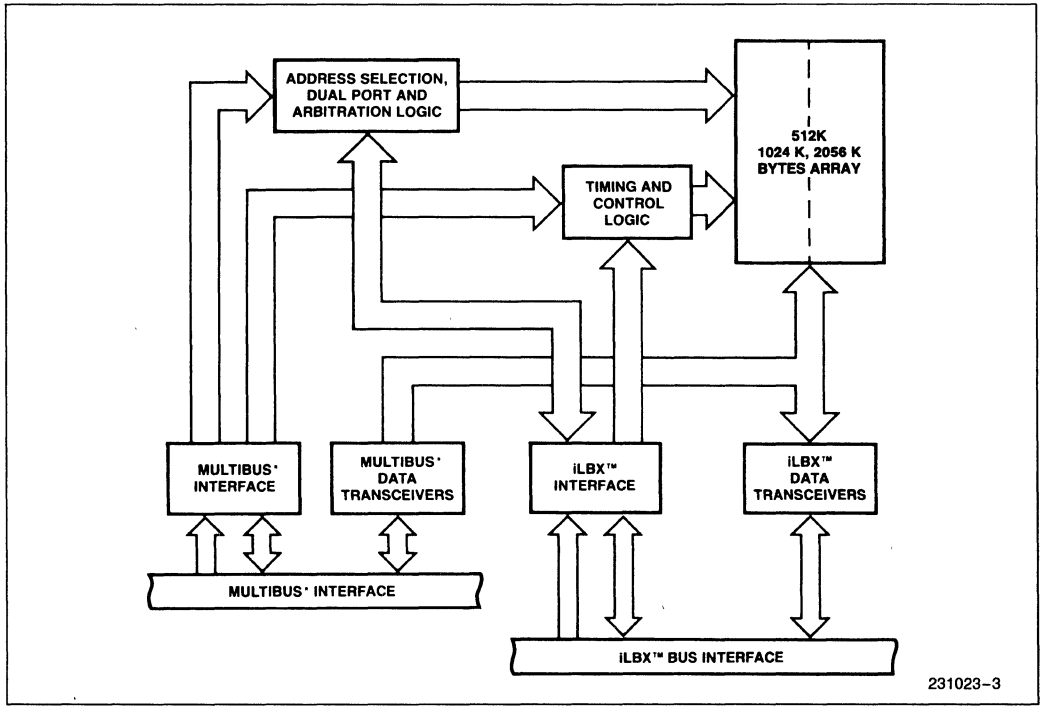


Figure 2. iSBC® 012CX/010CX/020CX Block Diagram

without accessing the MULTIBUS bus. Due to the unarbitrated nature of the iLBX interface significant improvements in memory access times result, typically a 2-6 Wait State improvement over MULTIBUS memory access.

System Memory Size

Maximum system memory size with this series of boards is 16 megabytes. Memory partitioning is independent for the MULTIBUS interface and the iLBX interface.

For MULTIBUS operations, on-board jumpers assign the board to one of four 4-megabyte pages. Each page is partitioned into 256 blocks of 16K bytes each. The smallest partition on any board in this series is 8K bytes. Jumpers assign the base address (lowest 16K block) within the selected 4-megabyte page.

The iLBX bus memory partitioning differs from the MULTIBUS bus partitioning in that the iLBX bus address space consists of 256 contiguous blocks of 64K bytes totaling 16 megabytes. As with the MULTIBUS bus partitioning, the base addresses are set with on-board jumpers.

Error Checking and Correcting (ECC)

Error checking and correction is accomplished with the Intel 8206 Error Checking and Correcting device. This ECC component, in conjunction with the ECC check bit RAM array, provides error detection and correction of single bit errors and detection only of double bit and most multiple bit errors. The ECC circuitry can be programmed via the Control Status Register (CSR) to various modes while error logging is supported by the Error Status Register (ESR). Both CSR and ESR communicate with the master CPU board through a single I/O port.

Battery Back-Up/Memory Protect

An auxiliary power bus is provided to allow separate power to the RAM array for systems requiring back-up of read/write memory. An active low TTL compatible memory protect signal is brought out on the auxiliary bus connector which, when asserted, disables read/write access to the RAM board. This input is provided for the protection of RAM contents during system power-down sequences.

SPECIFICATIONS

Word Size Supported

8- or 16-bits

Memory Size

524,288 bytes (iSBC 012CX board)
1,048,576 bytes (iSBC 010CX board)
2,097,152 bytes (iSBC 020CX board)

Access Times (All densities)

MULTIBUS® System Bus

Read/Full Write— 380 ns (max)
Write Byte — 530 ns (max)

iLBX™ Local Bus

Read/Full Write— 340 ns (max)
Write Byte — 440 ns (max)

Cycle Times (All densities)

MULTIBUS® System Bus

Read/Full Write— 490 ns (max)
Write Byte — 885 ns (max)

iLBX™ Local Bus

Read/Full Write— 375 ns
Write Byte — 740 ns

NOTE:
If an error is detected, read access time and cycle times are extended to 255 ns (max)

Memory Partitioning

Maximum System memory size is 16M Bytes for both MULTIBUS and iLBX BUS. MULTIBUS partitioning is by Page, Block and Base, while the iLBX BUS is by Block and Base only.

Page Address

MULTIBUS®— 0–4 megabytes; 4–8 megabytes, 8–12 megabytes; 12–16 megabytes

iLBX™ BUS — N/A

Base Address

MULTIBUS® System Bus— Any 16K byte boundary within the 4M-byte page.

iLBX™ Local Bus — Any 64K byte boundary selectable on board boundaries to 8M-bytes and some 64K-byte boundaries in the first megabyte. Others available if PAL programming is changed.

Power Requirements

Voltage—5 VDC ±5%

Product	Current	Standby (Battery Back-Up)
iSBC® 012CX Board	4.4A (typ.) 6.8A (max.)	2.2A (typ.) 2.4A (max.)
iSBC® 010CX Board	4.8A (typ.) 7.0A (max.)	2.1A (typ.) 2.3A (max.)
iSBC® 020CX Board	5.3A (typ.) 7.5A (max.)	2.2A (typ.) 2.4A (max.)

Environmental Requirements

Operating Temperature: 0°C to 55°C airflow of 200 linear feet per minute

Operating Humidity: To 90% without condensation

Physical Dimensions

Width: 30.48 cm (12 inches)

Height: 17.15 cm (6.75 inches)

Thickness: 1.27 cm (0.50 inches)

Weight: iSBC 012CX board: 6589 gm (23.5 ounces); iSBC 010CX board: 5329 gm (19.0 ounces); iSBC 020CX board: 6589 gm (23.5 ounces)

Reference Manuals

145158-003—iSBC® 028CX/iSBC® 056CX/iSBC® 012CX Hardware Reference Manual

144456-001—Intel iLBX™ 010CX, 020CX Specification

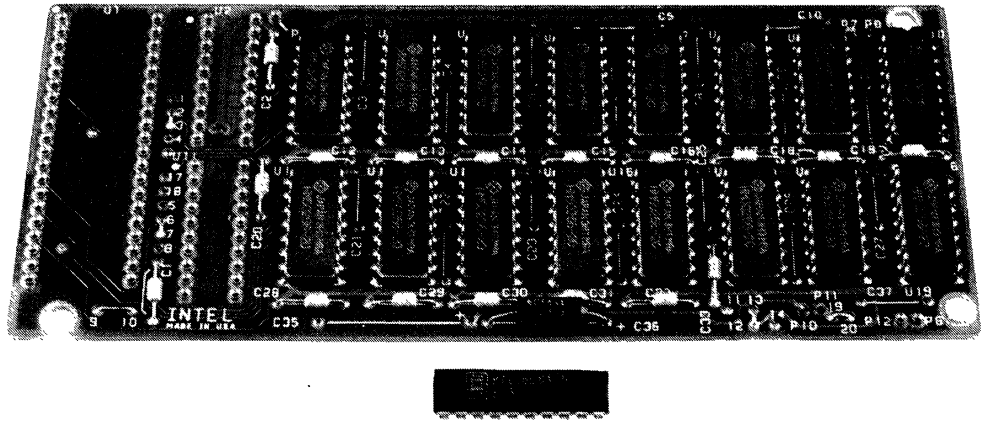
ORDERING INFORMATION

Part Number	Description
iSBC 012CX	512K byte RAM board with ECC
iSBC 010CX	1M byte RAM board with ECC
iSBC 020CX	2M byte RAM board with ECC

iSBC® 314*
512K BYTE RAM MULTIMODULE™ BOARD

- **On-Board Memory Expansion for the iSBC® 86/35 Single Board Computer**
- **iSBC 314 Module Provides 512K Bytes of Dual Port RAM Expansion for the iSBC 86/35 Board**
- **Reliable Mechanical and Electrical Interconnection**
- **Completes iSBC 86/35 Memory Array Providing a Full Megabyte Page of System Memory**
- **Increases System Throughput by Reducing Accesses to MULTIBUS® Global Memory**
- **Low Power Requirements**
- **Battery Backup Capability**

The iSBC® 314 512K byte RAM MULTIMODULE board provides simple, low cost expansion to double the on-board RAM capacity of the iSBC 86/35 Single Board Computer host to one megabyte. This RAM MULTIMODULE option offers system designers a simple, practical solution to expanding and improving the memory capability and performance of the iSBC 86/35 board. The iSBC 314 memory is configured on-board and can be accessed as quickly as the standard iSBC 86/35 memory, eliminating the need for accessing additional memory via the MULTIBUS system bus.



280000-1

*The iSBC® 314 Board is also manufactured under product code piSBC® 314 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280000-003

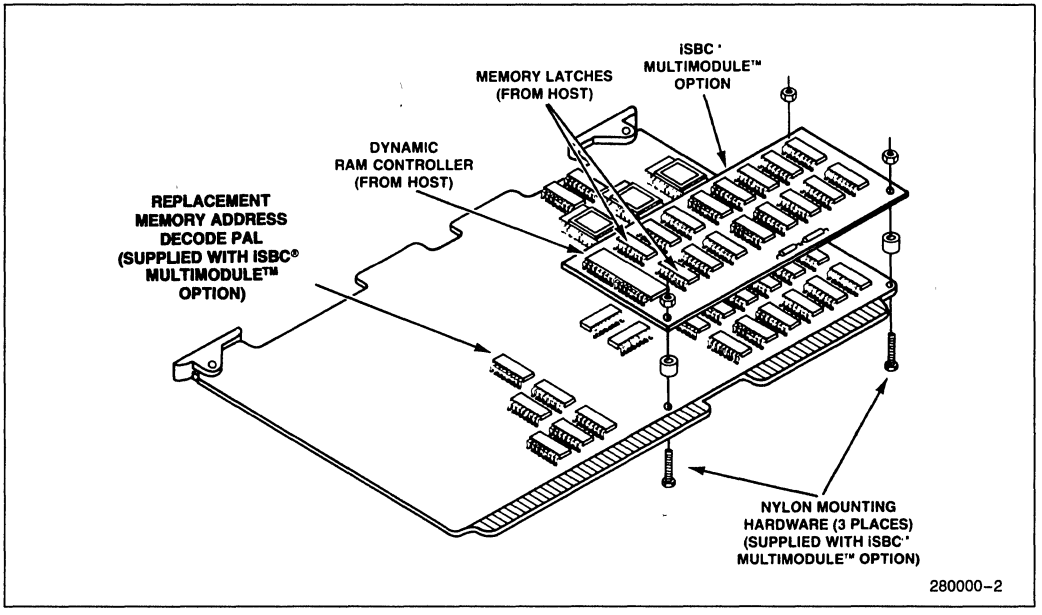


Figure 1. Installation of the MULTIMODULE™ RAM Module on the Host Single Board Computer

FUNCTIONAL DESCRIPTION

The iSBC 314 MULTIMODULE board measures 2.40″ by 5.75″ and mounts above the RAM array on the iSBC 86/35 Single Board Computer. The iSBC 314 board contains sixteen 256 Kbit x 1 dynamic RAM devices and three sockets; two for the memory latches and one for the Intel 8203 dynamic RAM controller. The addition of the iSBC 314 memory MULTIMODULE board to the iSBC 86/35 board makes possible a one megabyte single board solution.

To install the module, the latches and controller from the host iSBC 86/35 board, are removed and inserted into sockets on the iSBC 314 board. The module is then mounted onto the host board. Pins extending from the controller and latch sockets mate with device sockets underneath (see Figure 1). Additional pins mate to supply other signals to complete the electrical interface. The module is then secured at three additional points with nylon hardware to ensure the mechanical security of the assembly.

To complete the installation, one socketed PAL is replaced on the iSBC 86/35 board with the one supplied with the MULTIMODULE kit. This is the PAL which allows the host board logic to recognize its expanded on-board memory compliment.

SPECIFICATIONS

Word Size

8 or 16 bits (16-bit data paths)

Memory Size

512K bytes RAM

System Cycle Time (8 MHz, 2 Wait States)

750 ns (read); 750 ns (write)

NOTE:
1 wait state achieved with jumper change on iSBC 86/35 board.

Memory Addressing

iSBC 314 module with iSBC 86/35 board — 1M byte (total capacity); 0–FFFFFH. (See Figure 2, Memory Allocation)

Interface

The interface for the iSBC 314 MULTIMODULE board option is designed only for the iSBC 86/35 host board.

Wait-State Performance

A significant performance advantage of 2 wait-states is achieved when accessing memory on-board the iSBC 86/35 versus the performance of 6 wait-states when accessing memory off-board over the MULTIBUS. The iSBC 314 puts an additional 512K bytes of system memory on-board the iSBC 86/35 reducing the execution time by as much as 70%.

Memory Allocation

Segments of the combined host/MULTIMODULE RAM may be configured to be accessed either from off-board or on-board resources. The amount of memory allocated as either public or private resource may be configured in a variety of sizes. The address range boundaries for the 1 megabyte of RAM array of the iSBC 314 and iSBC 86/35 board combination are shown in Figure 2 for accesses from both on-board and off-board resources.

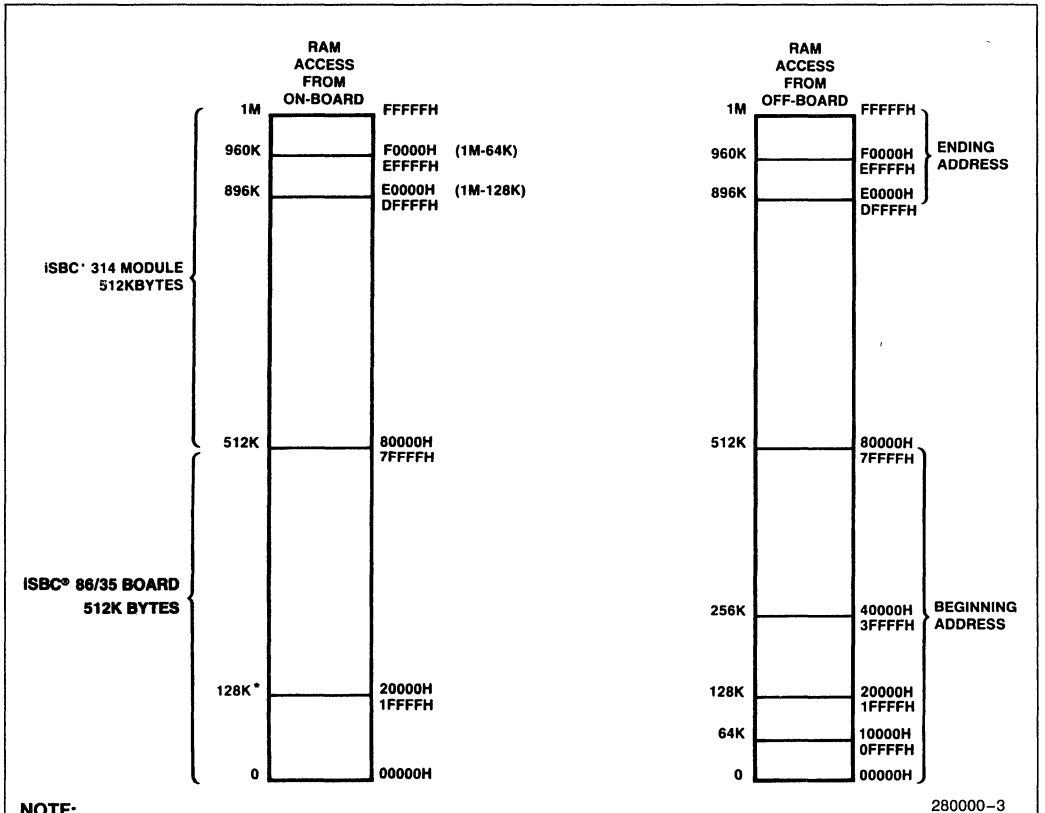
Auxiliary Power

The low power memory protection option included on the iSBC 86/35 board supports the iSBC 314 module.

Physical Characteristics

Width: 2.4 in. (6.10 cm)
Length: 5.75 in. (14.61 cm)
Depth*: 0.72 in. (1.83 cm)
Weight: 0.13 oz. (59g)

NOTE:
*Combined depth including host board.



NOTE:
All memory above this boundary may be disabled under software control to allow access to MULTIBUS® system bus.

280000-3

Figure 2. Address Range Selection

Electrical Characteristics

DC Power Requirements*

*Additional power required by the iSBC 314 MULTIMODULE is:

Typical: 60 mA @ +5V

Maximum: 140 mA @ +5V

Environmental Characteristics

Operating Temperature: 0°C to +55°C

Relative Humidity: to 90% (without condensation)

Reference Manual

All necessary documentation for the iSBC 314 MULTIMODULE board is included in the iSBC 86/35 Hardware Reference Manual (NOT SUPPLIED).

Order Number: 146245-002

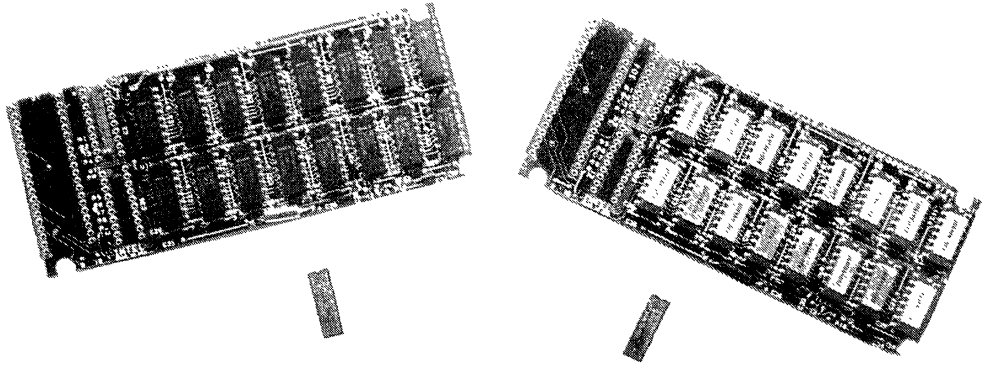
ORDERING INFORMATION

Part Number	Description
SBC 314	512K byte Memory MULTIMODULE option for iSBC 86/35 board

iSBC® 304*
128K BYTE RAM MULTIMODULE™ BOARD

- **iSBC® 304 Module Provides 128K Bytes of Dual Port RAM Expansion for the iSBC 86/30 or iSBC 86/35 Board**
- **On-board Memory Expansion Eliminates MULTIBUS® System Bus Latency and Increases System Throughput**

The iSBC® 304 module provides simple, low cost memory expansion for the iSBC 86/30 and iSBC 86/35 Single Board Computers. The iSBC 304 provides 128K bytes RAM expansion to the iSBC 86/35 giving a total capacity of 640K bytes RAM memory. The RAM MULTIMODULE option offers flexibility in defining and implementing Intel single board computer systems. RAM MULTIMODULES expand the memory configuration on-board, eliminating the need for accessing the additional memory via the MULTIBUS system bus.



210329-1

*The iSBC® 304 Board is also manufactured under product code piSBC® 304 by Intel Puerto Rico, Inc.

October 1989
Order Number: 210329-003

FUNCTIONAL DESCRIPTION

Each MULTIMODULE contains dynamic RAM devices and sockets for the dynamic RAM controller. To install the module, the latches and controller from the host CPU board are removed and inserted into sockets on the RAM MULTIMODULE. The module is then mounted onto the host board. Pins extending from the controller and latch sockets mate with device sockets underneath (see Figure 1). Additional pins mate to supply other signals to complete the electrical interface.

The module is then secured at three additional points with nylon hardware to ensure the mechanical security of the assembly.

To complete the installation, one socketed PROM is replaced on the host CPU board with the one supplied with the MULTIMODULE kit. This is the MULTIBUS address decode PROM which allows the host board logic to recognize its expanded on-board memory compliment.

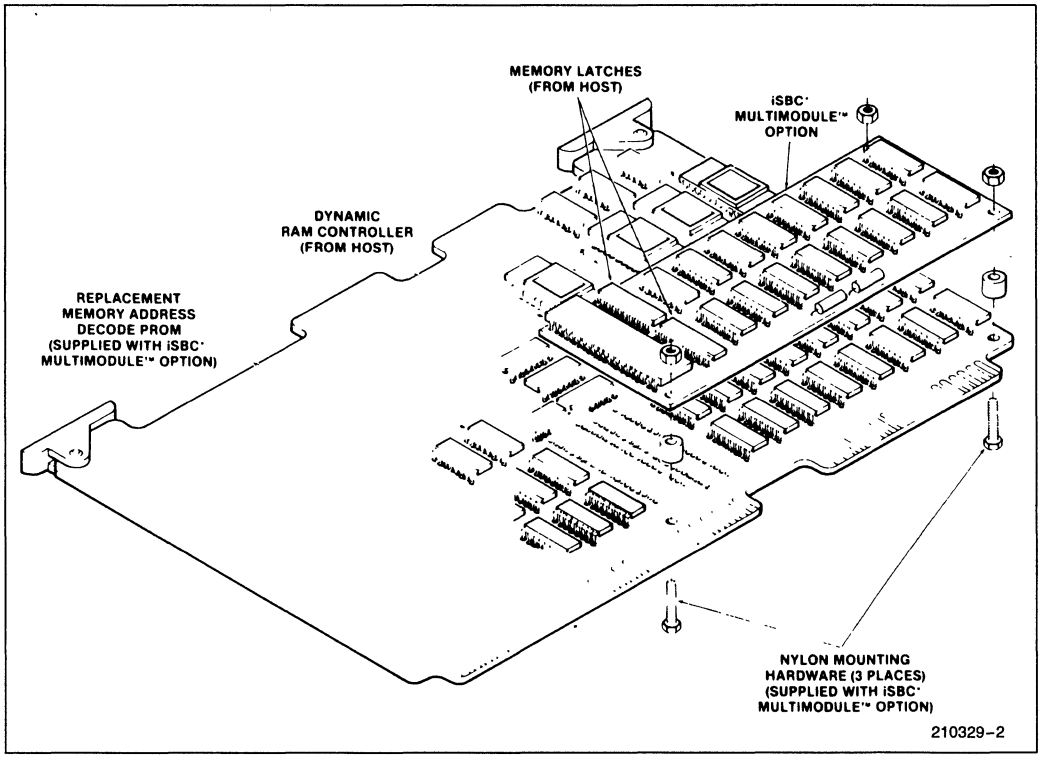


Figure 1. Installation of the MULTIMODULE™ RAM on the Host Single Board Computer

SPECIFICATIONS

Word Size

8 or 16 bits (16-bit data paths)

Memory Size

iSBC 304 Module—128K bytes RAM

Cycle Time

iSBC 304—700 ns (read); 700 ns (write)

Memory Addressing

CPU ACCESS

iSBC 304 (with iSBC 86/35)—640K bytes (total capacity); 0–9FFFFH (address range)

iSBC 304 (with iSBC 86/30)—256K bytes (total capacity); 0–3FFFFH (address range)

MULTIBUS® Access

Jumper selectable for any 32K byte boundary, but not crossing a 256K byte boundary.

Private Memory Allocation

Segments of the combined host/MULTIMODULE RAM memory may be configured as a private resource, protected from MULTIBUS system access. The amount of memory allocated as a private resource may be configured in increments of 25% of the total on-board memory ranging from 0% to 100%.

Auxiliary Power

The low power memory protection option included on the CPU host boards supports the RAM modules.

Physical Characteristics

Width: 2.4 in. (6.10 cm)
Height: 5.75 in. (14.61 cm)
Depth*: 0.72 in. (1.83 cm)
Weight: 0.13 oz. (59 g)

***NOTE:**
Combined depth including host board.

Electrical Characteristics

DC POWER REQUIREMENTS

iSBC 304: 640 mA at +15V incremental power

Environmental Characteristics

Operating Temperature: 0°C to 55°C

Relative Humidity: to 90% (without condensation)

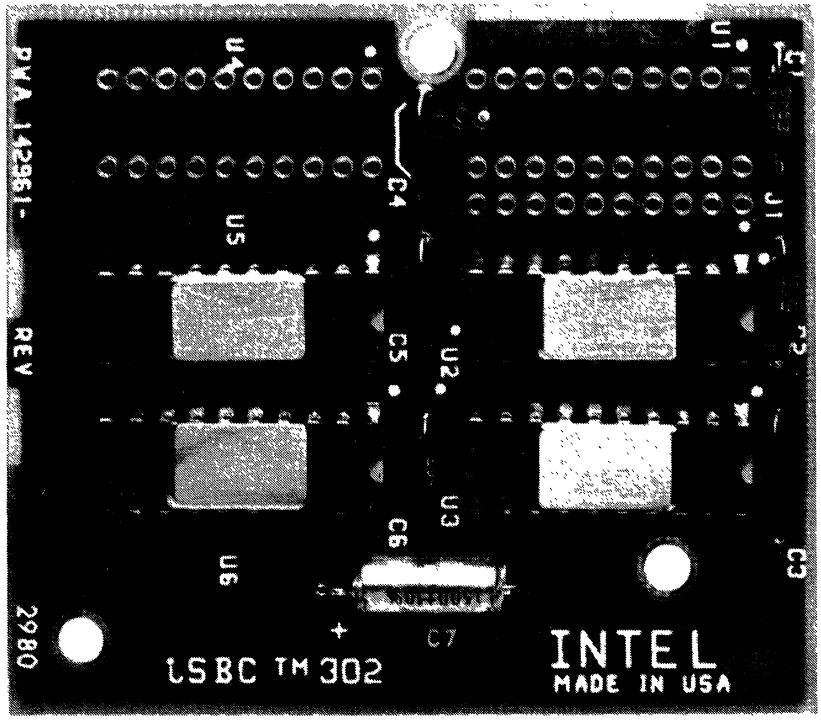
ORDERING INFORMATION

Order Code	Description
SBC304	128K MULTIMODULE option for iSBC 86/30 or iSBC 86/35 CPU boards

iSBC® 302*
8K BYTE RAM MULTIMODULE™

- **Expands On-Board Memory of the iSBC 86/05A and iSBC 88/25 Signal Board Computers**

The Intel iSBC 302 8K byte MULTIMODULE RAM provides simple, low cost expansion to double the RAM capacity on the iSBC 86/05A Single Board Computer to 16K bytes or increase RAM capacity on the iSBC 88/25 Single Board Computer to 12K bytes. Because the MULTIMODULE memory is configured on-board, it can be accessed as quickly as the standard on-board iSBC 86/05A or iSBC 88/25 memory, eliminating the need for accessing the additional memory via the MULTIBUS system bus.



280225-1

*The iSBC® 302 is also manufactured under product code piSBC® 302 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280225-002

FUNCTIONAL DESCRIPTION

The iSBC 302 MULTIMODULE and mounts above the RAM area on the iSBC 86/05A or iSBC 88/25 Single Board Computer. The iSBC 302 MULTIMODULE board contains four 4K x 4 static RAM devices and sockets for two of the RAM devices on the iSBC 80/05A board. With the iSBC 302 MULTIMODULE mounted on the iSBC 88/25 board, the two sockets on the iSBC 302 MULTIMODULE may be filled with 4K x 4 static RAMs. The two sockets on the iSBC 302 module have extended pins which mate with two sockets on the base board. Additional pins mate to the power supply and chip select lines to complete the electrical interface. The mechanical integrity of the assembly is assured with nylon hardware securing the module in two places.

SPECIFICATIONS

Word Size

8/16 bits

Memory Size

16,384 bytes of RAM

Cycle Time

Provides "no wait state"memory operations on the iSBC 86/05A board at 5 MHz or 8 MHz or the iSBC 88/25 at 5 MHz.

5 MHz cycle time — 800 ns

8 MHz cycle time — 500 ns

Memory Addressing

Memory addressing for the iSBC 302 MULTIMODULE board is controlled by the host board via the address and chip select signal lines.

Physical Characteristics

Width: 2.6 in. (6.60 cm)

Length: 2.3 in. (5.84 cm)

Height: 0.56 in. (1.42 cm) iSBC 302 board + Baseboard

Weight: 1.25 oz. (35 gm)

Electrical Characteristics

DC Power Requirements: 720 mA at +5V incremental power

Environmental Characteristics

Operating Temperature: 0°C to +55°C

Relative Humidity: to 90% (without condensation)

Reference Manuals

All necessary documentation for the iSBC 302 MULTIMODULE board is included in the CPU board Hardware Reference Manuals (NOT SUPPLIED).

iSBC 86/05A —Order No. 147162-002

iSBC 88/25 —Order No. 143825-002

ORDERING INFORMATION

Part Number	Description
SBC 302	8K byte RAM MULTIMODULE

iSBC® 301*
4K-BYTE RAM
MULTIMODULE™ BOARD

- **On-Board Memory Expansion to 8K Bytes for iSBC® 88/40A Single Board Computers**
- **Provides 4K Bytes of Static RAM Directly On-Board**

The Intel iSBC 301 4K-byte RAM MULTIMODULE Board provides simple, low cost expansion to double the RAM capacity on the iSBC 88/40A Single Board Computer to 8K bytes. Because memory is configured on-board, it can be accessed as quickly as the existing iSBC 88/40A memory, eliminating the need for accessing the additional memory via the MULTIBUS system bus.

280224-1

*The iSBC® Board is also manufactured under product code piSBC® 301 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280224-002

FUNCTIONAL DESCRIPTION

The iSBC 301 Board mounts above the RAM area on the iSBC 88/40A single board computer. It expands the on-board RAM capacity from 4K bytes to 8K bytes. The iSBC 301 MULTIMODULE board contains four 1K byte static RAM devices and a socket for one of the RAM devices on the iSBC 88/40A board. To install the iSBC 301 MULTIMODULE board, one of the RAMs is removed from the host board and inserted into the socket on the iSBC 301 board. The add-on board is then mounted into the vacated RAM socket on the host board. Pins extending from the RAM socket mate with the device's socket underneath (see Figure 1). Additional pins mate to the power supply and chip select lines to complete the electrical interface. The MULTIMODULE board is then secured at two additional points with nylon hardware to insure mechanical security of the assembly.

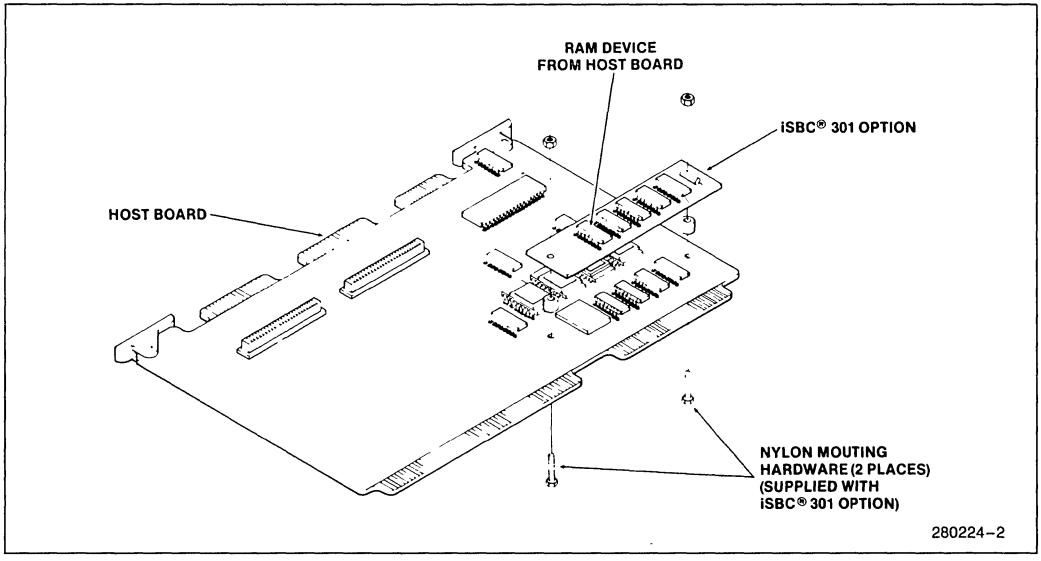


Figure 1. Installation of iSBC® 301 4K Byte RAM MULTIMODULE™ Board

SPECIFICATIONS

Word Size

8 bits

Memory Size

4096 bytes of RAM

Access Time

Read: 140 ns (from READ command)
200 ns (from ALE)
Write: 150 ns (from READ command)
190 ns (from ALE)

Memory Addressing

Memory addressing for the iSBC 301 4K-Byte-RAM MULTIMODULE Board is controlled by the host board via the address and chip select signal lines and is contiguous with the host board RAM.

iSBC 88/40A and iSBC 301 board: 00000-01FFF

Physical Characteristics

Width: 1.20 in. (3.05 cm)
Length: 3.95 in. (10.03 cm)
Height: 0.44 in. (1.12 cm) iSBC 301 Board
0.56 in. (1.42 cm)
iSBC 301 Board + host board
Weight: 0.69 oz. (19 gm)

Electrical Characteristics DC Power Requirements:

10 mA at +5 Volts incremental power

Environmental Characteristics

Operating Temperature: 0°C to +55°C
Relative Humidity: to 90% (without condensation)

Reference Manuals

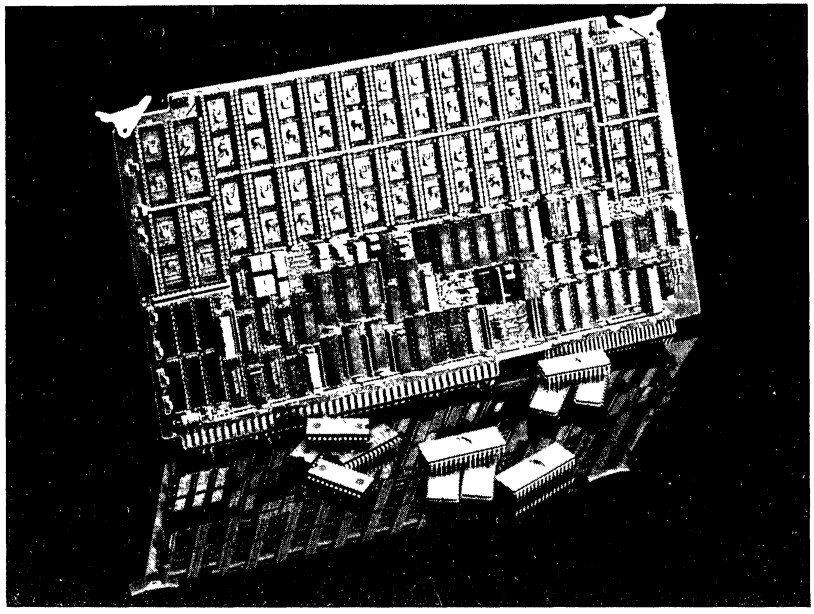
All necessary documentation for the iSBC 301 MULTIMODULE board is included in the CPU board Hardware Reference Manual (NOT SUPPLIED)

iSBC 88/40A—Order No. 147049-001

SPECIFICATIONS

Part Number	Description
SBC 301	4K Byte RAM MULTIMODULE Board

iSBC® 429 UNIVERSAL SITE MEMORY EXPANSION BOARD



CMOS MULTIBUS®I MEMORY EXPANSION BOARD SUPPORTS LATEST MEMORY TECHNOLOGY

The iSBC® 429 board provides a wide range of memory expansion capabilities for MULTIBUS designs. Up to 4 MBytes of memory can be installed using EPROM, Flash memory, SRAM, E²PROM or Static NVRAM.

The CMOS implementation of the iSBC 429 makes it ideal for low power applications.

All of Intel's Single Board Computers can communicate with the iSBC 429 using the MULTIBUS System bus. Alternatively, the iSBC 429 may be optionally configured to use the iLBX™ bus for faster access to the iSBC 186/03A, 286/10A, 286/12 series or 386/12 series of Single Board Computers.

FEATURES:

- Supports EPROM, Page Mode EPROM, E²PROM, Flash Memory, SRAM and Static NVRAM
- Thirty-two standard 32-pin JEDEC sites (supports both 28-pin and 32-pin devices) up to 4MByte capacity
- iLBX Bus or MULTIBUS Configurability
- Low power CMOS design
- Battery Backup/Memory Protect support
- Assignable anywhere within a 16 Megabyte address space on 4K byte boundaries

intel

September, 1989
Order Number 280668-001

iLBX Bus

The iSBC 429 board can be configured via jumpers to communicate with either the MULTIBUS interface or the iLBX Bus interface. Significant memory access time improvements can be realized using the iLBX Bus interface versus the MULTIBUS interface, due to its dedicated, unarbitrated architecture. Additional information on the iLBX Bus is available in the iLBX Specification, order number 145695-Rev. A.

CMOS DESIGN

For embedded control applications which are sensitive to power consumption, the iSBC 429 was designed with CMOS components and it will support many CMOS memory devices. Unpopulated, the iSBC 429 requires 5.25 watts at 5 volts.

FLASH MEMORY SUPPORT

The iSBC 429 board supports Intel's new CMOS Flash Memory devices. These new memory devices offer the most cost-effective and reliable alternative for updatable non-volatile memory. Memory contents can be erased and reprogrammed on-board during subassembly test, in-system during final test, and in-system after sale.

MEMORY BANKS

The thirty-two sites on the iSBC 429 board are partitioned into two banks of 16 sites each. Both banks are independently configurable to any of the device types supported on the board. Each bank can support up to 2 Megabytes using 27010 devices.

MEMORY ADDRESSING

The address space of each bank can be independently configured for starting address and size. The starting address can be on any 4 KByte boundary within the 16 MByte MULTIBUS address space. The size of each bank is a multiple of 64 KBytes.

MODE OF OPERATION

The iSBC 429 board can operate in one of two modes: the 8 bit only mode or the 8/16 bit mode. The 8 bit mode provides the most efficient memory configuration for systems handling 8 bit data only. The 8/16 bit mode allows the iSBC 429 board to be compatible with systems employing 8 bit and 16 bit masters. The mode of operation is selected by on-board jumpers and is available for both MULTIBUS and iLBX Bus configurations.

MEMORY ACCESS

The iSBC 429 board has jumper-selectable access times for each bank which allows the board to be tailored to the performance of the particular devices which are installed in the iSBC 429 board. The iSBC 429 accepts devices with an access time ranging from 150 ns with a minimum granularity of 99 ns and results in a board access time from 182 ns to 1667 ns. Each bank can be configured for access time.

INHIBITS

Inhibit signals are provided on the iSBC 429 board to allow ROM to overlay RAM for bootstrapping or diagnostic operations. Each bank of the iSBC 429 board can be overlayed with the system RAM by jumpers provided on the board. (i.e. If banks are overlapped, inhibits can be used to select the appropriate bank.)

BATTERY BACKUP

The iSBC 429 board supports battery backup operation via a connector on the board. An auxiliary power bus is provided to allow separate power to the memory array for systems requiring battery backup. Selection of this auxiliary power bus is made via jumpers on the board.

An active-low TTL compatible Memory Protect signal is brought out on the auxiliary connector which, when asserted, disables access to the memory array. This input is provided for the protection of Memory contents during system power-down sequences.

SPECIFICATIONS

MEMORY DEVICES SUPPORTED BY THE ISBC 429								
Size / Type	8K×8	16K×8	32K×8	64K×8	128K×8	256K×8	4×16K×8	8×16K×8
EPROM	2764	27128	27256	27512	27010	27020	—	—
ROM	✓[1]	✓	✓	✓	✓	✓	—	—
Page Mode EPROM	—	—	—	—	—	—	27513	27011
E²PROM[2]	2864A	—	—	—	—	—	—	—
Flash Memory[3]	27F64	—	27F256 28F256	—	—	—	—	—
Static NVRAM[4]	✓	—	✓	—	—	—	—	—
SRAM	✓	—	✓	—	—	—	—	—

1 "✓" denotes that the iSBC 429 board will support the device indicated, but that it is not currently available from Intel.
2 Five Volt only, Enhanced
3 12 Volt V_{pp} only
4 Static NVRAM devices exceed the height specification for MULTIBUS. The iSBC 429 will occupy more than one slot with these devices installed.

WORD SIZE

8 or 8/16 bits

MEMORY SIZE

Sockets are provided for up to thirty-two 32-pin or 28-pin devices which can provide up to 4 Megabytes of EPROM/ROM/SRAM/Flash Memory.

ACCESS TIME

Access time is jumperable from 182 ns to 1667 ns with a granularity of 99 ns to optimize performance for the devices which are installed and is equivalent for MULTIBUS and iLBX Bus.

POWER REQUIREMENTS

V_{cc} = 5 volts ±5%
V_{pp} = 12 volts ±5%
I_{cc} = 1.2 amps, maximum, without any memory devices in the board.

PHYSICAL CHARACTERISTICS

Width - 12.00 inches (30.48 cm)
Depth - 7.05 inches (17.91 cm)
Height - .5 inches (1.27 cm)

ENVIRONMENT

Operating Temperature - 0°C to +60°C (Convection cooling)
Relative Humidity - 90% non-condensing

ORDERING INFORMATION

PART NUMBER	*DESCRIPTION*
SBC 429	Universal Site Memory Expansion Board

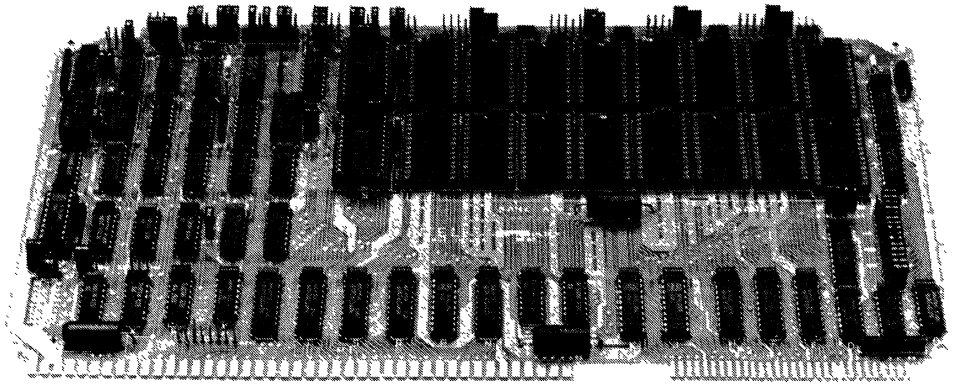
REFERENCE MANUAL

457317-001 - iSBC 429 Hardware Reference Manual (NOT SUPPLIED)

iSBC® 428 UNIVERSAL SITE MEMORY EXPANSION BOARD

- **Supports EPROM, ROM, E²PROM, SRAM, IRAM and NVRAM**
- **iLBX™ BUS or MULTIBUS® Selectable**
- **Provides Support for Battery Backup/ Memory Protect**
- **Sixteen 28-Pin Universal Sites**
- **Assignable Anywhere within a 16 Mbyte Address Space on 256K Byte Boundries**
- **Jumper Selectable Base Address on 4K Byte Boundaries**

The iSBC® 428 Universal Site Board is a member of Intel's complete line of Memory and I/O Expansion boards. The iSBC 428 Universal Site Memory Expansion Board interfaces directly to the iSBC 80, iSBC 88, or iSBC 86 Single Boad Computers via the MULTIBUS® System Bus to expand system memory requirements, while system memory requirements for iSBC 286 Single Board Computer can interface via either the MULTIBUS or the high speed iLBX™ Bus.



281013-1

October 1989
Order Number: 281013-001

FUNCTIONAL DESCRIPTION

General

The iSBC 428 board contains sixteen 28 pin sockets. The actual capacity of the board is determined by the type and quantity of components installed by the user. The iSBC 428 board is compatible with five different types and densities of devices: the 2K by 8 thru 64K by 8 EPROM/ROM devices, 2K by 8 thru 8K by 8 "Five Volt Only, Enhanced" E²PROM devices, 512 by 8 thru 16K by 8 NVRAM (Non-Volatile RAM) devices, 2K by 8 thru 32K by 8 SRAM devices, and 8K by 8 IRAM (Integrated RAM) devices. In addition the board can be accessed by either the MULTIBUS System Bus or Intel's new high speed iLBX Bus.

iLBX™ Bus

The iSBC 428 board can be configured via jumpers to communicate with either the MULTIBUS interface or the iLBX Bus interface. Significant memory access time improvements can be realized over the iLBX Bus interface (versus the MULTIBUS interface) due to its dedicated, unarbitrated architecture. Additional information on the iLBX Bus is available in the iLBX Specification #144456.

Memory Banks

The sixteen sites on the iSBC 428 board are partitioned into two banks of 8 sites each. Within each bank the 8 sites are futher partitioned into 2 groups of 4 sites each. Each group of 4 sites is configurable to each of the six device types described above via a "Configurator". The "Configurator" is an arrangement of push-on jumpers which configures each of the four groups of 4 sites. Within each bank devices of the same density must reside and within each group devices of the same type must reside (i.e., SRAM or EPROM).

Memory Addressing

Addressing of the iSBC 428 board is by pages. There are 64–256K pages which are jumpers selectable. Each of the two banks are independently addressable and can reside in any page. Actual beginning and ending addresses within a page are a function of the actual device size and, as with the pages, are determined by jumpers. Because of the paging based memory addressing architecture more than one iSBC 428 board can be placed in a system.

Mode of Operation

The iSBC 428 board can operate in one of two modes: the 8 bit only mode or the 8/16 bit mode. The 8 bit mode provides the most efficient memory configuration for systems handling 8 bit data only. The 8/16 bit mode allows the iSBC 428 board to be compatible with systems employing 8 bit and 16 bit masters. The mode of operation is selected by on board jumpers and is available for both MULTIBUS and iLBX Bus configurations.

Memory Access

The iSBC 428 board has jumper selectable access time, which allows the board to be tailored to the performance of the particular devices which are installed in the iSBC 428 board. The board can be configured via jumpers to accept devices with an access time range of 50 ns to 500 ns with a granularity of 50 ns and results in a board access time from 225 ns to 775 ns.

Interrupt

The iSBC 428 board has the capability of generating an interrupt for the write and erase operations of E²PROMs. The interrupt can be configured in two ways: one, to signal completion of the E²PROM write cycle, or two, allow polling by the system to determine the status of the E²PROM during the write programming time.

Inhibits

Inhibits are provided on the iSBC 428 board to allow ROM to overlay RAM for bootstrapping or diagnostic operations. Each bank of the iSBC 428 board can be overlayed with the system RAM by jumpers provided on the board.

Battery Backup

The iSBC 428 board supports battery backup operation via a connector on the board. An auxiliary power bus is provided to allow separate power to the memory array for systems requiring battery backup. Selection of this auxiliary power bus is made via jumpers on the board.

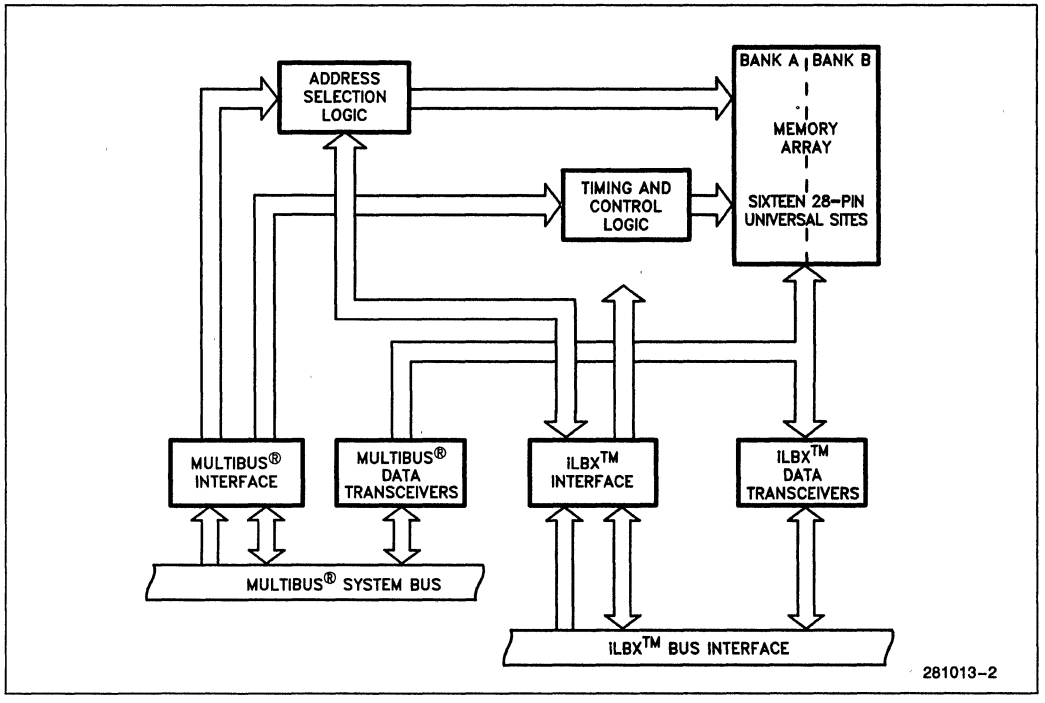
An active-low TTL compatible Memory Protect signal is brought out on the auxiliary connector which, when asserted, disables access to the memory array. This input is provided for the protection of Memory contents during system power-down sequences.

Devices Supported

Listed below are the current and future devices supported by the iSBC 428 board.

	Size							
Type	**512 x 8**	**2K x 8**	**4K x 8**	**8K x 8**	**16K x 8**	**32K x 8**	**64K x 8**	**Comments**
EPROM	—	2716	2732A	2764	27128	27256	27512	—
ROM	—	X	X	X	X	X	X	—
EEPROM	—	2817A	X	X	X	X	—	5V, Enhanced
SRAM	—	X	X	X	X	X	—	NMOS & CMOS
NVRAM	—	X	X	X	—	—	—	—
IRAM	—	—	—	2186	—	X	—	—

X—Denotes that the iSBC 428 board will support the device indicated but that it is not currently available from Intel.



iSBC® 428 Block Diagram

SPECIFICATIONS

Word Size

8 or 8/16 bits

Memory Size

Sockets are provided for up to sixteen 28 pin devices which can provide up to 512K bytes of EPROM/ROM/SRAM.

Access Time

Jumperable from 225 ns to 775 ns with a granularity of 50 ns and is equivalent for both MULTIBUS and the iLBX Bus.

Power Requirements

V_{CC} = 5 volts ±5%

I_{CC} = 2.0 amps, maximum, without any memory devices in the board.

Physical Characteristics

Length: 30.48 cm (12 inches)
Width: 17.15 cm (7.05 inches)
Depth: 1.27 cm (0.5 inches)

Environment

Operating Temperature: 0°C to +55°C
Relative Humidity: 90% non-condensing

ORDERING INFORMATION

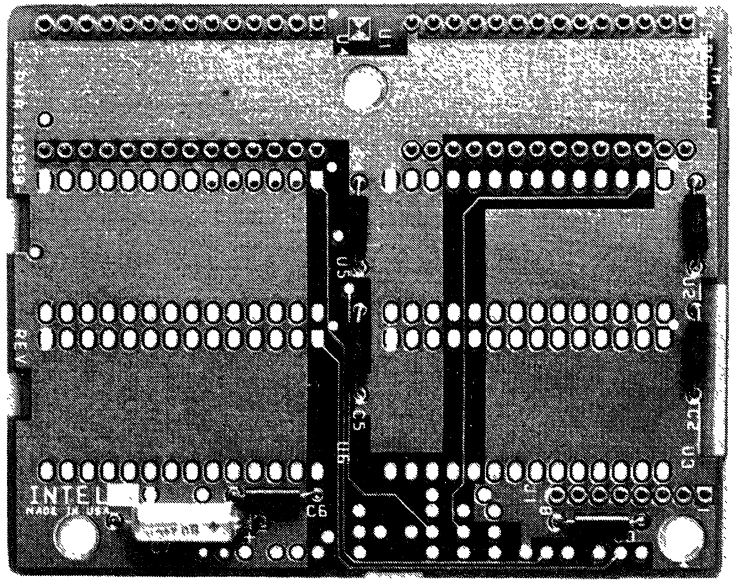
Order Code	Description
SBC428	Universal Site Memory Expansion Board

iSBC® 341
28-PIN MULTIMODULE™ EPROM

- **On-board Memory Expansion for iSBC® 86/05A, iSBC 88/25, iSBC 186/03A, iSBC 286/10A, iSBC 286/12 Series, and iSBC 88/40A Microcomputers**
- **Supports JEDEC 24/28-Pin Standard Memory Devices, Including EPROMs, Byte-Wide RAMs, and E²PROMs**
- **Sockets for Up to 256K Bytes of Expansion with Intel 27512 EPROMs**
- **On-Board Expansion Provides "No Wait State" Memory Access with Selected Devices**
- **Simple, Reliable Mechanical and Electrical Interface**

The iSBC 341 28-pin MULTIMODULE EPROM board provides simple, low-cost expansion of the on-board EPROM capacity of the iSBC 86/05A, the iSBC 88/25, iSBC 186/03A, iSBC 286/10A, iSBC 286/12 Series Single Board Computers and the iSBC 88/40A Measurement and Control Computer. Four additional 28-pin sockets support JEDEC 24/28-pin standard devices, including EPROMs, byte-wide static and psuedo-static RAMs.

The MULTIMODULE expansion concept provides the optimum mechanism for incremental memory expansion. Mounting directly on the microcomputer, the benefits include low cost, no additional power requirements beyond the memory devices, and higher performance than MULTIBUS-based memory expansion.



280214-1

October 1988
Order Number: 280214-001

FUNCTIONAL DESCRIPTION

The iSBC 341 28-pin MULTIMODULE EPROM option effectively doubles the number of sockets available for EPROM on the base microcomputer board on which it is mounted. The iSBC 341 board contains six 28-pin sockets. Two of the sockets have extended pins which mate with two of the sockets on the base board. Two of the EPROMs which would have been inserted in the base board are then reinserted in the iSBC 341 sockets. Additional interface pins also connect chip select lines and power. The mechanical integrity of the assembly is assured with nylon hardware securing the unit in two places.

Through its unique interface, the iSBC 341 board can support 8- or 16-bit data paths. The data path width is determined by the base board—being 8 bits for the iSBC 88/40A and iSBC 88/25 microcomputers, and 8/16 bits for the iSBC 86/05A, iSBC 186/03A, iSBC 286/10A, and iSBC 286/12 Series Single Board Computers.

SPECIFICATIONS

Word Size

8 or 8/16 bits (determined by data path width of base board).

Memory Size

256K bytes with available technology (JEDEC standard defines device pin-out to 512-bit devices).

Device Size (Bytes)	EPROM Type	Max iSBC® 341 Capacity (Bytes)
2K x 8	2716	8K
4K x 8	2732A	16K
8K x 8	2764	32K
16K x 8	27128	64K
32K x 8	27256	128K
64K x 8	27512	256K

Access Time

Varies according to base board and memory device access time. Consult data sheet of base board for details.

Memory Addressing

Consult data sheet of base board for addressing data.

POWER REQUIREMENTS

Devices(1)	Max Current @ 5V ±5%
2716	420 mA
2732A	600 mA
2764	600 mA

NOTE:
1. Incremental power drawn from host board for four additional devices.

Auxiliary Power

There are no provisions for auxiliary power (battery backup) on the iSBC 341 option.

Physical Characteristics

Width: 3.4 in. (8.64 cm)

Length: 2.7 in. (6.86 cm)

Height: 0.78 in. (1.98 cm)*

Weight: 5 oz. (141.5 gm)

*Includes height of mounted memory devices and base board.

All necessary mounting hardware (nylon screws, spacers, nuts) is supplied with each kit.

Environmental Characteristics

Operating Temperature: 0°C to +55°C

Relative Humidity: to 90% (without condensation)

Reference Manuals

All necessary documentation for the iSBC 341 module is included in the CPU board Hardware Reference Manuals (NOT SUPPLIED)

iSBC 186/03A — Order No. 148060-001
iSBC 86/05A — Order No. 147162-002
iSBC 88/25 — Order No. 143825-002
iSBC 88/40A — Order No. 147049-001
iSBC 286/10A — Order No. 147532-001
iSBC 286/12 — Order No. 147533-001

Manuals may be ordered from any Intel sales representative, distributor office, or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

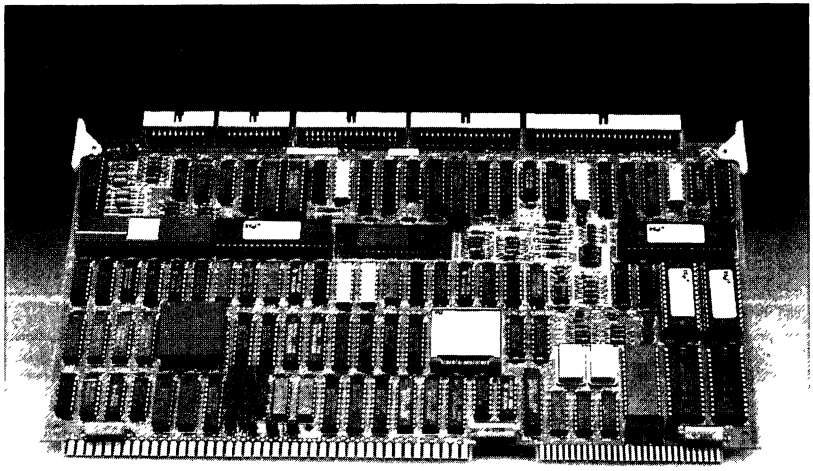
ORDERING INFORMATION

Part Number	Description
SBC 341	28-Pin MULTIMODULE EPROM

MULTIBUS® I Peripheral Controllers

10

iSBC® 221* PERIPHERAL CONTROLLER



MULTIBUS®I CONTROLLER FOR HIGH PERFORMANCE, HIGH CAPACITY PERIPHERALS

The iSBC® 221 is a multifunction peripheral controller that provides access to high-performance, high-capacity disk drives (hard, flexible, and streaming tape). I/O bound applications and/or those requiring high disk capacity will especially benefit from this fast, reliable controller. The iSBC 221 can replace the Intel iSBC 214 without changing the operating system device driver or the disk drives.

FEATURES:

- Support for ESDI and ST506/412 hard disk drives, SA 45X/46X/475 flexible disk drives, and QIC-02 streaming tape drives
- Multiple track caching via 128K on-board data buffer
- Dual bus structure
- 10 MHz 80186 Microprocessor
- Mirror backup/restore between tape and hard drive
- On-board self-test diagnostics
- Error-checking and correcting code logic
- Support for 4,096 cylinders and 16 heads

intel

* The iSBC® 221 is also manufactured under product code piSBC® 221 by Intel Puerto Rico, Inc

September, 1989
Order Number 280410-002

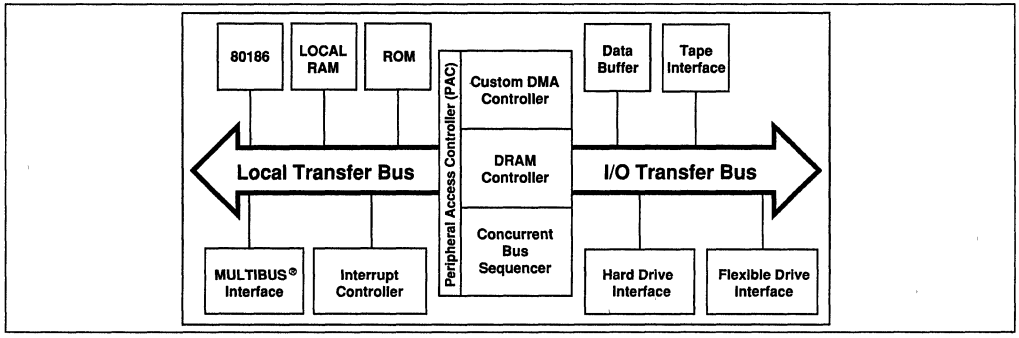


Figure 1: Simplified Block Diagram of iSBC 221 Peripheral Controller

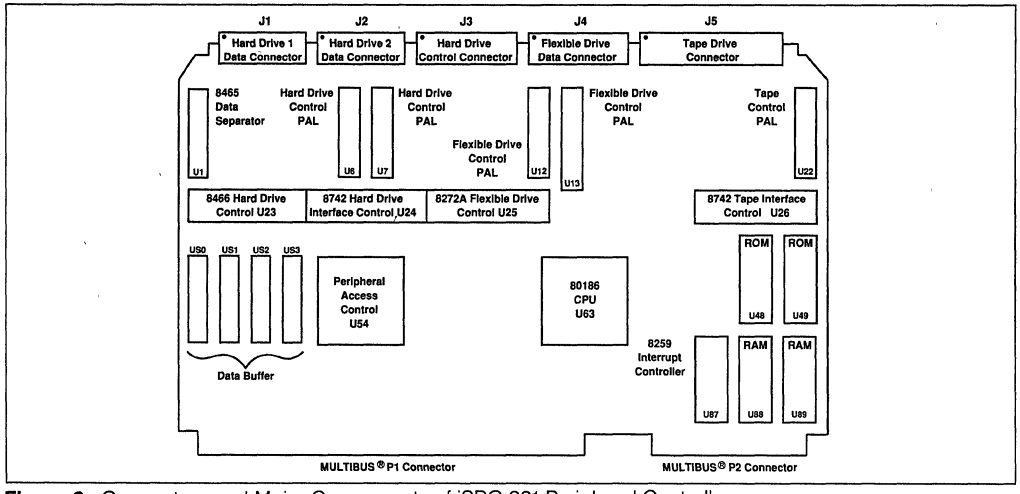


Figure 2: Connectors and Major Components of iSBC 221 Peripheral Controller

INTERFACE SUPPORT

	Interface	Transfer Rate
Hard Disk (up to 2)	EDSI	up to 10 Mbit/sec
	ST506/412	5 Mbit/sec
Flexible Disk (up to 4)	SA 475	250/500 Kbit/sec
	SA 460/465	125/250 Kbit/sec
	SA 450/455	125/250 Kbit/sec
Streaming Tape (up to 4)	QIC-02	90/112.5 Kbit/sec (typical)

HIGH PERFORMANCE

I/O-bound applications are accelerated by the combination of the ESDI standard, a 128K data buffer, a 10 MHz 80 186 microprocessor, and a dual bus structure. The dual bus structure allows the iSBC 221 to concurrently transfer data between the controller and the peripheral devices and between the controller and the host.

WORLDWIDE SERVICE AND SUPPORT

Intel provides support for board repair or on-site service. Development support options include phone support, subscription service, on-site consulting and customer training.

QUALITY AND RELIABILITY TESTING

The iSBC 221 is designed and manufactured in accordance with Intel's high quality standards. We then verify quality through rigorous testing in our state-of-the-art Environmental Test Laboratory.

SPECIFICATIONS

PHYSICAL CHARACTERISTICS

Length:	12.0in (304.8 mm)
Width	6.75 in. (171.5 mm)
Approximate Weight:	24 oz (680 g)

POWER REQUIREMENTS

+5 VDC @ 4.5A maximum
±12V @ 0.5A

ENVIRONMENTAL REQUIREMENTS

Operating Temperature:	0 to 55° @ 200 LFM
Non-operating:	−55 to 85°C
Humidity:	0 to 90% non-condensing

REFERENCE MANUAL

iSBC 221 Peripheral Controller User's Guide Order #451210

DEVICE DRIVERS

Check the latest release of the following operating systems for details:

iRMX I	XENIX*
iRMX II	UNIX*
iRMX III	

* XENIX is a trademark of Microsoft, Inc.
UNIX is a trademark of American Telephone and Telegraph, Inc.

ORDERING INFORMATION

Order Code	Description
SBC221	Peripheral Controller

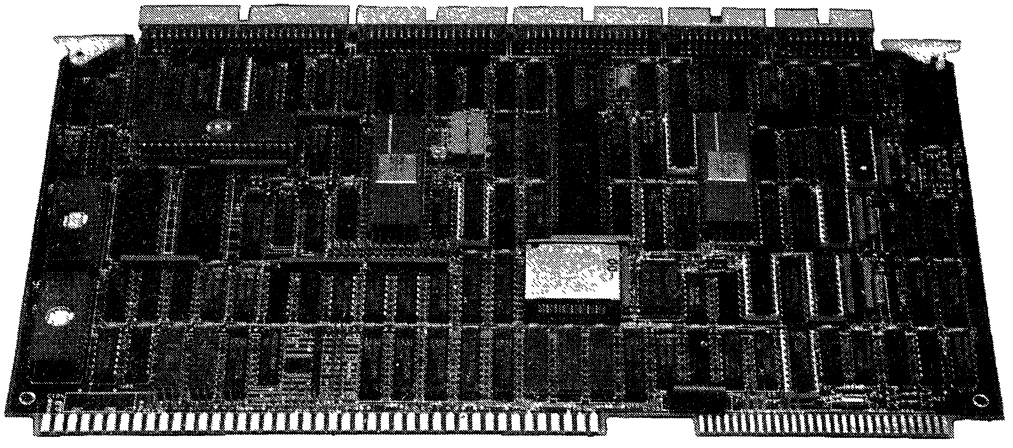
iSBC® 214*
PERIPHERAL CONTROLLER SUBSYSTEM

- **Based on the 80186 Microprocessor**
- **Controls up to Two ST506/412 5¼" Hard Disk Drives**
- **Controls up to Four Single/Double Sided and Single/Double Density 5¼" Flexible Disk Drives**
- **Controls up to Four QIC-02 Streaming Tape Drives**
- **Supports 20 or 24-Bit Addressing**
- **On-Board Diagnostics and Hard Disk ECC**
- **Incorporates Track Caching to Reduce Winchester Disk Access Times**
- **iRMX™ and UNIX* Operating System Support**

The iSBC 214 Subsystem is a single-board, multiple device controller that interfaces standard MULTIBUS® systems of three types of magnetic storage media. The iSBC 214 Peripheral Controller Subsystem supports the following interface standards: ST506/412 (Hard Disk), SA 450/460 (Flexible Disk), and QIC-02 (¼" Streaming Tape).

The board combines the functionality of the iSBC 215 Generic Hard Disk Controller and the iSBC 213 Data Separator, the iSBX™ 218A Flexible Disk Controller, and the iSBX 217C ¼" Tape Drive Interface Module. The iSBC 214 Subsystem emulates the iSBC 215G command set, allowing users to avoid rewriting their software.

The iSBC 214 Peripheral Controller Subsystem offers a single slot solution to the interface of multiple storage devices, thereby reducing overall power requirements, increasing system reliability, and freeing up backplane slots for addtional functionality. In addition, the new iSBC 214 Subsystem can be placed in a 16 Megabyte memory space.



280089-1

*UNIX is a trademark of Bell Laboratories.
*The iSBC® 214 is also manufactured under product code piSBC® 214 by Intel Puerto Rico, Inc.

October 1989
Order Number: 280089-002

The iSBC 214 represents a new Peripheral Controller Subsystem architecture which is designed around a dual bus structure and supported by real-time, multitasking firmware. The 80186 controls the local bus and manages the interface between the MULTIBUS and the controller. It is responsible for high speed data transfers of up to 1.6 megabytes per second between the iSBC 214 Subsystem and host memory. The 80186 and the multitasking firmware decode the command request, allocate RAM buffer space, and dispatch the tasks.

A second bus, the I/O Transfer Bus, supports data transfers between the controller and the various peripheral devices. It is this dual bus system that allows the iSBC 214 Subsystem to provide simultaneous data transfers between the controller and the storage devices, and between the controller and the MULTIBUS. (See Figure 1).

The iSBC 214 Subsystem implements an intelligent track caching scheme through dynamic allocation of buffer space. This provides reduced access times to the Winchester disk and improved system performance. Operating systems with file management designed to handle sequential data can be supplied directly from the cache without incremental access to the disk.

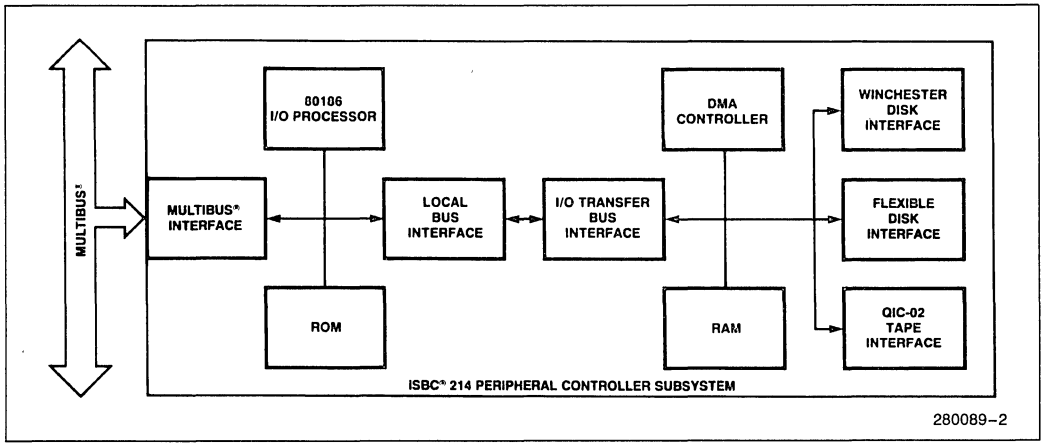


Figure 1. Block Diagram iSBC® 214 Peripheral Controller Subsystem

SPECIFICATIONS

Compatibility

CPU—any MULTIBUS computer or system mainframe.

Hard disk—Any ST506/412 compatible, 5.25″ disk drive.

Flexible disk—Any SA450/460 compatible, 5.25″ disk drive.

Tape drive—Any QIC-02 compatible, .25″ streaming tape drive.

Controller-to-drive cabling and connectors are not supplied with the controller. Cables can be fabricated with flat cable and commercially-available connectors as described in the iSBC 214 Hardware Reference Manual.

Physical Characteristics

Width: 6.75 in. (17.15 cm)
Height: 0.5 in. (1.27 cm)
Length: 12.0 in. (30.48 cm)
Shipping Weight: 19 oz. (540 g)

Ordering Information

Order Code	Description
SBC214	Peripheral Controller Subsystem.

Electrical Characteristics

Power Requirements: +5 VDC @ 4.5A max.

Environmental Characteristics

Temperature: 10°C to 55°C with airflow of 200 linear feet per minute (operating); −55°C to +85°C (non-operating).

Humidity: Up to 90% relative humidity without condensation (operating); all conditions without condensation or frost (non-operating).

Mounting: Occupies one slot or SBC system chassis or cardcage/backplane.

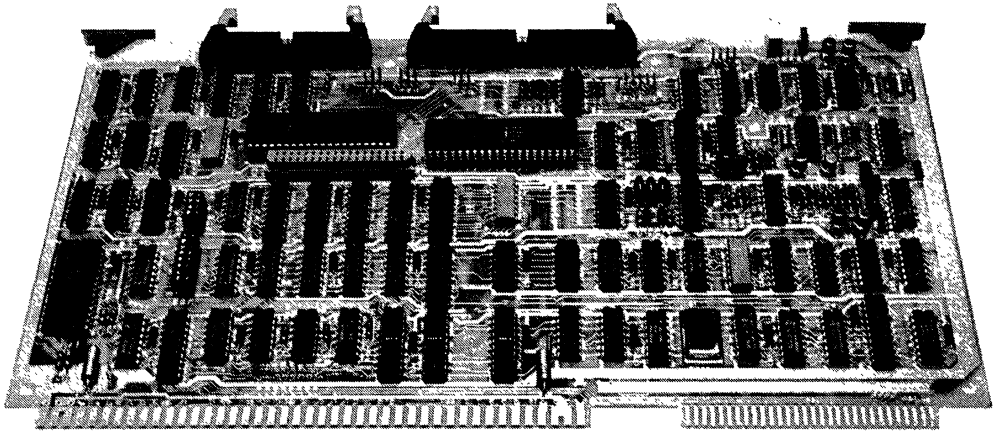
Reference Manual

134910: iSBC 214 Peripheral Controller Subsystem Hardware Reference Manual (not supplied). Reference Manual may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, CA 95051.

iSBC® 208*
FLEXIBLE DISKETTE CONTROLLER

- **Compatible with All iSBC® 80, iSBC 86, and iSBC 88 Single Board Computers**
- **Controls Most Single and Double Density Diskette Drives**
- **On-Board SBX Bus for Additional Functions**
- **User-Programmable Drive Parameters allow Wide Choice of Drives**
- **Phase Lock Loop Data Separator Assures Maximum Data Integrity**
- **Read and Write on Single or Multiple Sectors**
- **Single +5V Supply**
- **Capable of Addressing 16M Bytes of System Memory**

The Intel iSBC 208 Flexible Disk Controller is a diskette controller capable of supporting virtually any soft-sectored, double density or single density diskette drive. The standard controller can control up to four drives with up to eight surfaces. In addition to the standard IBM 3740 formats and IBM System 34 formats, the controller supports sector lengths of up to 8192 bytes. The iSBC 208 board's wide range of drive compatiblity is achieved without compromising performance. The operating characteristics are specified under user program control. The controller can read, write, verify, and search either single or multiple sectors. Additional parallel or serial I/O capability can be added to the iSBC 208 board via the SBX bus (IEEE 959) connector.



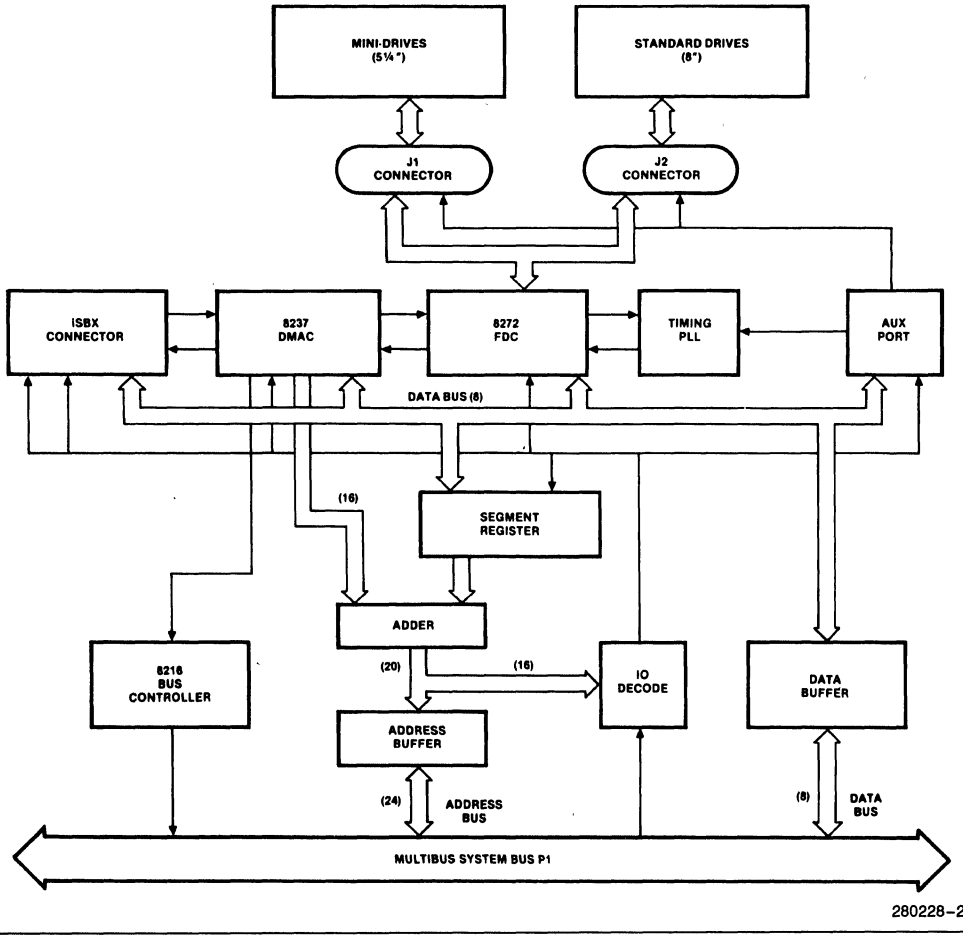
280228-1

*The iSBC® 208 is also manufactured under product code piSBC® 208 or siSBC® 208 by Intel Puerto Rico, Inc., or Intel Singapore, Ltd.

September 1989
Order Number: 280228-002

BLOCK DIAGRAM

iSBC® 208 Flexible Disk Controller Block Diagram



SPECIFICATIONS

Compatibility

CPU — Any MULTIBUS computer or system main frame

Devices— Double or single density standard (8") and mini (5¼") flexible disk drives. The drives may be single or double sided. Drives known to be compatible are:

Standard (8")		Mini (5¼")	
Caldisk	143M	Shugart	450 SA 400
Remex	RFD 4000	Micropolis	1015-IV
Memorex	550	Pertec	250
MFE	700	Siemens	200-5
Siemens	FDD 200-8	Tandon	TM-100
Shugart	SA 850/800	CDC	9409
Pertec	FD 650	MPI	51/52/91/92
CDC	9406-3		

Diskette— Unformatted IBM Diskette 1 (or equivalent single-sided media); unformatted IBM Diskette 2D (or equivalent double-sided)

Equipment Supplied

iSBC 208 Controller
Reference Schematic
Controller-to-drive cabling and connectors are not supplied with the controller. Cables can be fabricated with flat cable and commercially-available connectors as described in the iSBC 208 Hardware Reference Manual

Physical Characteristics

Width: 6.75 inches (17.15 cm)
Height: 0.5 inches (1.27 cm)
Length: 12.0 inches (30.48 cm)
Shipping Weight: 1.75 pounds (0.80 Kg)
Mounting: Occupies one slot of MULTIBUS system chassis. With an SBX board mounted, vertical height increases to 1.13 inches (2.87 cm).

Electrical Characteristics

Power Requirements: +5 VDC @ 3.0A

Data Organization and Capacity

Standard Size Drives

	Double Density						Single Density					
	IBM System 34			Non-IBM			IBM System 3740			Non-IBM		
Bytes per Sector	256	512	1024	2048	4096	8192	128	256	512	1024	2048	4096
Sectors per Track	26	15	8	4	2	1	26	15	8	4	2	1
Tracks per Diskette	77			256			77			256		
Bytes per Diskette (Formatted, per diskette surface)	512,512 (256 bytes/sector) 591,360 (512 bytes/sector) 630,784 (1024 bytes/sector)			630,784			256,256 (128 byte/sector) 295,680 (256 bytes/sector) 315,392 (512 bytes/sector)			315,392		

Drive Characteristics

	Standard Size	Mini Size
	Double/Single Density	Double/Single Density
Transfer Rate (K bytes/s)	62.5/31.25	31.25/15.63
Disk Speed (RPM)	360	300
Step Rate Time (Programmable)	1 to 16 ms/track in 1 ms increments	2 to 32 ms/track in 2 ms increments
Head Load Time (Programmable)	2 to 254 ms in 2 ms increments	4 to 508 ms in 4 ms increments
Head Unload Time (Programmable)	16 to 240 ms in 16 ms increments	32 to 480 ms in 32 ms increments

Environmental Characteristics

Temperature: 0°C to 55°C (operating); −55°C to +85°C (non-operating)

Humidity: Up to 90% Relative Humidity without condensation (operating); all conditions without condensation or frost (non-operating)

Reference Manual

143078— iSBC 208 Flexible Disk Controller Hardware Reference Manual (NOT SUPPLIED).

Reference manuals may be ordered from any Intel sales representative, distributor office, or from Intel Literature Department, 3065 Bowers Avenue, Santa, Clara, CA 95051.

ORDERING INFORMATION

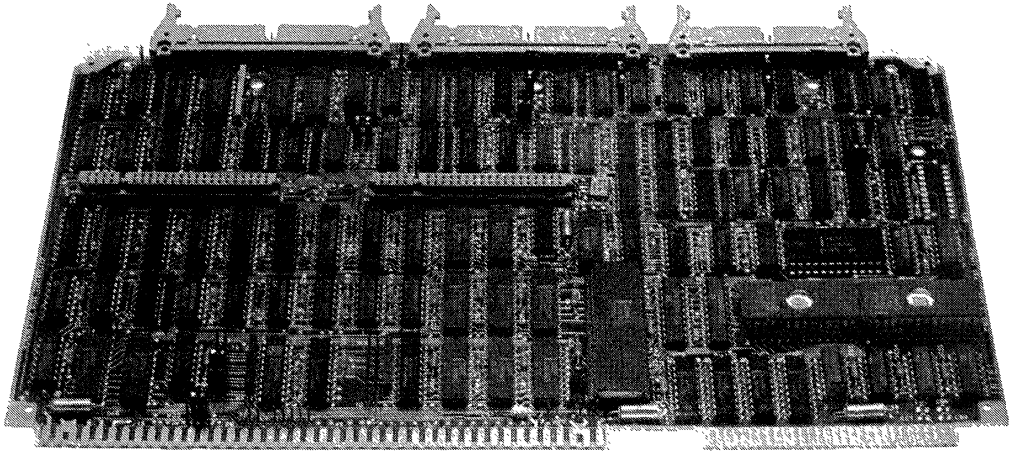
Order Code	Description
SBC208	Flexible Disk Controller

iSBC® 215*
GENERIC WINCHESTER CONTROLLER

- **Controls up to Four 5¼", 8" or 14" Winchester Disk Drives from Over Ten Different Vendors**
- **Compatible with Industry Standard MULTIBUS® (IEEE 796) Interface**
- **Supports ANSI X3T9/1226 Standard Interface**
- **Software Drivers Available for iRMX™ and Xenix* Operating Systems**
- **Intel 8089 I/O Processor Provides Intelligent DMA Capability**
- **On-Board Diagnostics and ECC**
- **Full Sector Buffering On-Board**
- **Capable of Directly Addressing 16 MB of System Memory**
- **Removable Back-up Storage Available Through the iSBX™ 218A Flexible Disk Controller and the iSBX 217C ¼" Tape Interface Module**

Using VLSI technology, the iSBC 215 Generic Winchester Controller combines three popular Winchester controllers onto one MULTIBUS board: the iSBC 215A open loop controller, the iSBC 215B closed loop controller, and an ANSI X3T9/1226 standard interface controller. The combined functionality of the iSBC 215 supports up to four 5¼", 8" or 14" Winchester drives from over 10 different vendors. Integrated back-up is available via two SBX bus modules; the iSBX 218A module for floppy disk drives and the iSBX 217C module for ¼" tape units.

From the MULTIBUS side, the iSBC 215 appears as one standard software interface, regardless of the drive type used. In short, the iSBC 215 allows its user to change drive types without rewriting software. The iSBC 215 is totally downward compatible with its predecessors, the iSBC 215A and 215B controller; allowing existing iSBC 215A and 215B users to move quickly to the more powerful iSBC 215. In addition, the iSBC 215 directly addresses up to 16 megabytes of system memory.



210618-1

Xenix is a trademark of Microsoft Corp.

*The iSBC® 215 is also manufactured under product code piSBC® 215 by Intel Puerto Rico, Inc.

September 1989
Order Number: 210618-003

BLOCK DIAGRAMS

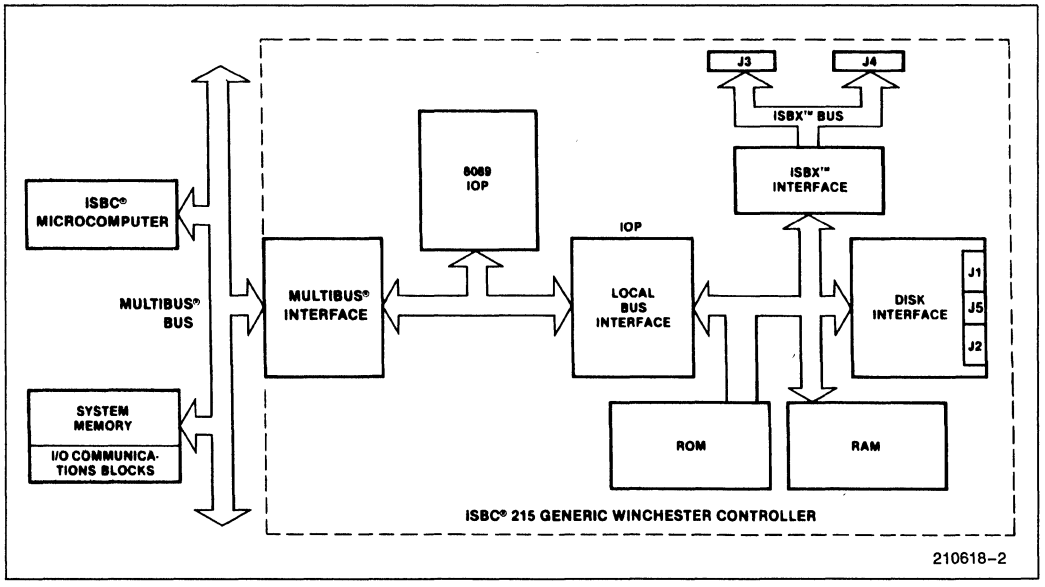
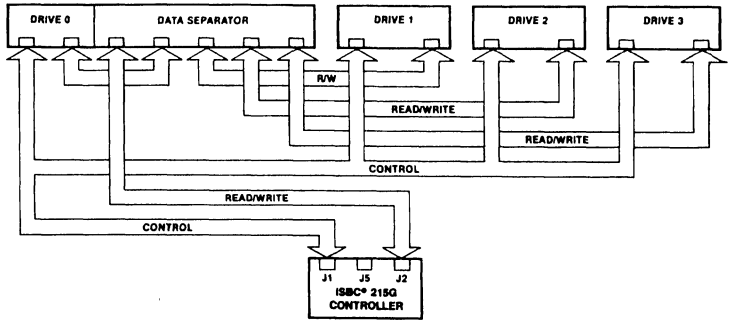


Figure 1. Block Diagram of iSBC® 215 Generic Winchester Disk Controller

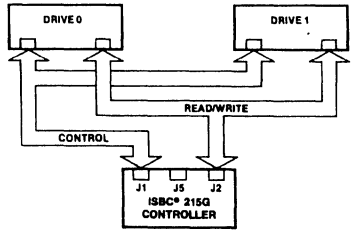
Interface with Shugart/Quantum/RMS Drives



210618-3

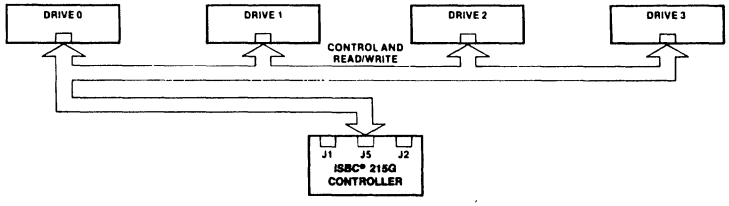
NOTE:
1. Shugart SA1000 or RMS Data Express.* *Data Express is a trademark of Rotating Memory Systems.

Interface with Memorex/Shugart Drives



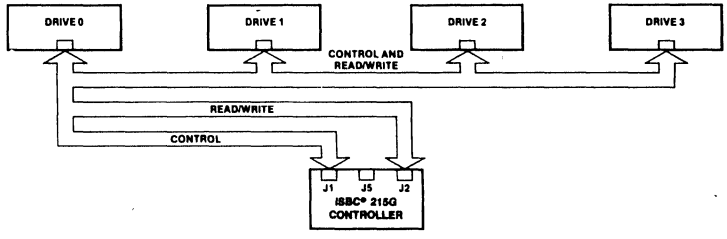
210618-4

Interface with ANSI Drive



210618-5

Interface with PRIAM Drives



210618-6

Figure 2. Controller to Drive Interfacing

SPECIFICATIONS

Compatibility

CPU—Any MULTIBUS computer or system mainframe.

Disk Drives—Winchester Disk Drives; both open-loop and closed-loop head positioner types. The following drives are known to be compatible:

Open-Loop
Shugart SA 1000 Series Shugart SA 4000 Series Memorex 100 Series Quantum Q2000 Series Fujitsu 2301, 2302 CDC 9410 RMS 5¼" Series Rodine 5¼" Series Ampex 5¼" Series CMI 5¼" Series
Closed-Loop
Priam 8" and 14" Drive Series
ANSI
3M 8430 Series Kennedy 6170 Series Micropolis 8" Series Pertec Trackstar Series Priam 8" Series Megavault (SLI) 8" Series
SBX Boards
iSBX™ 218A Flexible Disk Controller iSBX™ 217C ¼" Tape Interface

Equipment Supplied

iSBC 215 Generic Winchester Controller Reference Schematic

Controller-to-drive cabling and connectors are not supplied with the controller. Cables can be fabricated with flat cable and commercially-available connectors as described in the iSBC 215G Hardware Reference Manual.

Physical Characteristics

Width:	6.75 in. (17.15 cm)
Height:	0.5 in. (1.27 cm)
Length:	12.0 in. (30.48 cm)
Shipping Weight:	19 oz. (0.54 kg)
Mounting:	Occupies one slot of iSBC system chassis or cardcage/backplane

With an SBX board mounted, vertical height increases to 1.13 in. (2.87 cm).

Electrical Characteristics

Power Requirements

+5 VDC@4.52A max
−5 VDC@0.015A max[1]
+12 VDC@0.15A max[2]
−12 VDC@0.055A max[1,2]

NOTES:
1. On-board regulator and jumper allows −12 VDC usage from MULTIBUS.
2. Required for some SBX boards.

Data Organization

Bytes/Sector	Sectors/Track(1) 128	256	512	1024
Priam 8"	72	42	23	12
Priam 14"	107	63	35	18
RMS/Shugart 8" /Quantum/Ampes/Rodine/CM1	54	31	17	9
Fujitsu/Memorex	64	38	21	11
Shugart 14"	96	57	31	16
CDC Finch	64	41	23	12
3M (ANSI)	82	51	29	16
Megavault (ANSI)	73	43	21	12
Kennedy (ANSI)	74	43	23	12
Micropolis (ANSI)	71	44	25	13
Pertec (ANSI)	85	52	29	15

NOTE:
1. Maximum allowable for corresponding selection of bytes per sector.

Drives per Controller

5¼" Winchester Disk Drives—Up to four RMS, CMI, Rodine or Ampex drives.

8" Winchester Disk Drives—Up to four ANSI, Shugart, Quantum or Priam drives; up to two Memorex, CDC, or Fujitsu drives.

14" Winchester Disk Drives—Up to four Priam drivers; up to two Shugart drives.

Flexible Disk Drives—Up to four drives through the optional iSBX 218A Flexible Disk Controller connected to the iSBC 215 board's iSBX connector.

¼" Tape Drives—Up to four drives through the optional iSBX 217C ¼" Tape Interface Module connected to the iSBC 215 board's iSBX connector.

Environmental Characteristics

Temperature—0° to 55°C (operating); −55°C to +85°C (non-operating)

Humidity—Up to 90% relative humidity without condensation (operating); all conditions without condensation or frost (non-operating).

Reference Manual

144780—iSBC 215 Generic Winchester Controller Hardware Reference Manual (NOT SUPPLIED)

Reference manuals may be ordered from any Intel sales representative, distributor office, or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, CA 95051.

ORDERING INFORMATION

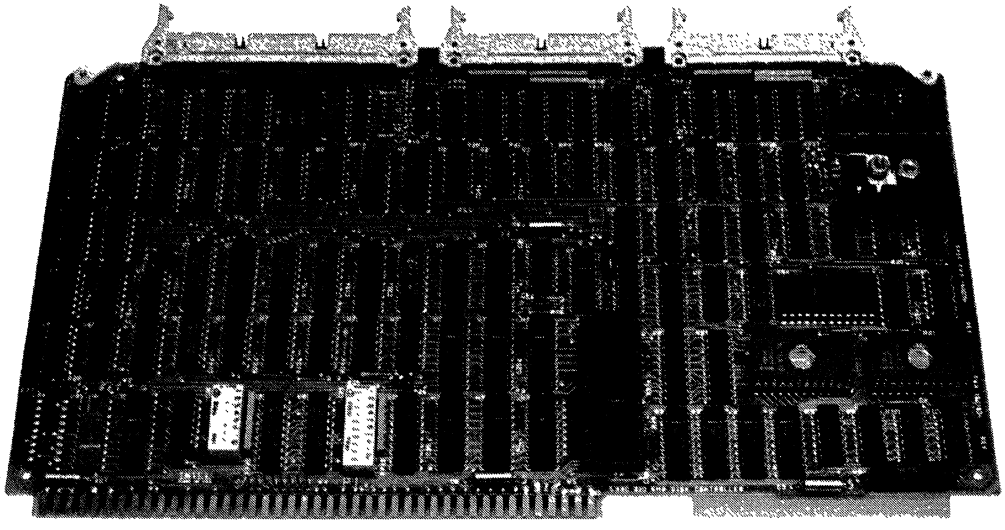
Order Code	Description
SBC215G	Generiç Winchester Controller

iSBC® 220*
SMD DISK CONTROLLER

- **Controls up to Four Soft Sectored SMD Interface Compatible Disk Drives**
- **12 MB to 2.4 GB per Controller**
- **Compatible with all iSBC® 80, iSBC® 88, and iSBC® 86 Single Board Computers**
- **Intel 8089 I/O Processor Provides Two High Speed DMA Channels as well as Controller Intelligence**
- **Software Drivers Available for iRMX™ and XENIX* Operating Systems**
- **On-Board Diagnostic and ECC**
- **Full Sector Buffering On-Board**
- **Capable of Addressing 1 MB of System Memory**
- **SMD Interface Available on Winchester, CMD, SMD and Large Fixed-Media Drives**

The iSBC 220 SMD Disk Controller brings very large mass storage capabilities to any iSBC 80, iSBC 88, or iSBC 86 MULTIBUS® system. The controller will interface to any soft sectored disk drive conforming to the industry standard SMD interface. Using simplified cable connections, up to four drives may be connected to the iSBC 220 Controller Board to give a total maximum capacity of 2.4 gigabytes. The Intel 8089 I/O Processor simplifies programming through the use of memory-based parameter blocks. A linked list technique allows the user to perform multiple disk operations.

*XENIX is a registered trademark of Microsoft.



143283-1

*The iSBC® 220 is also manufactured under product code piSBC® 220 or siSBC 220 by Intel Puerto Rico, Inc. or Intel Singapore, Ltd.

October 1989
Order Number: 143283-003

BLOCK DIAGRAMS

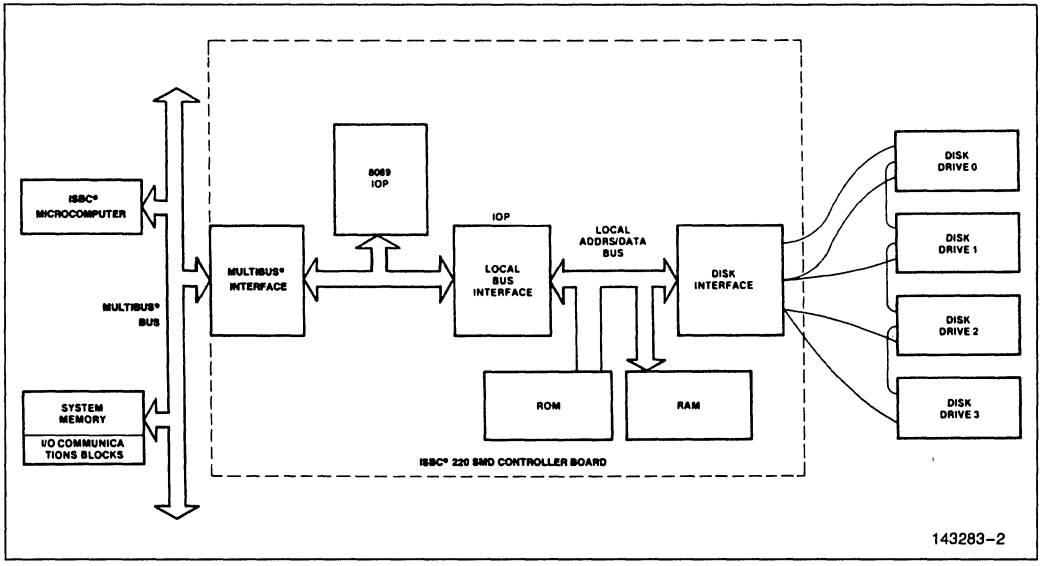


Figure 1. Simplified Block Diagram of iSBC® 220 SMD Disk Controller

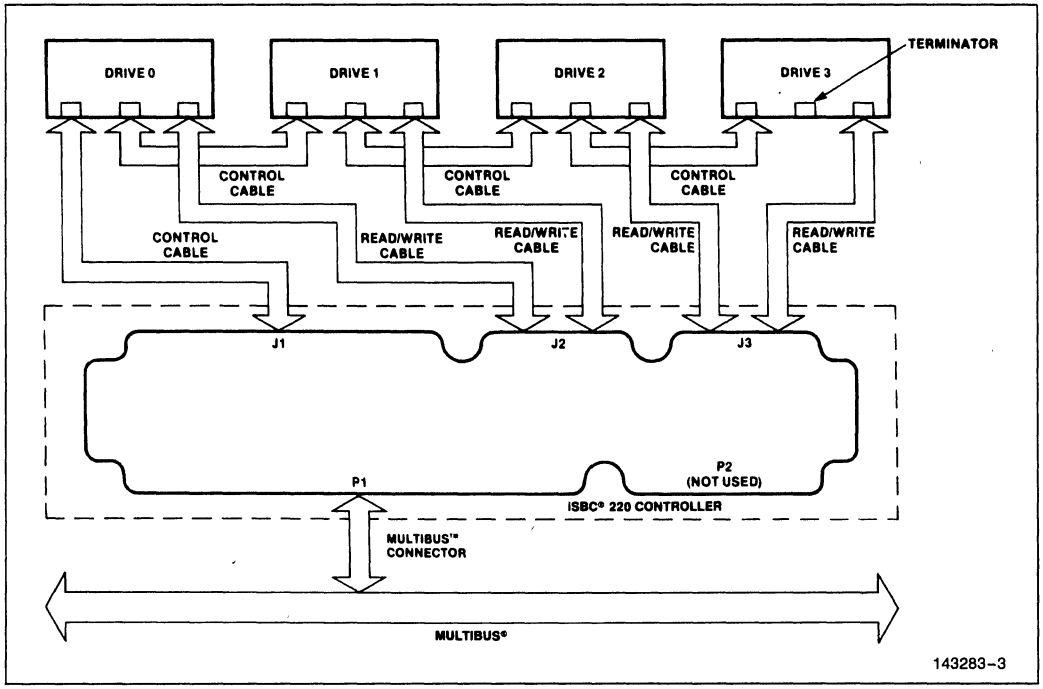


Figure 2. Typical Multiple Drive System

SPECIFICATIONS

Compatibility

CPU: Any MULTIBUS computer on system mainframe

Disk Drive: Any soft sectored SMD interface-compatible disk drive

Equipment Supplied

iSBC 220 SMD Disk Controller
Reference schematic

Controller-to-drive cabling and connectors are not supplied with the controller. Cables can be fabricated with flat cable and commercially-available connectors as described in the iSBC 220 SMD Disk Controller Hardware Reference Manual.

Physical Characteristics

Width: 6.75 in (17.15 cm)
Height: 0.5 in (1.27 cm)
Length: 12.0 in. (30.48 cm)
Shipping Weight: 19 oz. (0.54 kg)
Mounting: Occupies one slot of iSBC system chassis or cardcage/backplane

Electrical Characteristics

Power Requirements:
+5 VCD @ 3.25A max
−5 VDC @ 0.75A max(1)

NOTE:
1. On-board voltage regulator allows optional −12 VDC usage from MULTIBUS.

Data Organization and Capacity

Bytes per Sector(2): 128 256 521 1024
Sector per Track(2): 108 64 35 18

NOTE:
2. Software selectable.

Table 1. Drive Characteristics (Typical)

Disk (spindle) Speed	3600 rpm	
Tracks per Surface	823	
Head Positioning	Closed loop servo type, track following	
Access Time	Track to Track	6 ms
	Average	30 ms
	Maximum	55 ms
Data Transfer Rate	1.2 megabytes/second	
Storage Capacity	12 to 2.4 gigabytes	

Environmental Characteristics

Temperature: 0°C to 55°C (operating); −55°C to +85°C (non-operating)

Humidity: Up to 90% relative humidity without condensation (operating); all conditions without condensation or frost (non-operating)

Reference Manual

121597—iSBC 220 SMD Disk Controller Hardware Reference Manual (NOT SUPPLIED)

Reference manuals may be ordered from any Intel sales representative, distributor office, or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, CA 95051.

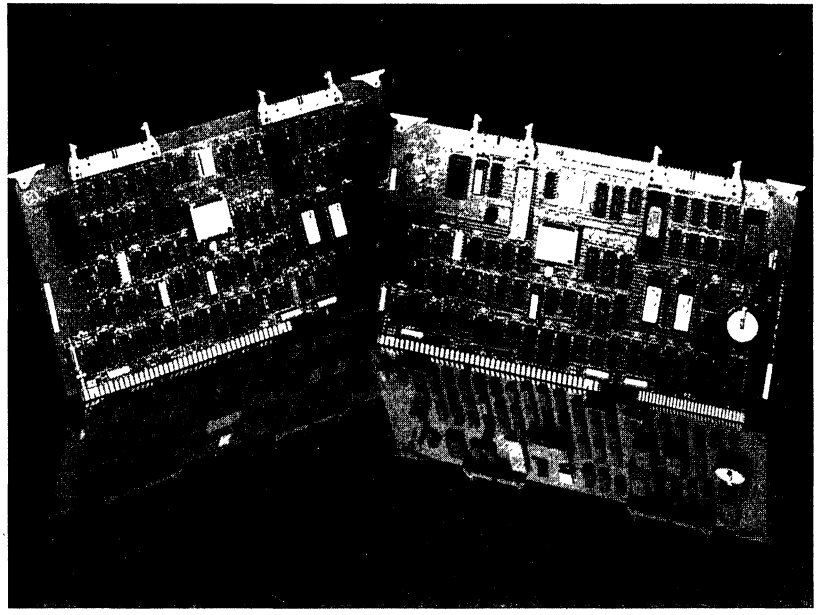
ORDERING INFORMATION

Order Code	Description
SBC220	SMD Disk Controller

MULTIBUS® I
Serial Communication Boards

11

iSBC® 548/549* TERMINAL CONTROLLERS



HIGH PERFORMANCE TERMINAL CONTROLLER BOARDS FOR MULTIBUS®I

The iSBC® 548 and iSBC® 549 are intelligent terminal controllers for MULTIBUS®I applications. The iSBC 548 provides basic multiuser support with 8 channels of RS 232 Ansychronous interface. The iSBC 549 combines 4 serial channels with a real-time clock and a line printer interface. Acting as intelligent slaves for communication expansion, these boards provide high performance, low cost solutions for multi-user systems.

FEATURES:

iSBC 548 FEATURES

- Supports eight channels asynchronous RS232 interface

iSBC 549 FEATURES

- Supports four channels asynchronous RS232 interface
- Line printer interface
- Real-time clock/calendar with battery backup

STANDARD iSBC 548/549 FEATURES

- 8 MHz 80186 Microprocessor
- Supports transfer rates up to 19.2K Baud
- 128 K Bytes Zero Wait State DRAM (32K Dual Port)
- Supports Full Duplex Asynchronous Transmissions
- Jumper selectable memory mapping, I/O mapping and MULTIBUS Interrupts

intel

* The iSBC® 548/549 is also manufactured under product code piSBC® 548/549 by Intel Puerto Rico, Inc

September, 1989
Order Number 280674-002

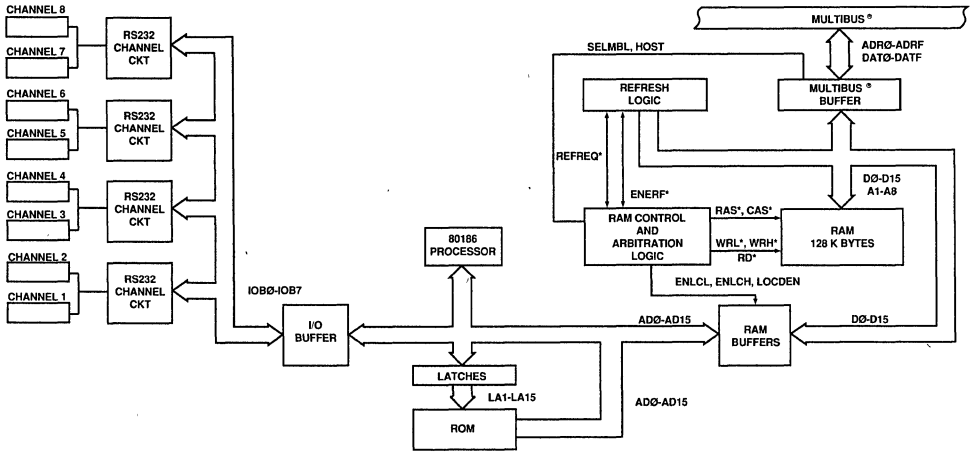


Figure 1: iSBC® 548 Functional Block Diagram

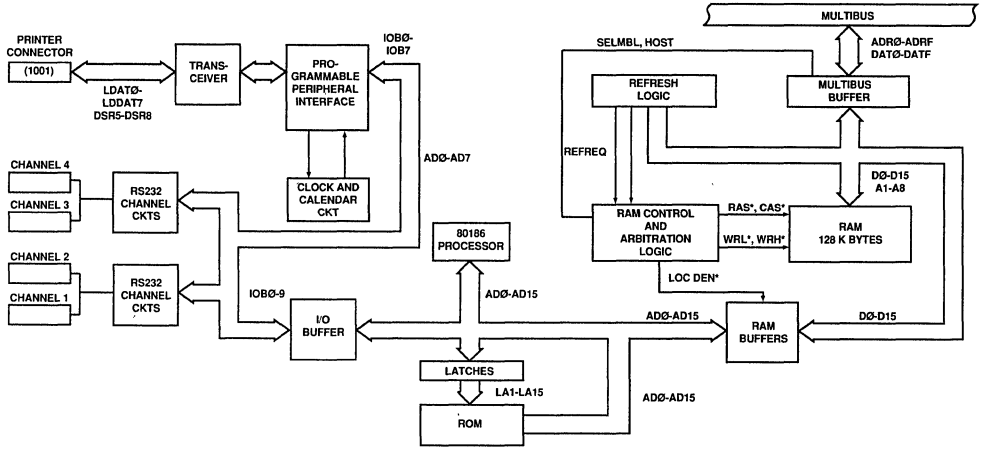


Figure 2: iSBC® 549 Functional Block Diagram

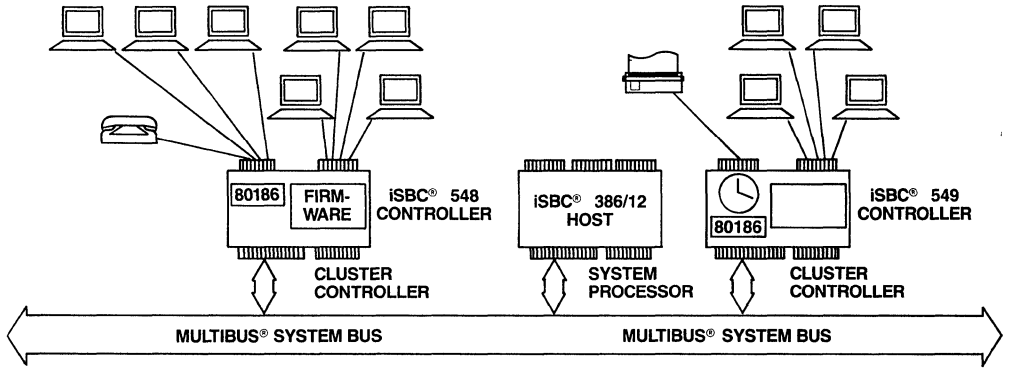


Figure 3: Terminal/Cluster Controller Application

FEATURES

ASYNCHRONOUS RS232 INTERFACE SUPPORT

The iSBC® 548/549 Asynchronous RS232 Internal support is presented in DTE Configuration. 82530 Serial Communications Controllers (SCCs) provide channels of half/full duplex serial I/O. Configurability of the 82530 allows handling all asynchronous data formats regardless of data size, number of start or stop bits, or parity requirements. The synchronous transmission features of the 82530 are not supported. An on-chip baud rate generator allows independent baud rates on each channel. The serial lines can be brought to the back-panel via 40-pin connectors and ribbon cable.

LINE PRINTER INTERFACE

The iSBC 549 incorporates a standard line printer interface compatible with IBM* or Centronics* line printers. Intelligent buffering on the iSBC 549 allows the CPU to offload printing tasks and return to higher priority jobs.

REAL-TIME CLOCK/CALENDAR

Multibus systems will benefit from the real-time clock present on the iSBC 549 in applications requiring time stamp operations, unattended boots and other calendar requirements. The clock/calendar circuit is backed up by a non-rechargeable battery which keeps the clock/calendar operating for six months with all other power off.

8 MHZ 80186 MICROPROCESSOR

The 80186 central processor component provides high-performance, flexibility, and powerful processing. The 80186/82530 combination with on-board PROM/EPROM sites, and dual-port RAM provides the intelligence and speed to manage multi-user communications.

TRANSFER RATES UP TO 19.2K BAUD

Collectively, each board has dual-port RAM providing an on-board buffer to handle incoming and outgoing messages at data rates up to 19.2K baud. The resident firmware supports asynchronous RS232 serial channels, provides modern control and performs power-up diagnostics. Each serial channel can be individually programmed to different baud rates to allow system configurations with differing terminal types.

MEMORY

The iSBC 548/549 have three areas of memory on-board: dual-port RAM, private RAM, and EPROM. Each board contains 128K bytes of on-board RAM, 32K bytes of dual-port RAM can be addressed by other MULTIBUS boards. The dual port memory is configurable in a 16M byte address space on 32K byte boundaries as addressed from the MULTIBUS port. The starting address is jumper selectable.

The second area of memory is 96K bytes of private RAM which is addressable by the 80186 on-board.

The third area of memory is EPROM memory expansion. Two 28-pin JEDEC sockets are provided. These sockets come populated with two EPROMs which contain the controller firmware. The boards can support 2764, 27128 and 27256 EPROMs, giving a total capacity of 64K bytes. The EPROM runs with zero wait states if EPROMs of access times 250 ns or less are used. No jumper changes are needed to access different size EPROMs.

WORLDWIDE SERVICE AND SUPPORT

Intel provides support for board repair or on-site service. Development options include phone support, subscription service, on-site consulting, and customer training.

QUALITY AND RELIABILITY

The iSBC 548 and iSBC 549 are designed and manufactured in accordance with Intel's high quality standards. We then verify quality through rigorous testing in our state-of-the-art Environmental Test Laboratory.

* IBM is a trademark of International Business Machines
* Centronics is a registered trademark of Centronics, Inc

SERIAL COMMUNICATIONS CHARACTERISTICS

Asynchronous only
- 6-8 bit character length
- 1, 1½, or 2 stop bits per character
- Parity
- Programmable clock
- Break Generation
- Framing error detection

Baud Rates

The on-board firmware can automatically detect and set baud rates of 150, 300, 600, 1200, 4800, 9600 and 19200. Other baud rates can be set by the host.

Serial RS232C Signals Supported
- CD Carrier Detect
- RXD Receive Data
- TXD Transmit Data
- DTR Data Terminal Ready
- SG Signal Ground
- DSR Data Set Ready
- RTS Ready to Send
- CTS Clear to Send
- RI Ringer Indicator

These signals are supported by the iSBC 548/549 Controller and on-board firmware. All signals may not be supported by the host operating system.

MEMORY

On-Board RAM – 128K bytes total
Private RAM – 96K bytes
Dual Port Ram – 32K bytes, can be addressed from MULTIBUS interface at any 32K boundary between 80000H and F8000H or between F80000H and FF8000H.

EPROM Options-Components	On-Board Capacity	Start Address
2764	16K	FC000H
27128	32K	F8000H
27256	64K	F000H

MULTIBUS SYSTEM BUS INTERFACE

The iSBC 548/549 boards meet MULTIBUS (IEEE 796) bus specifications D16 M24 116 V0 E.

DEVICE DRIVERS

Check the latest release of the iRMX I, II & III operating systems for details.

ENVIRONMENTAL CHARACTERISTICS

Temperature – 0 to 55°C at 200 Linear Feet/Minute (LFM) Air Velocity
Humidity – 5% to 90% non-condensing (25 to 70° C)

PHYSICAL CHARACTERISTICS

	iSBC 548	iSBC 549
Width	30.34cm (12.00 in)	30.34cm (12.00 in)
Length	16.87cm (6.75 in)	16.87cm (6.75 in)
Height	1.27 cm (.5 in)	1.27 cm (.5 in)
Weight	400 gm (14 oz)	358 gm (12.5 oz)

POWER REQUIREMENTS

Maximum Power Required per Voltage

Voltage (Volts)	Current (Amps)	Power (Watts)
iSBC 548		
+ 5	3.49	17.5
+12	.14	1.7
–12	.11	1.3
iSBC 549		
+ 5	3.26	16.3
+12	.07	.8
–12	.06	.7

ORDERING INFORMATION

Order Code	Description
SBC548	8 Channel High Performance Terminal Controller
SBC549	4 Channel High Performance Terminal Controller with Line Printer/Clock

REFERENCE MANUALS

iSBC 546/547/548/549 High Performance Terminal Controller Hardware Reference Manual – Order Number 122704

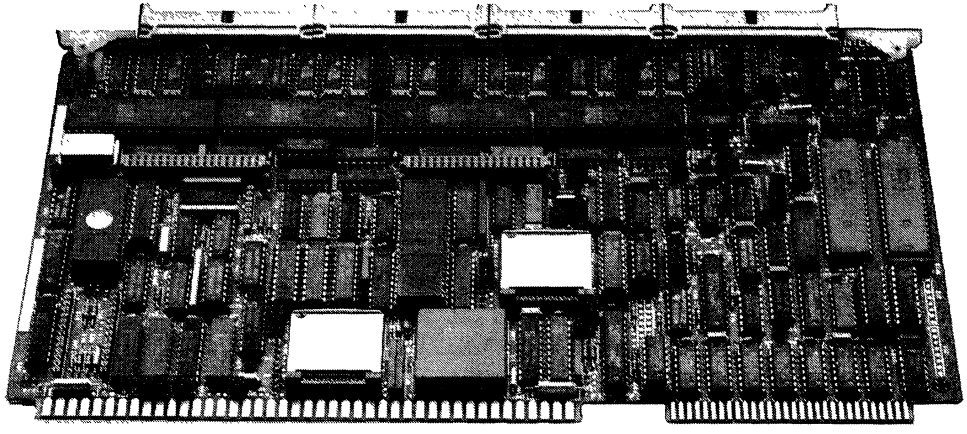
For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada).

iSBC® 188/56*
ADVANCED COMMUNICATING COMPUTER

- **iSBC® Single Board Computer or Intelligent Slave Communication Board**
- **8 Serial Communications Channels, Expandable to 12 Channels on a Single MULTIBUS® Board**
- **8 MHz 80188 Microprocessor**
- **Supports RS232C Interface on 6 Channels, RS422A/449 or RS232C Interface Configurable on 2 Channels**
- **Supports Async, Bisync HDLC/SDLC, On-Chip Baud Rate Generation, Half/Full-Duplex, NRZ, NRZI or FM Encoding/Decoding**
- **7 On-Board DMA Channels for Serial I/O, 2 80188 DMA Channels for the iSBX Bus Interface**
- **MULTIBUS Interface for System Expansion and Multimaster Configuration**
- **Two iSBX Connectors for Low Cost I/O Expansion**
- **256K Bytes Dual-Ported RAM On-Board**
- **Two 28-pin JEDEC PROM Sites Expandable to 6 Sites with the iSBC 341 MULTIMODULE Board for a Maximum of 192K Bytes EPROM**
- **Resident Firmware to Handle up to 12 RS232C Async Lines**

The iSBC 188/56 Advanced Communicating Computer is an intelligent 8-channel single board computer. This iSBC board adds the 8 MHz 80188 microprocessor-based communications flexibility to the Intel line of MULTIBUS OEM microcomputer systems. Acting as a stand-alone CPU or intelligent slave for communication expansion, this board provides a high performance, low-cost solution for multi-user systems. The features of the iSBC 188/56 board are uniquely suited to manage higher-layer protocol requirements needed in today's data communications applications. This single board computer takes full advantage of Intel's VLSI technology to provide state-of-the-art, economic, computer based solutions for OEM communications-oriented applications.

*The iSBC® 188/56 is also manufactured under product code piSBC 188/56 or siSBC 188/56 by Intel Puerto Rico, Inc. or Intel Singapore, Ltd.



280715-1

*IBM is a registered trademark of International Business Machines
*UNIX is a trademark of Bell Laboratories
*XENIX is a trademark of Microsoft Corporation

October 1989
Order Number: 280715-003

OPERATING ENVIRONMENT

The iSBC 188/56 board features have been designed to meet the needs of numerous communications applications. Typical applications include:

1. Terminal/cluster controller
2. Front-end processor
3. Stand-alone communicating computer

Terminal/Cluster Controller

A terminal/cluster controller concentrates communications in a central area of a system. Efficient handling of messages coming in or going out of the system requires sufficient buffer space to store messages and high speed I/O channels to transmit messages. More sophisticated applications, such as cluster controllers, also require character and format conversion capabilities to allow different types of terminals to be attached.

The iSBC 188/56 Advanced Communicating Computer is well suited for multi-terminal systems (see Figure 1). Up to 12 serial channels can be serviced in multi-user or cluster applications by adding two iSBX 354 MULTIMODULE boards. The dual-port RAM provides a large on-board buffer to handle incoming and outgoing messages at data rates up to 19.2K baud. Two channels are supported for continuous data rates greater than 19.2K baud. Each serial channel can be individually programmed for different baud rates to allow system configurations with differing terminal types. The firmware supplied on the iSBC 188/56 board supports up to 12 asynchronous RS232C serial channels, provides modem control and performs power-up diagnostics. The high performance of the on-board CPU provides intelligence to handle protocols and character handling typically assigned to the system CPU. The distribution of intelligence results in optimizing system performance by releasing the system CPU of routine tasks.

Front-End Processor

A front-end processor off-loads a system's central processor of tasks such as data manipulation and text editing of characters collected from the attached terminals. A variety of terminals require flexible terminal interfaces. Program code is often dynamically downloaded to the front-end processor from the system CPU. Downloading code requires sufficient memory space for protocol handling and program code. Flow control and efficient handling of interrupts require an efficient operating system to manage the hardware and software resources.

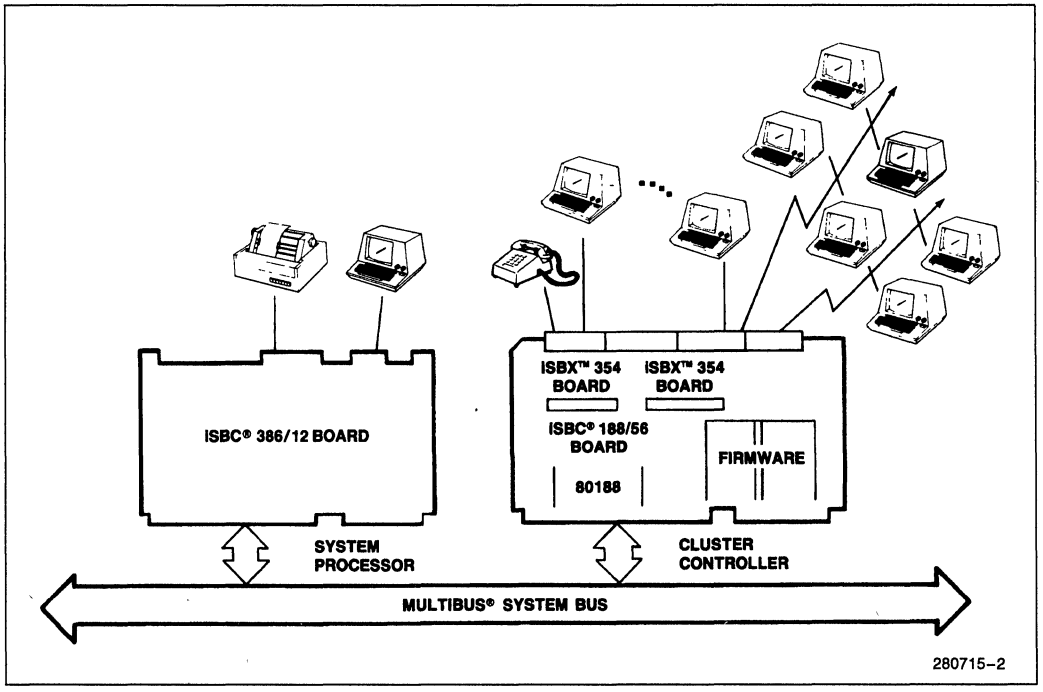


Figure 1. Terminal/Cluster Controller Application

The iSBC 188/56 board features are designed to provide a high performance solution for front-end processor applications (see Figure 2). A large amount of random access memory is provided for dynamic storage of program code. In addition, local memory sites are available for storing routine programs such as X.25, SNA or bisync protocol software. The serial channels can be configured for links to mainframe systems, point-to-point terminals, modems or multidrop configurations.

Stand-Alone Application

A stand-alone communication computer is a complete computer system. The CPU is capable of managing the resources required to meet the needs of multi-terminal, multi-protocol applications. These applications typically require multi-terminal support, floppy disk control, local memory allocation, and program execution and storage.

To support stand-alone applications, the iSBC 188/56 board uses the computational capabilities of an on-board CPU to provide a high-speed system solution controlling 8 to 12 channels of serial I/O (see Figure 3). The local memory available is large enough to handle special purpose code, execution code and routine protocol software. The MULTIBUS interface can be used to access additional system functions. Floppy disk control and graphics capability can be added to the iSBC standalone computer through the iSBX connectors.

ARCHITECTURE

The four major functional areas are Serial I/O, CPU, Memory and DMA. These areas are illustrated in Figure 4.

Serial I/O

Eight HDLC/SDLC serial interfaces are provided on the iSBC 188/56 board. The serial interface can be expanded to 12 channels by adding 2 iSBX 354 MULTIMODULE boards. The HDLC/SDLC interface is compatible with IBM* system and terminal equipment and with CCITT's X.25 packet switching interface.

Four 82530 Serial Communications Controllers (SCC) provide eight channels of half/full duplex serial I/O. Six channels support RS232C interfaces. Two channels are RS232C/422/449 configurable and can be tri-stated to allow multidrop networks. The 82530 component is designed to satisfy several serial communications requirements; asynchronous,

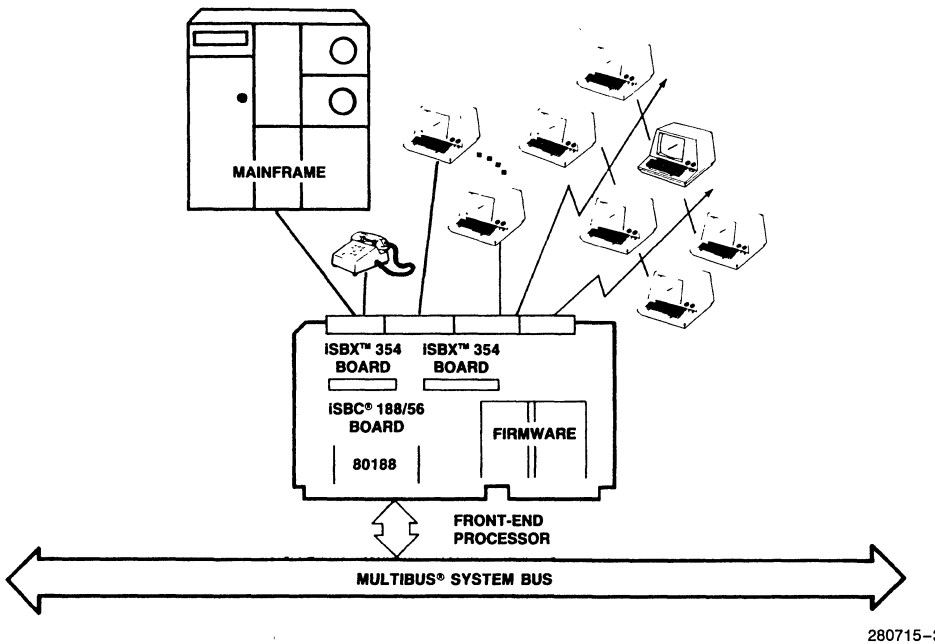


Figure 2. Front-End Processor Application

byte-oriented synchronous (HDLC/SDLC) modes. The increased capability at the serial controller point results in off-loading the CPU of tasks formerly assigned to the CPU or its associated hardware. Configurability of the 82530 allows the user to configure it to handle all asynchronous data formats regardless of data size, number of start or stop bits, or parity requirements. An on-chip baud rate generator allows independent baud rates on each channel.

The clock can be generated either internally with the SCC chip, with an external clock or via the NRZ1 clock encoding mechanism.

All eight channels can be configured as Data Terminal Equipment (DTE) or Data Communications Equipment (DCE). Table 1 lists the interfaces supported.

Table 1. iSBC® 188/56 Interface Support

Connection	Synchronous	Asynchronous
	Modem to Direct	Modem to Direct
Point-to-Point	X** Channels	X Channels
Multidrop	0 and 1	0 and 1
Loop	X	N/A

**All 8 channels are denoted by X.

Central CPU

The 80188 central processor component provides high performance, flexibility and powerful processing. The 80188 component is a highly integrated microprocessor with an 8-bit data bus interface and a 16-bit internal architecture to give high performance. The 80188 is upward compatible with 86 and 186 software.

The 80188/82530 combination with on-board PROM/EPROM sites, and dual-port RAM provide the intelligence and speed to manage multi-user, multi-protocol communication operations.

Memory

There are two areas of memory on-board: dual-port RAM and universal site memory. The iSBC 188/56 board contains 256K bytes of dual-port RAM that is addressable by the 80188 on-board. The dual-port memory is configurable anywhere in a 16M byte address space on 64K byte boundaries as addressed from the MULTIBUS port. Not all of the 256K bytes are visible from the MULTIBUS bus side. The amount of dual-port memory visible to the

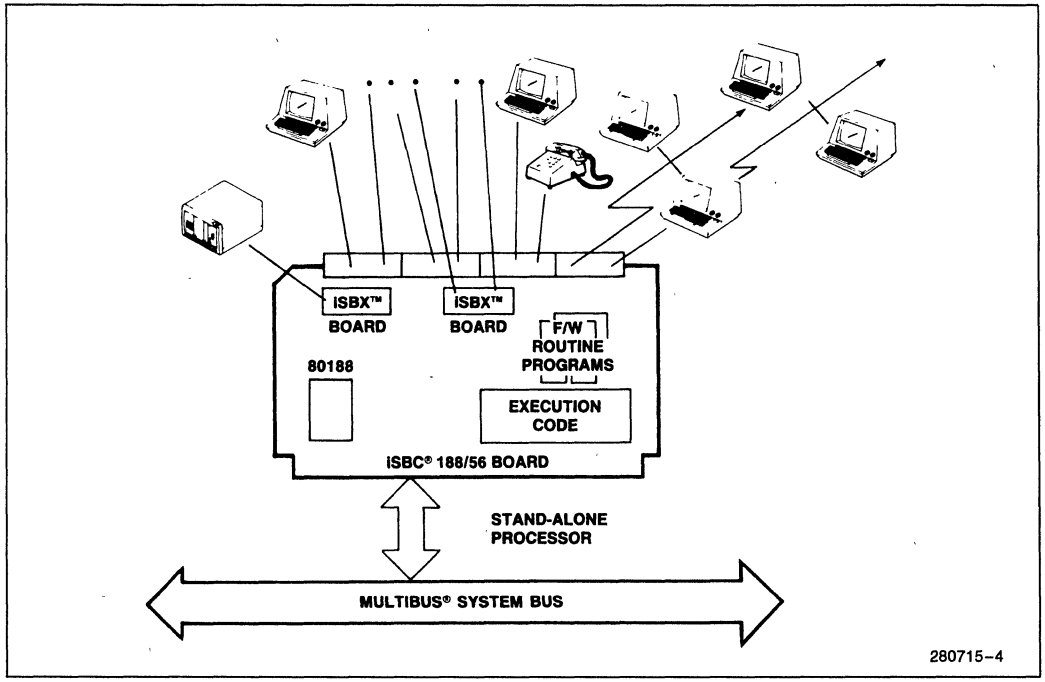


Figure 3. Stand-Alone Application

MULTIBUS side can be set (with jumpers) to none, 16K bytes, or 48K bytes. In a multiprocessor system these features provide local memory for each processor and shared system memory configurations where the total system memory size can exceed one megabyte without addressing conflicts.

The second area of memory is universal site memory providing flexible memory expansion. Two 28-pin JEDEC sockets are provided. One of these sockets is used for the resident firmware as described in the FIRMWARE section.

The default configuration of the boards supports 16K byte EPROM devices such as the Intel 27128 component. However, these sockets can contain ROM, EPROM, Static RAM, or EEPROM. Both sockets must contain the same type of component (i.e. as the first socket contains an EPROM for the resident firmware, the second must also contain an EPROM with the same pinout). Up to 32K bytes can be addressed per socket giving a maximum universal site memory size of 64K bytes. By using the iSBC 341 MULTIMODULE board, a maximum of 192K bytes of universal site memory is available. This provides sufficient memory space for on-board network or resource management software.

On-Board DMA

Seven channels of Direct Memory Access (DMA) are provided between serial I/O and on-board dual port RAM by two 8237-5 components. Each of channels 0, 1, 2, 3, 5, 6, and 7 is supported by their own DMA line. Serial channels 0 and 1 are configurable for full duplex DMA. Configuring the full duplex DMA option for Channels 0 and 1 would require Channels 2 and 3 to be interrupt driven or polled. Channel 4 is interrupt driven or polled only.

Two DMA channels are integrated in the 80188 processor. These additional channels can be connected to the iSBX interfaces to provide DMA capability to iSBX MULTIMODULE boards such as the iSBX 218A Floppy Disk Controller MULTIMODULE board.

OPERATING SYSTEM SUPPORT

Intel offers run-time foundation software to support applications that range from general purpose to high-performance solutions.

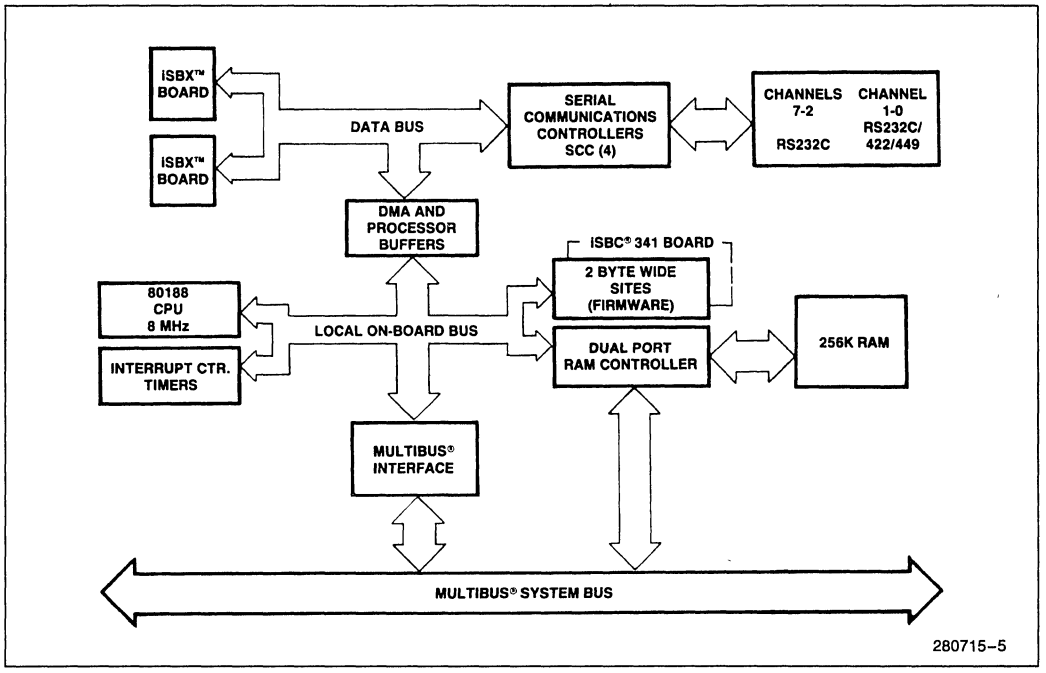


Figure 4. Block Diagram of iSBC® 188/56 Board

The iRMX Operating System provides a rich set of features and options to support sophisticated stand-alone communications applications on the iSBC 188/56 Advanced Communicating Computer. If the iSBC 188/56 board is acting as an intelligent slave in a system environment, an iRMX driver resident in the host CPU is available.

The System Debug Monitor (SDM) supports target system debugging for the iSBC 188/56 Advanced Communicating COMMputer board. The monitor contains the necessary hardware, software and documentation required to interface the iSBC 188/56 target system to an Intel microcomputer development system for debugging application software.

The XENIX* Operating System includes a software driver for the iSBC 188/56 board (and up to two iSBX 354 MULTIMODULE Boards) acting as an intelligent slave for multi-user applications requiring multiple persons running independent, terminal-oriented jobs.

FIRMWARE

The iSBC 188/56 Communicating COMMputer board is supplied with resident firmware that supports up to 12 RS232C asynchronous serial channels. In addition, the firmware provides a facility for a host CPU to download and execute code on the iSBC 188/56 board. Simple power-up confidence tests are also included to provide a quick diagnostic service. The firmware converts the iSBC 188/56 COMMputer board to a slave communications controller. As a slave communications controller, it requires a separate MULTIBUS host CPU board and requires the use of MULTIBUS interrupt line to signal the host processor. Table 2 summarizes the features of the firmware.

Table 2. Features of the iSBC® 188/56 Firmware

Feature	Description
Asynchronous Serial Channel Support	Supports the serial channels in asynchronous ASCII mode. Parameters such as baud rate, parity generation, parity checking and character length can be programmed independently for each channel.
Block Data Transfer (On Output)	Relieves the host CPU of character-at-a-time interrupt processing. The iSBC 188/56 board accepts blocks of data for transmission and interrupts the processor only when the entire block is transmitted.
Limited Modem Control	Provides software control of the Data Terminal Ready (DTR) line on all channels. Transitions on the Carrier Detect (CD) line are sensed and reported to the host CPU.
Tandem Modem Support	Transmits an XOFF character when the number of characters in its receive buffer exceeds a threshold value and transmits an XON character when the buffer drains below some other threshold.
Download and Execute Capability	Provides a capability for the host CPU to load code anywhere in the address space of the iSBC 188/56 board and to start executing at any address in its address space.
Power Up Confidence Tests	On board reset, the firmware executes a series of simple tests to establish that crucial components on the board are functional.

INTERRUPT CAPABILITY

The iSBC 188/56 board has two programmable interrupt controllers (PICs). One is integrated into the 80188 processor and the other in the 80130 component. The two controllers are configured with the 80130 controller as the master and the 80188 controller as the slave. Two of the 80130 interrupt inputs are connected to the 82530 serial controller components to provide vector interrupt capabilities by the serial controllers. The iSBC 188/56 board provides 22 interrupt levels. The highest level is the NMI (Non-Maskable Interrupt) line which is directly tied to the 80188 CPU. This interrupt is typically used for signaling catastrophic events (e.g. power failure).

There are 5 levels of interrupts internal to the 80188 processor. Another 8 levels of interrupts are available from the 80130 component. Of these 8, one is tied to the programmable interrupt controller (PIC) of the 80188 CPU. An additional 8 levels of interrupts are available at the MULTIBUS interface. The iSBC 188/56 board does not support bus vectored interrupts. Table 3 lists the possible interrupt sources.

SUPPORT FOR THE 80130 COMPONENT

Intel does not support the direct processor execution of the iRMX nucleus primitives from the 80130 component. The 80130 component provides timers and interrupt controllers.

Table 3. Interrupt Request Sources

Device	Function	Number of Interrupts
MULTIBUS Interface INT0–INT7	Requests from MULTIBUS resident peripherals or other CPU boards.	8
82530 Serial Controllers	Transmit buffer empty, receive buffer full and channel errors 1 and external status.	8 per 82530 Total = 32
Internal 80188 Timer and DMA	Timer 0, 1, 2 outputs and 2 DMA channel interrupts.	5
80130 Timer Outputs	Timer 0, 1, 2 outputs of 80130.	3
Interrupt from Flag Byte Logic	Flag byte interrupt set by MULTIBUS master (through MULTIBUS® I/O Write).	1
Bus Flag Interrupt	Interrupt to MULTIBUS® (Selectable for INT0 to INT7) generated from on-board 80188 I/O Write.	1
SBX Connectors SBX DMA	Function determined by iSBX board. DMA interrupt from iSBX (TDMA).	4 (Two per Connector) 2
Bus Fail-Safe Timeout Interrupt.	Indicates iSBC 188/48 board timed out either waiting for MULTIBUS access or timed out from no acknowledge while on MULTIBUS System Bus.	1
Latched Interrupt	Converts pulsed event to a level interrupt. Example: 8237A-5 EOP.	1
OR-Gate Matrix	Concentrates up to 4 interrupts to 1 interrupt (selectable by stake pins).	1
Ring Indicator Interrupt	Latches a ring indicator event from serial channels 4, 5, 6, or 7.	1
NOR-Gate Matrix	Inverts up to 2 interrupts into 1 (selectable by stake pins).	1

EXPANSION

EPROM Expansion

Memory may be expanded by adding Intel compatible memory expansion boards. The universal site memory can be expanded to six sockets by adding the iSBC 341 MULTIMODULE board for a maximum total of 192K bytes of universal site memory.

iSBX™ I/O Expansion Bus (IEEE 959)

Two 8-bit iSBX bus connectors are provided on the iSBC 188/56 board. Using iSBX modules additional functions can be added to extend the I/O capability of the board. In addition to specialized or custom designed iSBX boards, there is a broad range of iSBX boards from Intel including serial and parallel I/O, analog I/O, and IEEE 488 GPIB, boards.

The serial I/O SBX boards available include the iSBX 354 Dual Channel Expansion MULTIMODULE board. Each iSBX 354 MULTIMODULE board adds two channels of serial I/O to the iSBC 188/56 board for a maximum of twelve serial channels. The 82530 serial communications controller on the MULTIMODULE board handles a large variety of serial communications protocols. This is the same serial controller as is used on the iSBC 188/56 board to offer directly compatible expansion capability for the iSBC 188/56 board.

MULTIBUS® INTERFACE

The iSBC 188/56 board can be a MULTIBUS master or intelligent slave in a multimaster system. The iSBC 188/56 board incorporates a flag byte signalling mechanism for use in multiprocessor environments where the iSBC 188/56 board is acting as an intelligent slave. The mechanism provides an interrupt handshake from the MULTIBUS System Bus to the on-board-processor and vice-versa.

The Multimaster capabilities of the iSBC 188/56 board offers easy expansion of processing capacity and the benefits of multiprocessing. Memory and I/O capacity may be expanded and additional functions added using Intel MULTIBUS compatible expansion boards.

SPECIFICATIONS

Word Size

Instruction—8, 16, 24 or 32 bits
Data Path—8 bits

Processor Clock	82530 Clock	DMA Clock
8 MHz	4.9152 MHz	4 MHz

Dual Port RAM

iSBC 188/56 Board—256 bytes

As viewed from the 80188—64K bytes

As viewed from the MULTIBUS System Bus—Choice: 0, 16K or 48K

EPROM

iSBC® 188/56 Board Using:	Size	On Board Capacity	Address Range
2732	4K	8K bytes	$FE000-FFFFF_H$
2764	8K	16K bytes	$FC000-FFFFF_H$
27128	16K	32K bytes	$F8000-FFFFF_H$
27256	32K	64K bytes	$F0000-FFFFF_H$
27512	64K	128K bytes	$E0000-FFFFF_H$

Memory Expansion

EPROM with iSBC® 341 Board Using:	Capacity	Address Range
2732	24K bytes	$F8000-FFFFF_H$
2764	48K bytes	$F0000-FFFFF_H$
27128	96K bytes	$E0000-FFFFF_H$
27256	192K bytes	$C0000-FFFFF_H$

I/O Capacity

Serial—8 programmable lines using four 82530 components

SBX Bus—2 SBX single-wide boards

Serial Communications Characteristics

Synchronous—Internal or external character synchronization on one or two synchronous characters

Asynchronous—5–8 bits and 1, 1½, or 2 stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection.

Baud Rates

Synchronous X1 Clock	
Baud Rate	**82530 Count Value (Decimal)**
64000	36
48000	49
19200	126
9600	254
4800	510
2400	1022
1800	1363
1200	2046
300	8190
Asynchronous X16 Clock	
Baud Rate	**82530 Count Value (Decimal)**
19200	6
9600	14
4800	30
2400	62
1800	83
1200	126
300	510
110	1394

Interfaces

iSBX™ BUS

The iSBC 188/56 board meets IEEE 959 compliance level D8/8 DMA

MULTIBUS® SYSTEM BUS

The iSBC 188/56 board meets IEEE 796 compliance level Master/Slave D8 M24 I16 VO EL.

SERIAL RS232C SIGNALS

CD	Carrier
CTS	Clear to Send
DSR	Data Set Ready
DTE TXC	Transmit Clock
DTR	Data Terminal Ready
RTS	Request to Send
RXC	Receive Clock
RXD	Receive Data
SG	Signal Ground
TXD	Transmit Data
RI	Ring Indicator

RS422A/449 SIGNALS

RC	Receive Common
RD	Receive Data
RT	Receive Timing
SD	Send Data
TT	Terminal Timing

Environmental Characteristics

Temperature: 0 to 55°C at 200 Linear Feet/Min. (LFM) Air Velocity

Humidity: to 90%, non-condensing (25°C to 70°C)

Physical Characteristics

Width: 30.48 cm (12.00 in)
Length: 17.15 cm (6.75 in)
Height: 1.04 cm (0.41 in)
Weight: 595 gm (21 oz)

Electrical Characteristics

The power required per voltage for the iSBC 188/56 board is shown below. These numbers do not include the current required by universal memory sites or expansion modules.

Voltage (Volts)	Current (Amps) typ.	Power (Watts) typ.
+5	4.56A	22.8W
+12	0.12A	1.5W
−12	0.11A	1.3W

Reference Manual

iSBC 188/56 Advanced Data Communications Computer Reference Manual Order Number 148209.

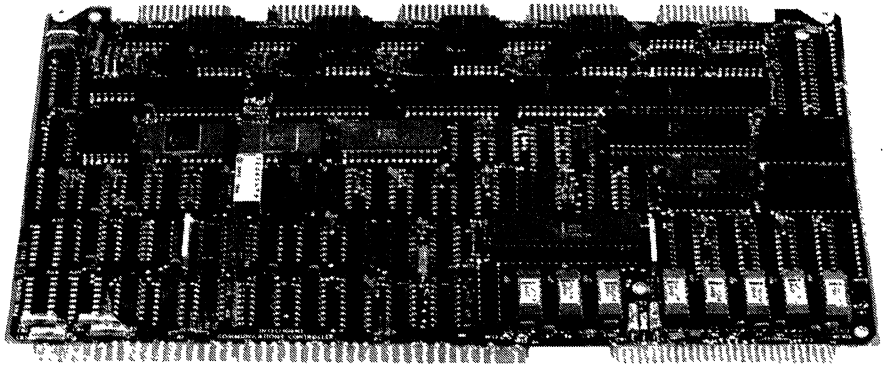
ORDERING INFORMATION

Order Code	Description
SBC188/56	8-Serial Channel Advanced Communicating Computer

iSBC® 544A*
INTELLIGENT COMMUNICATIONS CONTROLLER

- **iSBC® Communications Controller Acting as a Single Board Communications Computer or an Intelligent Slave for Communications Expansion**
- **On-Board Dedicated 8085A Microprocessor Providing Communications Control and Buffer Management for Four Programmable Synchronous/Asynchronous Channels**
- **Sockets for Up To 8K Bytes of EPROM**
- **16K Bytes of Dual Port Dynamic Read/Write Memory with On-Board Refresh**
- **Extended MULTIBUS® Addressing Permits Board Partitioning into 16K-Byte Segments in a 1-Megabyte Address Space**
- **Ten Programmable Parallel I/O Lines Compatible with Bell 801 Automatic Calling Unit**
- **Twelve Levels of Programmable Interrupt Control**
- **Individual Software Programmable Baud Rate Generation for Each Serial I/O Channel**
- **Three Independent Programmable Interval Timer/Counters**
- **Interface Control for Auto Answer and Auto Originate Modem**

The iSBC 544A Intelligent Communications Controller is a member of Intel's family of MULTIBUS® single-board computers, memory, I/O, and peripheral controller boards. The iSBC 544A board is a complete communications controller on a single 6.75 x 12.00 inch printed circuit card. The on-board 8085A CPU may perform local communications processing by directly interfacing with on-board read/write memory, nonvolatile read only memory, four synchronous/asynchronous serial I/O ports, RS232/RS366 compatible parallel I/O, programmable timers, and programmable interrupts.

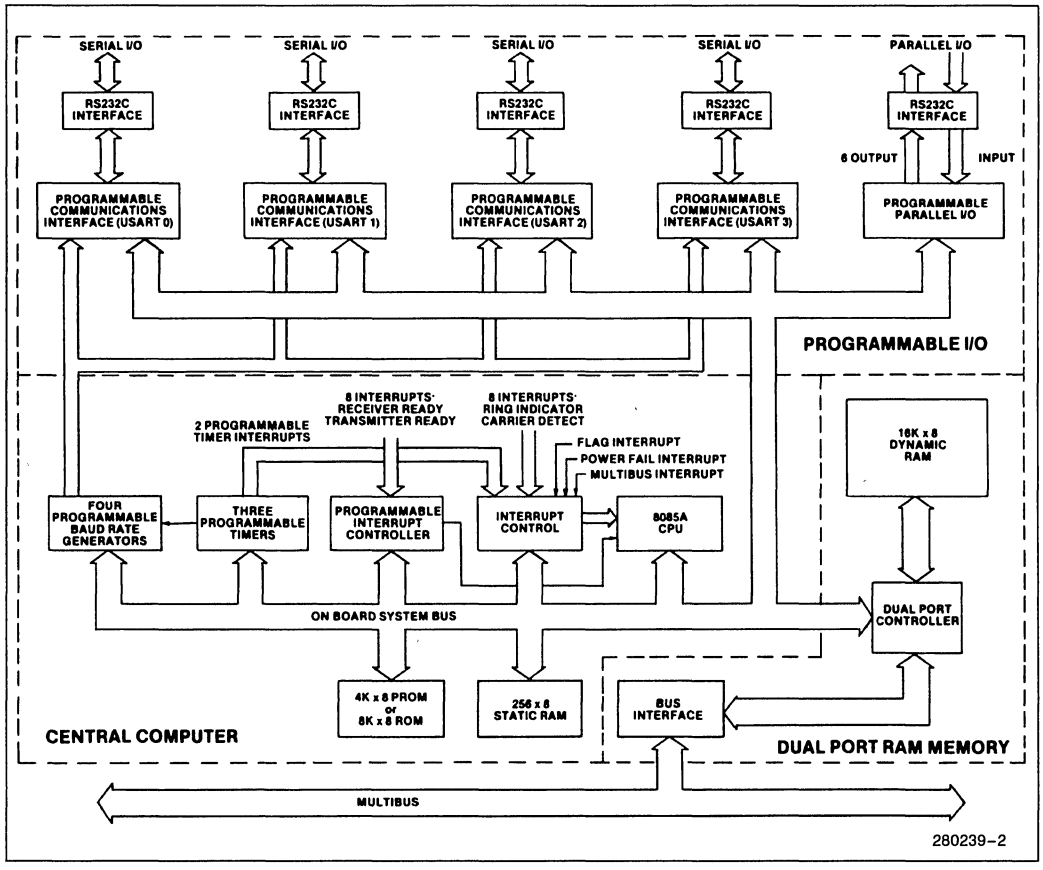


280239-1

*The iSBC® 544A is also manufactured under product code piSBC® 544A or siBC® 544A by Intel Puerto Rico, Inc., or Intel Singapore, Ltd.

September 1989
Order Number: 280239-002

BLOCK DIAGRAM



iSBC® 544A Intelligent Communications Controller Block Diagram

SPECIFICATIONS

Serial Communications Characteristics

Synchronous — 5-8 bit characters; automatic sync insertion; parity.

Asynchronous — 5-8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection; break character detection.

Baud Rates

Frequency (KHz)(1) (Software Selectable)	Baud Rate (Hz)(2)		
	Synchronous	Asynchronous	
		÷16	÷64
153.6	—	9600	2400
76.8	—	4800	1200
38.4	38400	2400	600
19.2	19200	1200	300
9.6	9600	600	150
4.8	4800	300	75
6.98	6980	—	110

NOTES:
1. Frequency selected by I/O writes of appropriate 16-bit frequency factor to Baud Rate Register.
2. Baud rates shown here are only a sample subset of possible software programmable rates available. Any frequency from 18.75 Hz to 614.4 KHz may be generated utilizing on-board crystal oscillator and 16-bit Programmable Interval Timer (used here as a frequency divider).

8085A CPU

Word Size — 8, 16 or 24 bits/instruction; 8 bits of data

Cycle Time — 1.45/μs ±0.01% for fastest executable instruction; i.e., four clock cycles.

Clock Rate — 2.76 MHz ± 0.1%

System Access Time

Dual port memory — 740 ns

NOTE:
Assumes no refresh contention.

Memory Capacity

On-Board ROM/PROM — 4K, or 8K bytes of user installed ROM or EPROM

On-Board Static RAM — 256 bytes on 8155

On-Board Dynamic RAM (on-board access) — 16K bytes. Integrity maintained during power failure with user-furnished batteries (optional)

On-Board Dynamic RAM (MULTIBUS access) — 4K, 8K, or 16K bytes available to bus by swtich selection

Memory Addressing

On-Board ROM/PROM — 0-0FFF (using 2716 EPROMs or masked ROMs); 0-1FFF (using 2732A EPROMs)

On-Board Static RAM — 256 bytes: 7F00-7FFF

On-Board Dynamic RAM (on-board access) — 16K bytes: 8000-BFFF.

On-Board Dynamic RAM (MULTIBUS® access) — any 4K increment 00000-FF000 which is switch and jumper selectable. 4K, 8K or 16K bytes can be made available to the bus by switch selection.

I/O Capacity

Serial — 4 programmable channels using four 8251A USARTs

Parallel — 10 programmable lines available for Bell 801 ACU, or equivalent use. Two auxiliary jumper selectable signals

I/O Addressing

On-Board Programmable I/O

Port	Data	Control
USART 0	D0	D1
USART 1	D2	D3
USART 2	D4	D5
USART 3	D6	D7
8155 PPI	E9 (Port A)	E8
	EA (Port B)	
	EB (Port C)	

Interrupts

Address for 8259A Registers (Hex notation, I/O address space)

E6 Interrupt request register
E6 In-service register
E7 Mask register
E6 Command register
E7 Block address register
E6 Status (polling register)

NOTE:
Several registers have the same physical address: Sequence of access and one data bit of the control word determines which register will respond.

Interrupt levels routed to the 8085 CPU automatically vector the processor to unique memory locations:

24 TRAP
3C RST 7.5
34 RST 6.5
2C RST 5.5

Timers

Addresses for 8253 Registers (Hex notation, I/O address space)

Programmable Interrupt Timer One

D8	Timer 0	BDG0
D9	Timer 1	BDG1
DA	Timer 2	BDG2
DB	Control register	

Programmable Interrupt Timer Two

DC	Timer 0	BDG3
DD	Timer 1	BDG4
DE	Timer 2	TINT1
DF	Control register	

Address for 8155 Programmable Timer

E8	Control	
	Timer (LSB)	TINT0
ED	Timer (MSB)	TINT0

Input Frequencies — Jumper selectable reference 1.2288 MHz ± 0.1% (0.814 μs period nominal) or 1.843 MHz ±0.1% crystal (0.542 μs period, nominal)

Output Frequencies (at 1.2288 MHz)

Function	Single Timer/Counter		Dual Timer/Counter (two timers cascaded)	
	Min	Max	Min	Max
Real-Time Interrupt Interval	1.63 μs	53.3 μs	3.26 μs	58.25 min
Rate Generator (frequency)	18.75 Hz	614.4 KHz	0.00029 Hz	307.2 KHz

Interfaces

Serial I/O — EIA Standard RS232C signals provided and supported:

Carrier Detect	Receiver Data
Clear to Send	Ring Indicator
Data Set Ready	Secondary Receive Data*
Data Terminal Ready	Secondary Transmit Data *
Request to Send	Transmit Clock
Receive Clock	Transmit Data
	DTE Transmit clock

* Optional if parallel I/O port is not used as Automatic Calling Unit.

Parallel I/O — Four inputs and eight outputs (includes two jumper selectable auxiliary outputs). All signals compatible with EIA Standard RS232C. Directly compatible with Bell Model 801 Automatic Calling Unit, or equivalent.

On-Board Addressing

All communications to the parallel and serial I/O ports, to the timers, and to the interrupt controller, are via read and write commands from the on-board 8085A CPU.

Auxiliary Power

An auxiliary power bus is provided to allow separate power to RAM for systems requiring battery backup of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

Memory Protect

An active-low TTL compatible memory protect signal is brought out on the auxiliary connector which, when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during the system power-down sequences.

Bus Drivers

Function	Characteristic	Sink Current (mA)
Data	Tri-state	50
Address	Tri-state	15
Commands	Tri-state	32

NOTE:
Used as a master in the single board communications computer mode.

Physical Characteristics

Width:	30.48 cm (12.00 inches)
Depth:	17.15 cm (6.75 inches)
Thickness:	1.27 cm (0.50 inch)
Weight:	3.97 gm (14 ounces)

Electrical Characteristics

DC Power Requirements

Configuration	Current Requirements			
	$V_{CC} = +5V \pm 5\%$ (max)	$V_{DD} = \pm 12V \pm 5\%$ (max)	$V_{BB} = -5V^{(3)} \pm 5\%$ (max)	$V_{AA} = -12V \pm 5\%$ (max)
With 4K EPROM (using 2716)	I_{CC} = 3.4A max	I_{DD} = 350 mA max	I_{BB} = 5 mA max	I_{AA} = 200 mA max
Without EPROM	3.3A max	350 mA max	5 mA max	200 mA max
RAM only(1)	390 mA max	176 mA max	5 mA max	—
RAM(2) refresh only	390 mA max	20 mA max	5 mA max	

NOTES:
1. For operational RAM only, for AUX power supply rating.
2. For RAM refresh only. Used for battery backup requirements. No RAM accessed.
3. V_{BB} is normally derived on-board from V_{AA}, eliminating the need for a V_{BB} supply. If it is desired to supply V_{BB} from the bus, the current requirement is as shown.

Environmental Characteristics

Operating Temperature: 0°C to 55°C (32°F to 131°F)

Relative Humidity: To 90% without condensation

Reference Manual

502160 — iSBC 544 Intelligent Communications Controller Board Hardware Reference Manual (NOT SUPPLIED)

Reference manuals are shipped with each product only if designated SUPPLIED (see above). Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

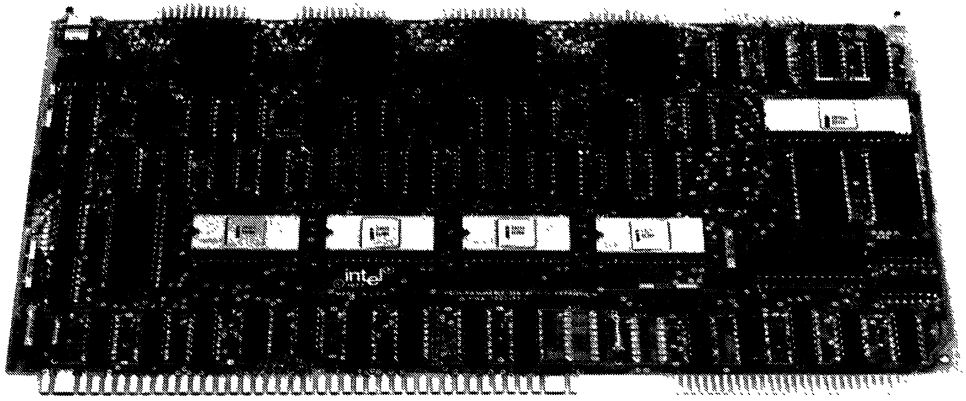
ORDERING INFORMATION

Order Code	Description
SBC 544A	Intelligent Communications Controller

iSBC® 534*
FOUR CHANNEL COMMUNICATION EXPANSION BOARD

- **Serial I/O Expansion Through Four Programmable Synchronous and Asynchronous Communications Channels**
- **Individual Software Programmable Baud Rate Generation for Each Serial I/O Channel**
- **Two Independent Progammable 16-Bit Interval Timers**
- **Sixteen Maskable Interrupt Request Lines with Priority Encoded and Programmable Interrupt Algorithms**
- **Jumper Selectable Interface Register Addresses**
- **16-Bit Parallel I/O Interface Compatible with Bell 801 Automatic Calling Unit**
- **RS232C/CCITT V.24 Interfaces Plus 20 mA Optically Isolated Current Loop Interfaces (Sockets)**
- **Programmable Digital Loopback for Diagnostics**
- **Interface Control for Auto Answer and Auto Originate Modems**

The iSBC 534 Four Channel Communication Expansion Board is a member of Intel's complete line of memory and I/O expansion boards. The iSBC 534 interfaces directly to any single board computer via the MULTIBUS to provide expansion of system serial communications capability. Four fully programmable synchronous and asynchronous serial channels with RS232C buffering and provision for 20 mA optically isolated current loop buffering are provided. Baud rates, data formats, and interrupt priorities for each channel are individually software selectable. In addition to the extensive complement of EIA Standard RS232C signals provided, the iSBC 534 provides 16 lines of RS232C buffered programmable parallel I/O. This interface is configured to be directly compatible with the Bell Model 801 automatic calling unit. These capabilities provide a flexible and easy means for interfacing Intel iSBC based systems to RS232C and optically isolated current loop compatible terminals, cassettes, asynchronous and synchronous modems, and distributed processing networks.

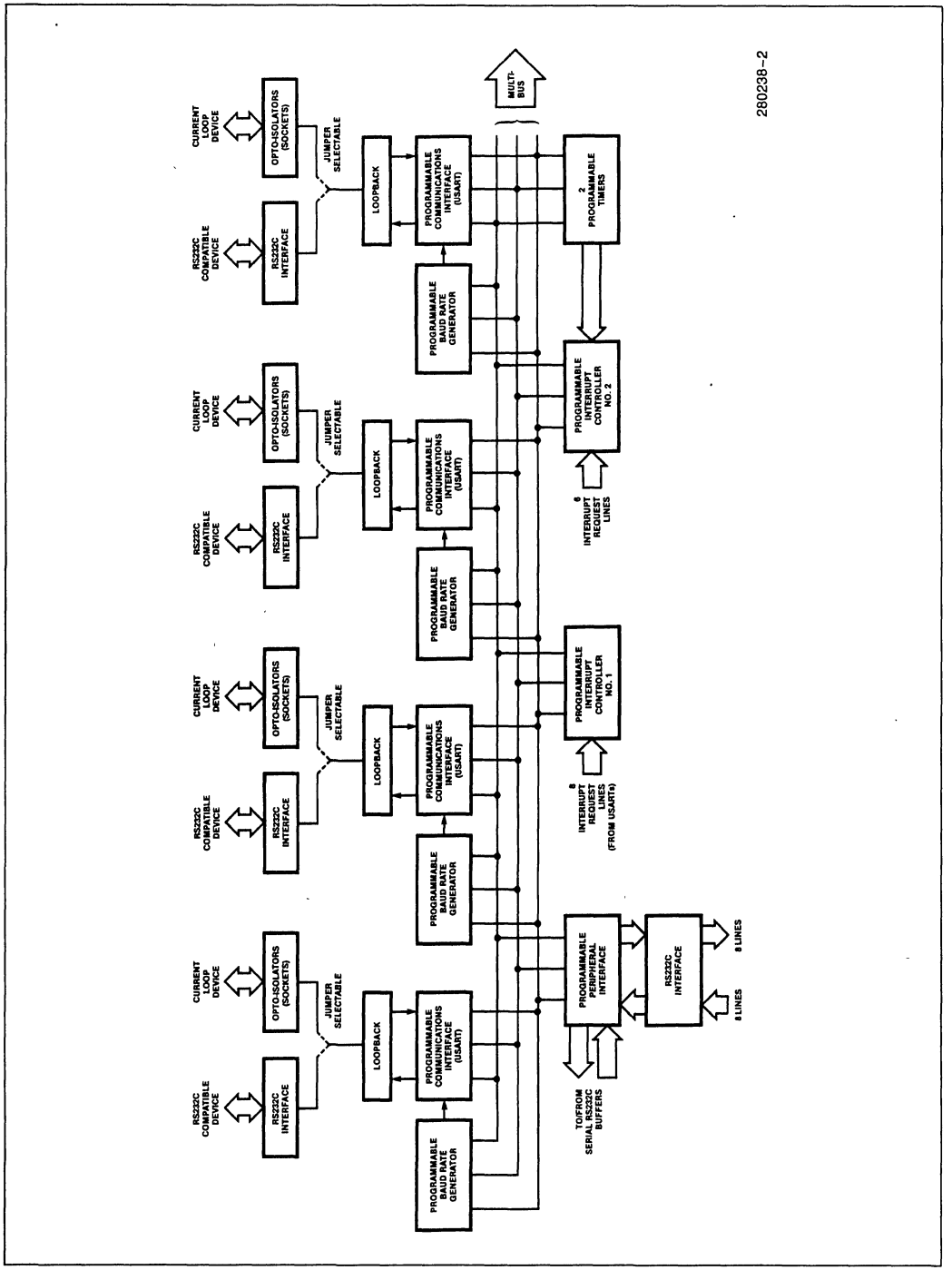


280238-1

*The iSBC® 534 is also manufactured under product code piSBC® 534 or siSBC® 534 by Intel Puerto Rico, Inc. or Intel Singapore, Ltd.

September 1989
Order Number: 280238-002

BLOCK DIAGRAM



iSBC® 534 Four Channel Communications Expansion Board Block Diagram

SPECIFICATIONS

Serial Communications Characteristics

Synchronous— 5-8 bit characters; internal or external character synchronization; automatic sync insertion.

Asynchronous— 5-8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection.

Sample Baud Rates(1)

Frequency(2) (kHz, Software Selectable)	Baud Rate (Hz)		
	Synchronous	Asynchronous	
		÷ 16	÷ 64
153.6	—	9600	2400
76.8	—	4800	1200
38.4	38400	2400	600
19.2	19200	1200	300
9.6	9600	600	150
4.8	4800	300	75
6.98	6980	—	110

NOTES:
1. Baud rates shown here are only a sample subset of possible software programmable rates available. Any frequency from 18.75 Hz to 614.4 kHz may be generated utilizing on-board crystal oscillator and 16-bit programmable interval timer (used here as frequency divider).
2. Frequency selected by I/O writes of appropriate 16-bit frequency factor to Baud Rate Register.

Interval Timer and Baud Rate Generator Frequencies

Input Frequency (On-Board Crystal Oscillator)— 1.2288 MHz ± 0.1% (0.813 μs period, nominal)

Function	Single Timer		Dual/Timer Counter (Two Timers Cascaded)	
	Min	Max	Min	Max
Real-Time Interrupt Interval	1.63 μs	53.3 ms	3.26 μs	58.25 minutes
Rate Generator (Frequency)	18.75 Hz	614.4 kHz	0.0029 Hz	307.2 kHz

Interfaces—RS232C Interfaces

EIA Standard RS232C Signals provided and supported:

Carrier detect	Receive data
Clear to send	Ring indicator
Data set ready	Secondary receive data
Data terminal ready	Secondary transmit data
Request to send	Transmit clock
Receive clock	Transmit data

Parallel I/O—8 input lines, 8 output lines, all signals RS232C compatible

I/O Addressing

The USART, interval timer, interrupt controller, and parallel interface registers of the iSBC 534 are configured as a block of 16 I/O address locations. The location of this block is jumper selectable to begin at any 16-byte I/O address boundary (i.e., 00H, 10H, 20H, etc.).

I/O Access Time

400 ns	USART registers
400 ns	Parallel I/O registers
400 ns	Interval timer registers
400 ns	Interrupt controller registers

Compatible Opto-Isolators

Function	Supplier	Part Number
Driver	Fairchild General Electric Monsanto	4N33
Receiver	Fairchild General Electric Monsanto	4N37

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.50 in. (1.27 cm)
Weight: 14 oz. (398 gm)

Electrical Characteristics

Average DC Current

Voltage	Without Opto-Isolators	With Opto-Isolators(1)
V_{CC} = +5V	1.9 A, max	1.9 A, max
V_{DD} = +12V	275 mA, max	420 mA, max
V_{AA} = −12V	250 mA, max	400 mA, max

NOTE:
1. With four 4N33 and four 4N37 opto-isolator packages installed in sockets provided to implement four 20 mA current loop interfaces.

Environmental Characteristics

Operating Temperature: 0°C to +55°C

Reference Manual

502140—iSBC 534 Hardware Reference Manual (NOT SUPPLIED)

Reference manuals are shipped with each product only if designated SUPPLIED (see above). Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

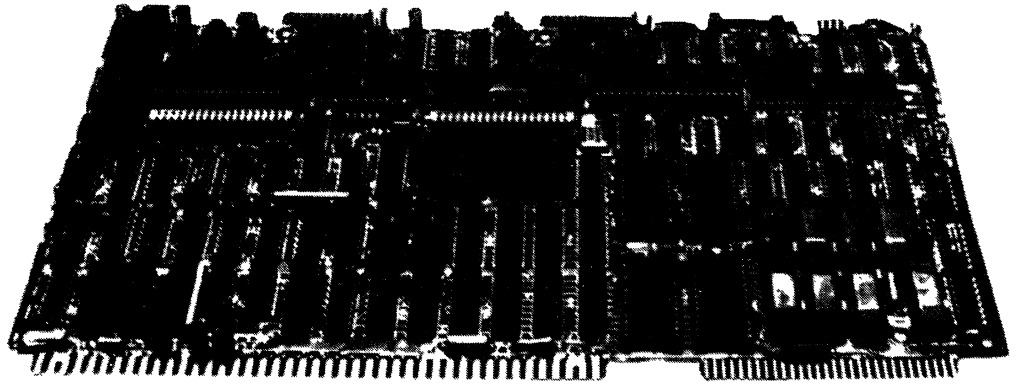
ORDERING INFORMATION

Order Code	Description
SBC534	Four Channel Communication Expansion Board

iSBC® 88/45*
ADVANCED DATA COMMUNICATIONS PROCESSOR BOARD

- **Three HDLC/SDLC Half/Full-Duplex Communication Channels—Optional ASYNC/SYNC on Two Channels**
- **Supports RS232C (Including Modem Support), CCITT V.24, or RS422A/449 Interfaces**
- **On-Board DMA Supports 800K Baud Operation**
- **Self-Clocking NRZI SDLC Loop Data Link Interface**
  - **— Point-to-Point**
  - **— Multidrop**
- **Software Programmable Baud Rate Generation**
- **8088 (8088-2) Microprocessor Operates at 8 MHz**
- **iSBC® 337A Numeric Data Processor Option Supported**
- **16K Bytes Static RAM (12K Bytes Dual-Ported)**
- **Four 28-Pin JEDEC Sites for EPROM/RAM Expansion; Four Additional 28-Pin JEDEC Sites Added with iSBC® 341 Board**
- **Two SBX Bus (IEEE 959) Connectors**
- **MULTIBUS® Interface Supports Multimaster Configuration**

The iSBC 88/45 Advanced Data Communications Processor (ADCP) Board adds 8 MHz, 8088 (8088-2) 8-bit microprocessor-based communications flexibility to the Intel line of MULTIBUS OEM microcomputer systems. The iSBC 88/45 ADCP board offers asynchronous, synchronous, SDLC, and HDLC serial interfaces for gateway networking or general purpose solutions. The iSBC 88/45 ADCP board provides the CPU, system clock, EPROM/RAM, serial I/O ports, priority interrupt logic, and programmable timers to facilitate higher-level application solutions.

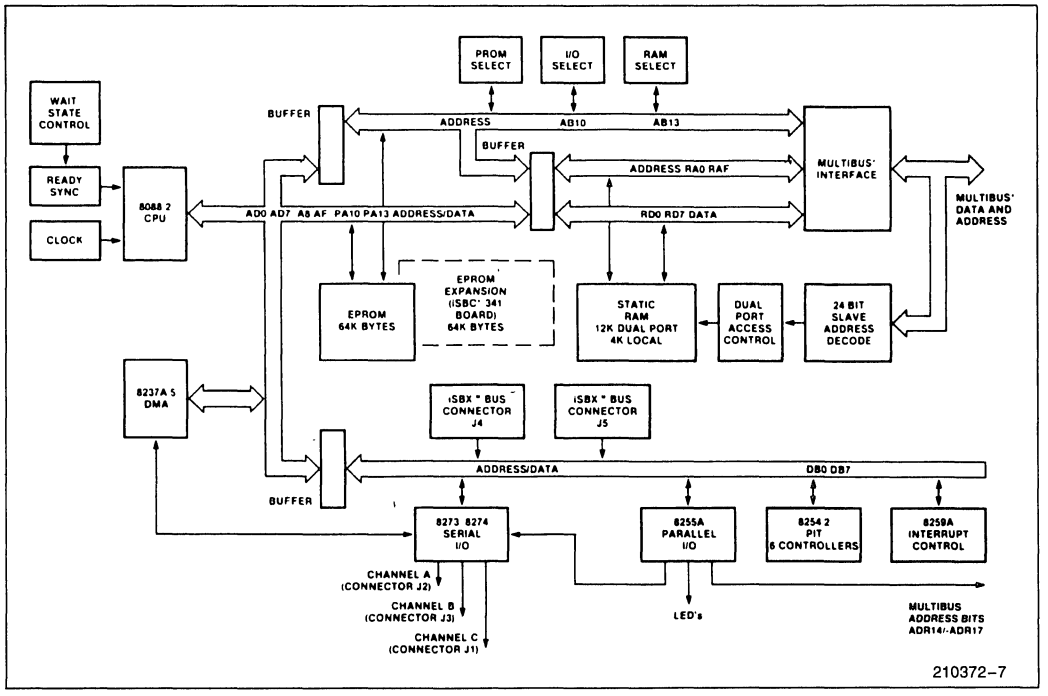


210372-1

*The iSBC® 88/45 is also manufactured under product code piSBC® 88/45 or siSBC® 88/45 by Intel Puerto Rico, Inc. or Intel Singapore, Ltd.

September 1989
Order Number: 210372-003

BLOCK DIAGRAM



Block Diagram of the iSBC® 88/45 ADCP Board

SPECIFICATIONS

Word Size

Instruction: 8, 16, 24, or 32 bits
Data: 8 or 16 bits

System Clock

8 MHz: ±0.1%

NOTE:
Jumper selectable for 4 MHz operation with iSBC 337 Numeric Data Processor module or ICE-88 product.

Cycle Time

Basic Instruction Cycle at 8.00 MHz: 1.25 μs, 250 ns (assumes instruction in the queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles).

Memory Cycle Time

RAM: 500 ns (no wait states)
EPROM: jumper selectable from 500 ns to 625 ns.

On-Board RAM*

K Bytes	Hex Address Range
16 (total) 12 (dual-ported)	0000–3FFF 1000–3FFF

*Four iSBC 88/45 EPROM sockets support JEDEC 24/28-pin standard EPROMs and RAMs (3 sockets); iSBC 341 (4 sockets)

Environmental Characteristics

Temperature: 0°C to +55°C, free moving air across the base board and MULTIMODULE board

Humidity: 90%, non-condensing

Physical Characteristics

Width: 30.48 cm (12.00 in)
Length: 17.15 cm (6.75 in)
Height: 1.50 cm (0.59 in)
Weight: 6.20 gm (22 oz)

Configurations

iSBC® 88/45 Supported Configurations

Connection	Synchronous		Asynchronous	
	Modem	Direct	Modem*	Direct
Point-to-Point	X**	X	X	X
Multidrop	X	X	X	X
Loop	N.A.	N.A.	C (Only)	C (Only)

*Modem should not respond to break.
**Channels A, B, and C denoted by X.

Memory Capacity/Addressing

On-Board EPROM*

Device	Total K Bytes	Hex Address Range
2716	8	FE000–FFFFF
2732A	16	FC000–FFFFF
2764	32	F8000–FFFFF
27128	64	F0000–FFFFF

With optional iSBC® 341 MULTIMODULE™ EPROM

Device	Total K Bytes	Hex Address Range
2716	16	FC000–FFFFF
2732A	32	F8000–FFFFF
2764	64	F0000–FFFFF
27128	128	E0000–FFFFF

*Four iSBC 88/45 EPROM sockets support JEDEC 24/28-pin standard EPROMs and RAMs (static and iRAM, 3 sockets); iSBC 341 sockets also support EPROMs and RAMs.

Timer Input Frequency—8.00 MHz ±0.1%

Interfaces

IEEE 959 SBX I/O Bus Expansion

Serial RS232C Signals—

CTS	CLEAR TO SEND
DSR	DATA SET READY
DTE TXC	TRANSMIT CLOCK
DTR	DATA TERMINAL READY
FG	FRAME GROUND
RTS	REQUEST TO SEND
RXC	RECEIVE CLOCK
RXD	RECEIVE DATA
SG	SIGNAL GROUND
TXD	TRANSMIT DATA

Serial RS422A/449 Signals—

CS	CLEAR TO SEND
DM	DATA MODE
RC	RECEIVE COMMON
RD	RECEIVE DATA
RS	REQUEST TO SEND
RT	RECEIVE TIMING
SC	SEND COMMON
SD	SEND DATA
SG	SIGNAL GROUND
TR	TERMINAL READY
TT	TERMINAL TIMING

Electrical Characteristics

DC Power Dissipation—28.3 Watts

DC Power Requirements

Configuration	Current Requirements (All Voltages ±5%) +5V	+12V	−12V
Without EPROM(1)	5.1A	20 mA	20 mA
With 8K EPROM (Using 2716)	+0.14A	—	—
With 16K EPROM (Using 2732A)	+0.20A	—	—
With 32K EPROM (Using 2764)	+0.24A	—	—
With 64K EPROM (Using 27128)	+0.24A	—	—

NOTE:
1. AS SHIPPED—no EPROMs in sockets, no iSBC 341 module. Configuration includes terminators for two RS422A/449 and one RS232C channels.

Serial Communication Characteristics

Channel	Device	Supported Interface	Max. Baud Rate
A	8274(1)	RS442A/449 RS232C CCITT V.24	800K SDLC/HDLC 125K Synchronous 50K Asynchronous
B	8274	RS232C CCITT V.24	125K Synchronous(2) 50K Asynchronous
C	8273(3)	RS442A/449 RS232C CCITT V.24	64K SDLC/HDLC(3) 9.6K SELF CLOCKING

NOTES:
1. 8274 supports HDLC/SDLC/SYNC/ASYNC multiprotocol
2. Exceed RS232C/CCITT V.24 rating of 20K baud
3. 8273 supports HDLC/SDLC

BAUD RATE EXAMPLES (Hz)

8254 Timer Divide Count N	Synchronous K Baud	Asynchronous ÷16 K Baud	Asynchronous ÷32 K Baud	Asynchronous ÷64 K Baud
10	800	50.0	25.0	12.5
26	300	19.2	9.6	4.8
31	256	16.1	8.06	4.03
52	154	9.6	4.8	2.4
104	76.8	4.8	2.4	1.2
125	64	4.0	2.0	1.0
143	56	3.5	1.7	0.87
167	48	3.0	1.5	0.75
417	19.2	—	—	—
833	9.6	—	—	—
EQUATION	$\frac{8{,}000{,}000}{N}$	$\frac{500K}{N}$	$\frac{250K}{N}$	$\frac{125K}{N}$

Line Drivers (Supplied)

Device	Characteristic	Qty	Installed
1488	RS232C	3	1
1489	RS232C	3	1
3486	RS422A	2	2
3487	RS422A	2	2

Reference Manual

143824—iSBC 88/45 Advanced Data Communications Processor Board Hardware Reference Manual (not supplied).

Reference manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, CA 95051.

ORDERING INFORMATION

Order Code	Description
SBC88/45	iSBC 88/45 Advanced Data Communications Processor Board

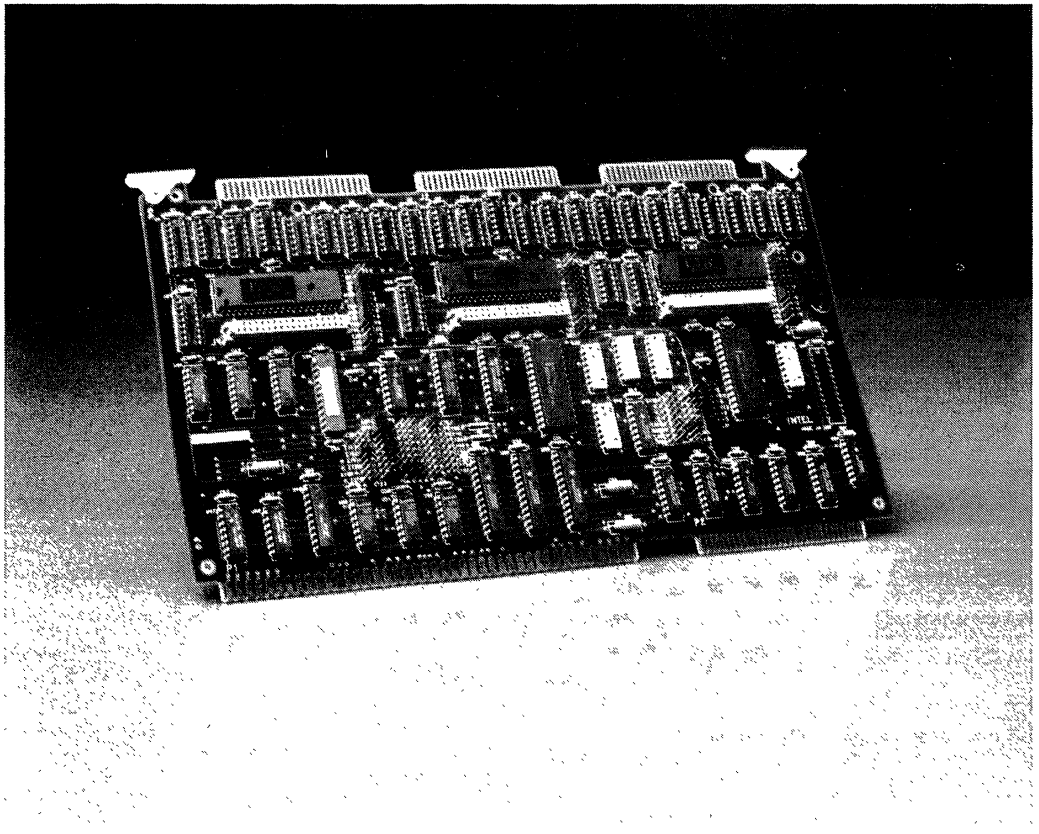
MULTIBUS® I Digital and Analog I/O Boards

12

iSBC® 519A
PROGRAMMABLE I/O EXPANSION BOARD

- **I/O Expansion via Direct MULTIBUS® Interface**
- **72 Programmable I/O Lines with Sockets for Interchangeable Line Drivers and Terminators**
- **Provides Full 16-Bit I/O Addressability**
- **Provides 3 iSBX Multimodule Connectors**
- **Provides 16 Maskable Interrupt Request Lines**
- **Jumper Selectable 0.5, 1.0, 2.0, or 4.0 ms Interval Timer**
- **Provides Eight Maskable Interrupt Request Lines with Priority Encoded and Programmable Interrupt Algorithms**

The iSBC 519A Programmable I/O Expansion Board is a member of Intel's complete line of iSBC memory and I/O expansion boards. The iSBC 519A interfaces directly to any iSBC single board computer via the system bus to expand input and output port capacity. The iSBC 519A provides 72 programmable I/O lines. The system software is used to configure the I/O lines to meet a wide variety of peripheral requirements. The flexibility of the I/O interface is further enhanced by the capability of selecting the appropriate combination of optional line drivers and terminators to provide the required sink current, polarity, and drive/termination characteristics for each application. Address selection is accomplished by using wire-wrap jumpers. The board operates with a single +5V power supply.



280230-1

October 1989
Order Number: 280230-002

FUNCTIONAL DESCRIPTION

The 72 programmable I/O lines on the iSBC 519A are implemented utilizing three Intel 82C55A programmable peripheral interfaces. The system software is used to configure the I/O lines in combinations of undirectional input/output and bidirectional ports. In order to take full advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators. The 72 programmable I/O lines and signal ground lines are brought out to three 50-pin edge connectors that mate with flat, round, or woven cable.

Interval Timer

Typical I/O read access time is 350 nanoseconds.

Typical I/O read/write cycle time is 450 nanoseconds. The interval timer provided on the iSBC 519A may be used to generate real time clocking in systems requiring the periodic monitoring of I/O functions. The time interval is derived from the constant clock (BUS CCLK) and the timing interval is jumper selectable. Intervals of 0.5, 1.0, 2.0, and 4.0 milliseconds may be selected when an iSBC single board computer is used to generate the clock. Other timing intervals may be generated if the user provides a separate constant clock reference in the system.

Eight-Level Vectored Interrupt

Two Intel 82C59A programmable interrupt controllers (PIC) provide vectoring for interrupt levels. As shown in Table 1, a selection of three priority processing algorithms is available to the system designer so that the manner in which requests are serviced may be configured to match system requirements. Priority assignments may be reconfigured dynamically via software at any time during system operation.

Table 1. Interrupt Priority Options

Algorithm	Operation
Fully nested	Interrupt request line priorities fixed.
Auto-rotating	Equal priority. Each level, after receiving service, becomes the lowest priority level until next interrupt occurs.
Specific priority	System software assigns lowest priority level. Priority of all other levels are based in sequence numerically on this assignment.

SBX Expansion Bus (IEEE 959) Capabilities

Three SBX bus connectors are provided on the iSBC 519A board. Up to three single-wide or one double-wide and one single-wide IEEE 959 SBX board can be added to the iSBC 519A board. A wide variety of expansion options are available.

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.50 in. (1.27 cm),
1.16 in. (2.95 cm) with iSBX modules
Weight: 14 oz. (397.3 gm)

Electrical Characteristics

Average DC Current

Voltage	Without Termination(1)	With Termination(2)
V_{CC} = +5V ±5%	I_{CC} = 1.5A max	3.5A max

NOTES:
1. Does not include power required for operational I/O drivers and I/O terminators.
2. With 18 220Ω/330Ω input terminators installed, all terminator inputs low.

Environmental Characteristics

Operating Temperature: 0°C to +55°C

Reference Manual

9800385B—iSBC iSBC 519A Hardware Reference manual (NOT SUPPLIED)

Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

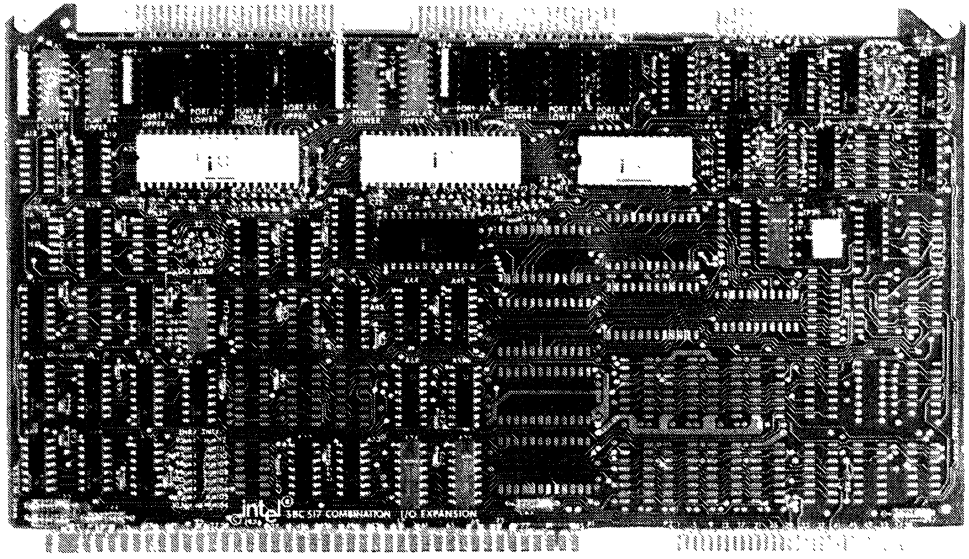
ORDERING INFORMATION

Order Code	Description
SBC519A	Programmable I/O Expansion Board

iSBC® 517*
COMBINATION I/O EXPANSION BOARD

- **48 Programmable I/O Lines with Sockets for Interchangeable Line Drivers and Terminators**
- **Synchronous/Asynchronous Communications Interface with RS232C Drivers and Receivers**
- **Eight Maskable Interrupt Request Lines with a Pending Interrupt Register**
- **1 ms Interval Timer**

The iSBC 517 Combination I/O Expansion Board is a member of Intel's complete line of iSBC memory and I/O expansion boards. The board interfaces directly with any iSBC single board computer via the system bus to expand serial and parallel I/O capacity. The combination I/O board contains 48 programmable parallel I/O lines. The system software is used to configure the I/O lines to meet a wide variety of system peripheral requirements. The flexibility of the I/O interface is significantly enhanced by the capability of selecting the appropriate combination of optional line drivers and terminators to provide the required sink current, polarity, and drive/termination characteristics for each application. A programmable RS232C communications interface is provided on the iSBC 517. This interface may be programmed by the system software to provide virtually any asynchronous or synchronous serial data transmission technique (including IBM Bi-Sync). A comprehensive RS232C interface to CRTs, RS232C compatible cassettes, and asynchronous and synchronous modems is thus on the board. An on-board register contains the status of eight interrupt request lines which may be interrogated from the system bus, and each interrupt request line is maskable under program control. The iSBC 517 also contains a jumper selectable 1 ms interval timer and interface logic for eight interrupt request lines.

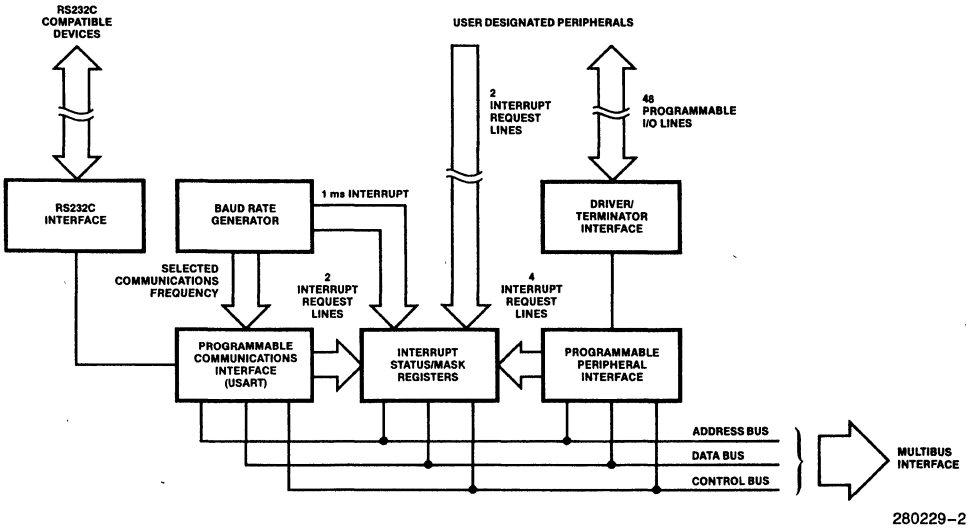


280229-1

*The iSBC® 517 is also manufactured under product code piSBC® 517 by Intel Puerto Rico, Inc.

September 1989
Order Number: 280229-002

BLOCK DIAGRAM



NOTE:
Interrupts originating from the programmable communications interface and programmable peripheral interface are jumper selectable.

iSBC® 517 Combination I/O Expansion Board Block Diagram

SPECIFICATIONS

Parallel I/O Port Operation Modes

Ports	Lines (qty)	Mode of Operation					
		Unidirectional				Bidirectional	Control
		Input		Output			
		Unlatched	Latched & Strobed	Latched	Latched & Strobed		
1	8	X	X	X	X	X	
2	8	X	X	X	X		
3	4	X		X			X(1)
	4	X		X			X(1)
4	8	X	X	X	X	X	
5	8	X	X	X	X		
6	4	X		X			X(2)
	4	X		X			X(2)

NOTES:
1. Part of port 3 must be used as control port when either port 1 or port 2 are used as a latched and strobed input or a latched and strobed output port or port 1 is used as a bidirectional port.
2. Part of port 6 must be used as a control port when either port 4 or port 5 are used as a latched and strobed input or a latched and strobed output port or port 4 is used as a bidirectional port.

I/O Addressing

Port	1	2	3	4	5	6	8255 No. 1 Control	8255 No. 2 Control	USART Data	USART Control
Address	X4	X5	X6	X8	X9	XA	X7	XB	XC	XD

NOTE:
X is any hex digit assigned by jumper selection.

I/O Transfer Rate

Parallel—Read or write cycle time 760 ns max
Serial—(USART)

Frequency (kHz) (Jumper Selectable)	Baud Rate (Hz)		
	Synchronous	Asynchronous (Program Selectable)	
		÷16	÷64
153.6	—	9600	2400
76.8	—	4800	1200
38.4	38400	2400	600
19.2	19200	1200	300
9.6	9600	600	150
4.8	4800	300	75
6.98	6980	—	110

Serial Communications Characteristics

Synchronous—5–8 bit characters; internal or external character synchronization; automatic sync insertion.

Asynchronous—5–8 bit characters; peak characters generation; 1, 1½, or 2 stop bits; false start bit detectors.

Interrupts

Eight interrupt request lines may originate from the programmable peripheral interface (4 lines), the USART (2 lines), or user specified devices via the I/O edge connector (2 lines) or interval timer.

Interrupt Register Address

X1 Interrupt mask register
X0 Interrupt status register

NOTE:
X is any hex digit assigned by jumper selection.

Timer Interval

1.003 ms ±0.1% when 110 baud rate is selected
1.042 ms ±0.1% for all other baud rates

Line Drivers and Terminators

I/O Drivers—The following line drivers and terminators are compatible with all the I/O driver sockets on the iSBC 517.

Driver	Characteristics	Sink Current (mA)
7438	I,OC	48
7437	I	48
7432	NI	16
7426	I,OC	16
7409	NI,OC	16
7408	NI	16
7403	I,OC	16
7400	I	16

NOTE:
I = Inverting; NI = non-inverting; OC = open-collector.

Ports 1 and 4 have 25 mA totem-pole drivers and 1 kΩ terminators.

Line Terminators—220Ω/330Ω divider or 1 kΩ pull-up—user supplied

Bus Drivers

Function	Characteristics	Sink Current (mA)
Data	Tri-state	50
Commands	Tri-state	25

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.50 in. (1.27 cm)
Weight: 14 oz. (397.3 gm)

Electrical Characteristics

Average DC Current
V_{CC} = +5V ±5%
V_{DD} = +12V ±5%
V_{AA} = −12 ±5%
I_{CC} = 2.4 mA max
I_{DD} = 40 mA max
I_{AA} = 60 mA max

NOTE:
Does not include power required for optional I/O drivers and I/O terminators. With eight 220Ω/330Ω input terminators installed, all terminator inputs low.

Environmental Characteristics

Operating Temperature—0°C to +55°C

Reference Manual

9800388B—iSBC 517 Hardware Reference manual (NOT SUPPLIED)

Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

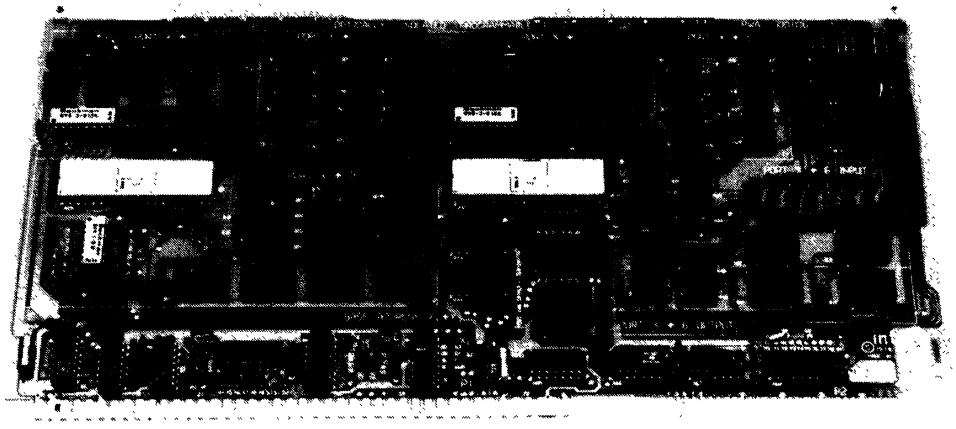
ORDERING INFORMATION

Order Code	Description
SBC517	Combination I/O Expansion Board

iSBC® 556*
OPTICALLY ISOLATED I/O BOARD

- **Up to 48 Digital Optically Isolated Input/Output Data Lines for MULTIBUS® Systems**
- **Choice of**
  - **— 24 Fixed Input Lines**
  - **— 16 Fixed Output Lines**
  - **— 8 Programmable Lines**
- **Provisions for Plug-In, Optically Isolated Receivers, Drivers, and Terminators**
- **Voltage/Current Levels**
  - **— Input up to 48V**
  - **— Output up to 30V, 60 mA**
- **Common Interrupt for up to 8 Sources**
- **+5V Supply Only**

The iSBC 556 Optically Isolated I/O Board provides 48 digital input/output lines with isolation between process application or peripheral device and the system CPU board(s). The iSBC 556 contains two 8255A programmable interface devices, and sockets for user supplied optically isolated drivers, receivers, and input resistor terminators, together with common interrupt logic and interface circuitry for the system bus. Input signals can be single-ended or differential types with user defined input range (resistor terminator and opto-isolated receiver selection), allowing flexibility in design of voltage and threshold levels. The output allows user selection of Opto-Isolated Darlington Pair which can be used as an output driver either as an open collector or current switch.



280231-1

*The iSBC® 556 is also manufactured under product code piSBC® 556 by Intel Puerto Rico, Inc.

October 1989
Order Number: 280231-002

Table 1. I/O Ports Opto-Isolator Receivers, Drivers, and Terminators

Port No. X = I/O Base Address	Type of I/O	Lines (qty)	Resistor Terminator Pac-Rp 16-Pin DIP Bourns 4116R-00 or Equivalent	Dual Opto-Isolator 8-Pin DIP Monsanto MC T66 or Equivalent	Driver 7438 or Equivalent	Pull-Up iSBC® 902
X + 0	Input	8	1	4	—	
X + 1	Output	8	—		—	
X + 2	Input/ Control	8	1		—	
X + 4	Input	8	1	4	—	
X + 5	Output	8			—	
X + 6	Input/ Output	8	1 if input		2 if input	2 if input
X + 7	Control					

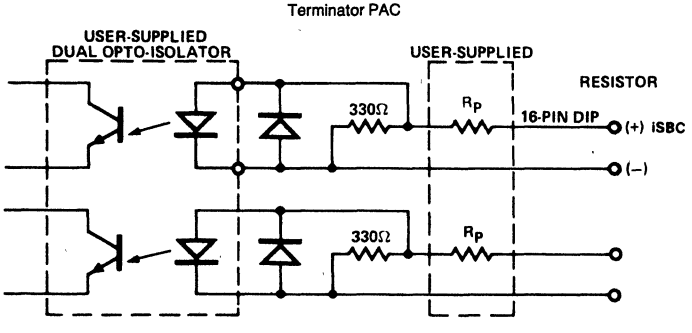
SPECIFICATIONS

Number of Lines

24 input lines
16 output lines
8 programmable lines: 4 input — 4 output

I/O Interface Characteristics

Line-to-Line Isolation: 235V DC or peak AC
Input/Output Isolation: 500V DC or peak AC



RP determines voltage and current range.

I/O Addressing

Port	8255 #1			Control	8255 #2			Control
	A	B	C		A	B	C	
Address	X+0	X+1	X+2	X+3	X+4	X+5	X+6	X+7

Where: base address is from 00H to 1FH (jumper selectable)

Physical Characteristics

Width: 12.00 in. (30.48 cm)

Height: 6.75 in. (17.15 cm)

Depth: 0.50 in. (1.27 cm)

Weight: 12 oz. (397.3 gm)

Electrical Characteristics

Average DC Current

V_{CC} = +5V ±5%, 1.0A without user supplied isolated receiver/driver

I_{CC} = 1.6A max with user supplied isolator receiver/driver

Environmental Characteristics

Temperature: 0°C to 55°C

Relative Humidity: 0% to 90%, non-condensing

Reference Manual

502170— iSBC 556 Hardware Reference Manual (Order Separately)

Reference manuals are shipped with each product only if designated SUPPLIED (see above). Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

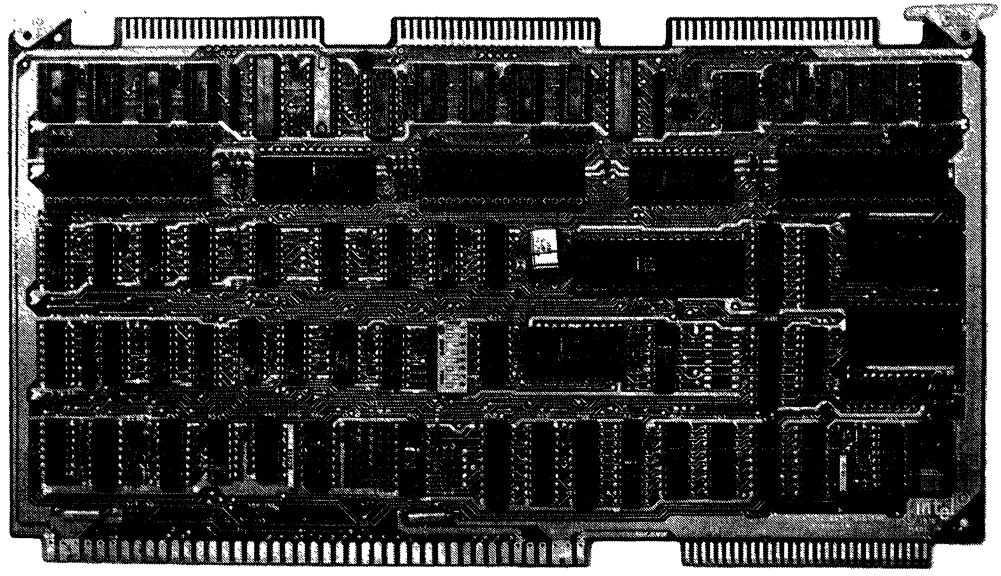
ORDERING INFORMATION

Order Code	Description
SBC556	Optically Isolated I/O Board

iSBC® 569*
INTELLIGENT DIGITAL CONTROLLER

- **Single Board Digital I/O Controller with up to Four Microprocessors to Share the Digital Input/Output Signal Processing**
- **3 MHz 8085A Central Control Processor**
- **Three Sockets for 8041/8741A Universal Peripheral Interface (UPI-41A) for Distributed Digital I/O Processing**
- **Three Operational Modes**
  - **— Stand-Alone Digital Controller**
  - **— MULTIBUS® Master**
  - **— Intelligent Slave (Slave to MULTIBUS Master)**
- **2K Bytes of Dual Port Static Read/Write Memory**
- **Sockets for up to 8K Bytes of Intel 2758, 2716, 2732 Erasable Programmable Read Only Memory**
- **48 Programmable Parallel I/O Lines with Sockets for Interchangeable Line Drivers or Terminators**
- **Three Programmable Counters**
- **12 Levels of Programmable Interrupt Control**
- **Single +5V Supply**
- **MULTIBUS Standard Control Logic Compatible with Optional iSBC 80 and iSBC® 86 CPU, Memory, and I/O Expansion Boards**

The Intel iSBC® 569 Intelligent Digital Controller is a single board computer (8085A based) with sockets for three 8041A/8741A Universal Peripherals Interface chips (UPI-41A). These devices, which are programmed by the user, may be used to offload the 8085A processor from time consuming tasks such as pulse counting, event sensing and parallel or serial digital I/O data formatting with error checking and handshaking. The iSBC 569 board is a complete digital controller with up to four processors on a single 6.75 inches x 12.00 inches (17.15 cm x 30.48 cm) printed circuit board. The 8085A CPU, system clock, read/write memory, non-volatile memory, priority interrupt logic, programmed timers, MULTIBUS control and interface logic, optional UPI processors and optional line driver and terminators all reside on one board.

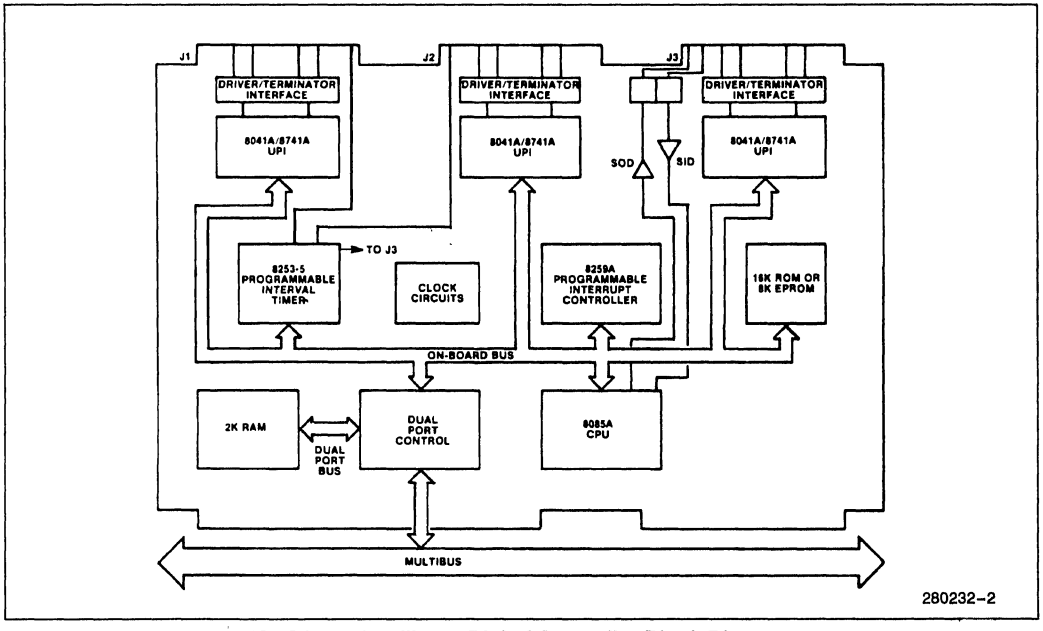


280232-1

*The iSBC® 569 is also manufactured under product code piSBC® 569 by Intel Puerto Rico, Inc.

October 1989
Order Number: 280232-002

BLOCK DIAGRAM



iSBC® 569 Intelligent Digital Controller Block Diagram

SPECIFICATIONS

8085A CPU

Word Size: 8, 16 or 24 bits

Cycle Time: 1.30 μs ±0.1% for fastest executable instruction; i.e., four clock cycles.

Clock Rate: 3.07 MHz ±0.1%

System Access Time

Dual port memory—725 ns

Memory Capacity

On-board ROM/EPROM—2K, 4K, 8K, or 16K bytes of user installed ROM or EPROM.

On-board RAM—2K bytes of static RAM. Fully accessible from on-board 8085A. Separately addressable from system bus.

Off-board expansion—up to 64K bytes of EPROM/ROM or RAM capacity.

I/O Capacity

Parallel-Timers—Three timers, with independent gate input, clock input, and timer output user-accessible. Clock inputs can be strapped to an external source or to an on-board 1.3824 MHz reference. Each timer is connected to a 8259A Programmable Interrupt Controller and may also be optionally connected to UPI processors.

UPI-I/O—Three UPI-41A interfaces, each with two 8-bit I/O ports plus the two UPI Test Inputs. The 8-bit ports are user-configurable (as inputs or outputs) in groups of four.

Serial—1 TTL compatible serial channel utilizing SID and SOD lines of on-board 8085A CPU.

On-Board Addressing

All communications to the UPI-41A processors, to the programmable reset latch, to the timers, and to the interrupt controller are via read and write commands from the on-board 8085A CPU.

Memory Addressing

On-board ROM/EPROM—0-07FF (using 2758 EPROMs); 0-OFFF (using 2716 EPROMs or 2316 ROMs); 0-1FFF (using 2732 EPROMs); 0-3FFF (using the 2364 ROMs)

On-board RAM—8000-87FF System access—any 2K increment 00000-FF800 (switch selection); 1K bytes may be disabled from bus access by switch selection.

I/O Addressing

Source	Addresses
8253	0E0H-0E3H
UPI0	0E4H-0E5H
UPI1	0E6H-0E7H
UPI2	0E8H-0E9H
PROGRAMMABLE RESET	0EAH-0EBH
8259A	0ECH-0EDH

Timer Specifications

Input Frequencies—jumper selectable reference

Internal: 1.3824 MHz ±0.1% (0.723 μs, nominal)
External: User supplied (2 MHz maximum)

Output Frequencies (at 1.3824 MHz)

Function	Min[1]	Max[1]
Real-time interrupt interval	1.45 μsec	47.4 msec
Rate Generator (frequency)	21.09 Hz	691.2 KHz
1. Single 16-bit binary count		

Physical Characteristics

Width: 30.48 cm (12.00 inches)
Depth: 17.15 cm (6.75 inches)
Thickness: 1.27 cm (0.50 inch)
Weight: 3.97 gm (14 ounces)

Electrical Characteristics

DC Power Requirements—+5V @ 2.58A with no optional devices installed. For each 8741A add 135 mA. For each 220/330 resistor network, add 60 mA. Add the following for each EPROM/ROM installed.

Type	+5.0V Current Requirement	
	1ROM	2ROM
2758	100 mA	125 mA
2716	100 mA	125 mA
2316E	120 mA	240 mA
2732	40 mA	55 mA
2364	40 mA	55 mA

Line Drivers and Terminators

I/O /Drivers—The following line drivers are all compatible with the I/O driver sockets on the iSBC 569 Intelligent Digital Controller.

Driver	Characteristics	Sink Current (mA)
7438	I,OC	48
7437	I	48
7432	NI	16
7426	I,OC	16
7409	NI,OC	16
7408	NI	16
7403	I,OC	16
7400	I	16

NOTE:
I = inverting; NI = non-inverting; OC = open collector.

I/O Terminators—220Ω/330Ω divider or 1 kΩ pull-up (DIP) - user supplied

Environmental Characteristics

Operating Temperature: 0° C to 55° C (32° F to 131 °F)
Relative Humidity: To 90% without condensation

Reference Manual

502180—iSBC 569 Intelligent Digital Controller Board Hardware Reference Manual (NOT SUPPLIED)

Reference manuals are shipped with each product only if designated SUPPLIED (see above). Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

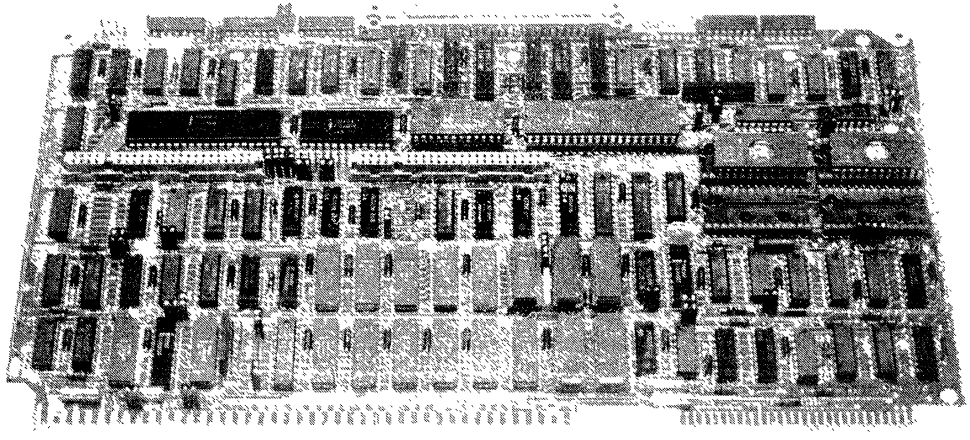
ORDERING INFORMATION

Order Code	Description
SBC569	Intelligent Digital Controller

iSBC® 589*
INTELLIGENT DMA CONTROLLER

- **Configurable as Either an Intelligent Slave or MULTIBUS® Master**
- **5 MHz 8089 I/O Processor**
- **MULTICHANNEL™ DMA I/O Bus Interface with Supervisor, Controller or Basic Talker/Listener Capabilities**
- **Two 8/16-Bit iSBX™ Bus Connectors**
- **DMA Transfer Rates Up to 1.25 Mbytes per Second**
- **User Command Interface Firmware Package Provides High Level I/O Commands**
- **8 Kbytes of High-Speed Dual-Ported Static Read/Write Memory**
- **Sockets for up to 32 Kbytes of Read Only Memory or Additional Byte-Wide Static RAMs**
- **Three Programmable Timers**

The iSBC 589 Intelligent DMA Controller is a member of Intel's complete line of MULTIBUS microcomputer systems which take full advantage of VLSI technology to provide economical computer based solutions for OEM applications. The iSBC 589 board is a general purpose, programmable, high-speed DMA controller on a single 6.75 x 12.00 inch printed circuit board. Using the board's dual-port RAM and standard EPROM resident firmware, the on-board Intel 8089 I/O Processor can perform memory to memory block transfers and complex I/O operations via two iSBX connectors and the MULTICHANNEL I/O bus at DMA transfer rates up to 1.25 Mbytes per second. Acting as an intelligent slave, the iSBC 589 board enhances the system's overall performance by relieving the host CPU of time consuming I/O operations. The board's unique combination of performance, on-board intelligence and flexible hardware I/O interfaces make the iSBC 589 board the ideal solution for applications with specialized I/O requirements such as high-speed data acquisition, graphics, instrument automation and specialized peripheral control, that previously would have necessitated an expensive custom designed I/O controller.

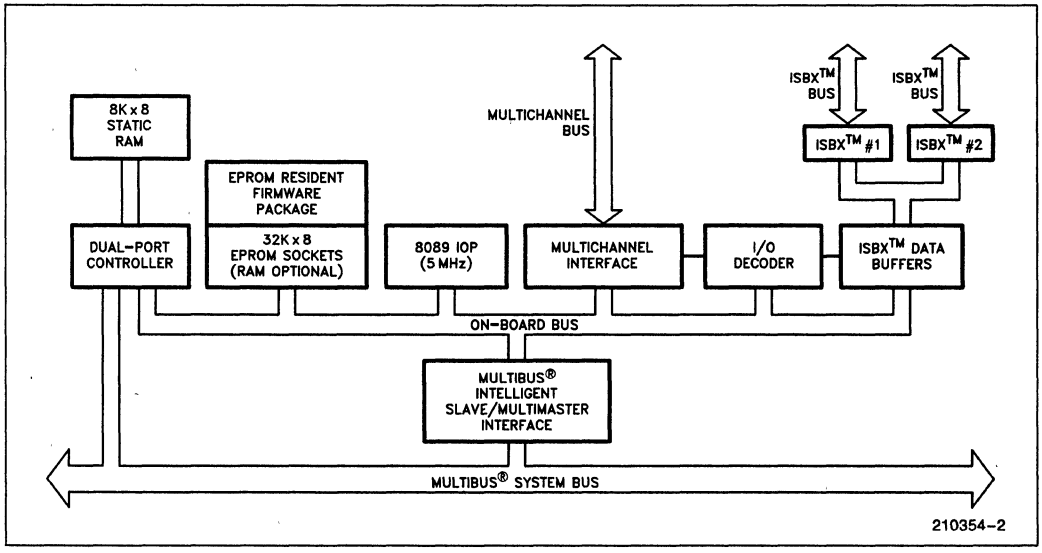


210354-1

*The iSBC® 589 is also manufactured under product code piSBC® 589 by Intel Puerto Rico, Inc.

October 1989
Order Number: 210354-002

BLOCK DIAGRAM



SPECIFICATIONS

8089 IOP

WORD SIZE

Instruction—16 to 40-bits

Data—8, 16-bits

SYSTEM CLOCK

5.0 MHz ± 0.1%

CYCLE TIME

2.2 μs for the fastest instructions

System Access Time

Dual-Port Memory— 550 ns (worst case, without contention from on-board access)

I/O Capacity

MULTICHANNEL I/O Bus— 1 MULTICHANNEL port which supports 8 and 16-bit transfers and can be configured as a Basic Talker/Listener, Controller or Supervisor

iSBX™ MULTIMODULE™— Two (2) iSBX MULTIMODULE boards

I/O Addressing

Interface	I/O Addresses
SBX Connector #1	FF80 thru FF9F
SBX Connector #2	FFA0 thru FFBF
MULTICHANNEL	FFD0 thru FFEE
Interval Timer	FFC8 thru FFCE
Other On-Board Devices	FFC0 thru FFC6
	FFF0 thru FFFE

Memory Capacity

ON-BOARD EPROM

Device	Total Capacity	Address Range
2716	8 Kbytes	$FE000-FFFFF_H$
2732A	16 Kbytes	$FC000-FFFFF_H$
2764	32 Kbytes	$F8000-FFFFF_H$

ON-BOARD RAM

Total Capacity— 8 Kbytes

On-Board Address— $00000-01FFF_H$

MULTIBUS® Address—Jumper selectable on 8 Kbyte boundaries. Default is 0_H.

I/O Transfer Rates (Microseconds/Transfer)

	MULTICHANNEL	iSBX™	MULTIBUS®		On-Board RAM
			Shared	Buslock	
MULTICHANNEL	—	2.0	2.4	2.2	1.8
SBX	2.0	2.0	2.4	2.2	2.0
MULTIBUS (Shared)	2.4	2.4	2.8	—	2.2
MULTIBUS (Buslock)	2.2	2.2	—	2.4	2.0
On-Board RAM	1.8	1.8	2.2	2.0	1.6

Timers

Input Frequencies—Jumper selectable at 1.25 MHz, 625 kHz or 312.5 kHz

Output Frequencies/Timing Intervals—

Function	Single Timer/Counter		Dual Timer/Counter (Two Timers Cascaded)	
	Minimum	Maximum	Minimum	Maximum
Real-Time Delay	1.6 μs	210 ms	3.2 μs	1.37×10^4 sec
Programmable One-Shot	1.6 μs	210 ms	3.2 μs	1.37×10^4 sec
Rate Generator	4.76 Hz	625 kHz	7.3×10^{-5} Hz	312.5 kHz
Square-Wave Rate Generator	4.76 Hz	625 kHz	7.3×10^{-5} Hz	312.5 kHz
Software Triggered Strobe	1.6 μs	210 ms	3.2 μs	1.37×10^4 sec
Hardware Triggered Strobe	1.6 μs	210 ms	3.2 μs	1.37×10^4 sec

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 7.05 in. (17.9 cm)
Depth 0.50 in. (1.27 cm)
Weight: 16 oz. (453.6 gm)

Environmental Characteristics

Operating Temperature— 0°C to +55°C
Relative Humidity — to 90% (without condensation)

Electrical Characteristics

DC POWER REQUIREMENTS

Configuration	Current Requirements (+5V +5% Maximum)
Without EPROM	4.7 Amps
Without 8K EPROM (Using Four 2716s)	5.4 Amps
With 8K EPROM* (Using two 2732As)	5.0 Amps
With 16K EPROM (Using Four 2732As)	5.3 Amps
With 32K EPROM (Using Four 2164s)	5.3 Amps

*Factory Default Configuration

Reference Manuals

142996— iSBC 589 Intelligent DMA Controller Board Hardware Reference Manual (Not Supplied)

Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051

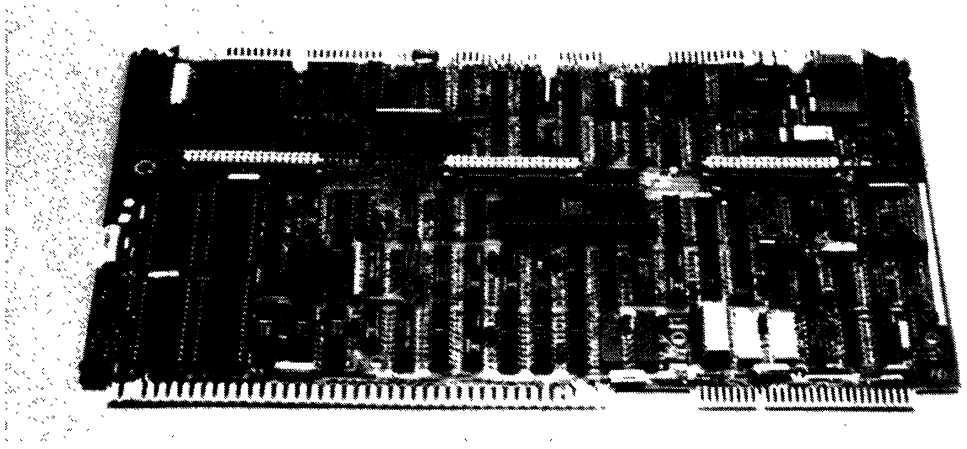
ORDERING INFORMATION

Order Code	Description
SBC589	Intelligent DMA Controller Board

iSBC® 88/40A
MEASUREMENT AND CONTROL COMPUTER

- **High Performance 4.8/6.67 MHz 8088 8-Bit HMOS Processor**
- **12-Bit KHz Analog-to-Digital Converter with Programmable Gain Control**
- **16-Bit Differential/32 Single-Ended Analog Input Channels**
- **Three iSBX™ Bus Connectors for Analog, Digital, and other I/O Expansion**
- **4K Bytes Static RAM, Expandable via iSBC® 301 MULTIMODULE™ RAM to 8K Bytes (1K Byte Dual-Ported)**
- **Four EPROM/E²PROM Sockets for up to 64K Bytes, Expandable to 128K Bytes with iSBC® 341 Expansion MULTIMODULE™**
- **MULTIBUS® Intelligent Slave or Multimaster**

The Intel iSBC 88/40A Measurement and Control Computer is a member of Intel's large family of Single Board Computers that takes full advantage of Intel's VLSI technology to provide an economical self-contained computer based solution for applications in the areas of process control and data acquisition. The on-board 8088 processor with its powerful instruction set allows users of the iSBC 88/40A board to update process loops as much as 5–10 times faster than previously possible with other 8-bit microprocessors. For example, the high performance iSBC 88/40A can concurrently process and update 16 control loops in less than 200 milliseconds using a traditional PID (Proportional-Integral-Derivative) control algorithm. The iSBC 88/40A board consists of a 16 differential/32 single ended channel analog multiplexer with input protected circuits, A/D converter, programmable central processing unit, dual port and private RAM, read only memory sockets, interrupt logic, 24 channels of parallel I/O, three programmable timers and MULTIBUS control logic on a single 6.75 by 12.00-inch printed circuit card. The iSBC 88/40A board is capable of functioning by itself in a stand-alone system or as a multimaster or intelligent slave in a large MULTIBUS system.



280220–1

September 1989
Order Number: 280220-002

BLOCK DIAGRAM

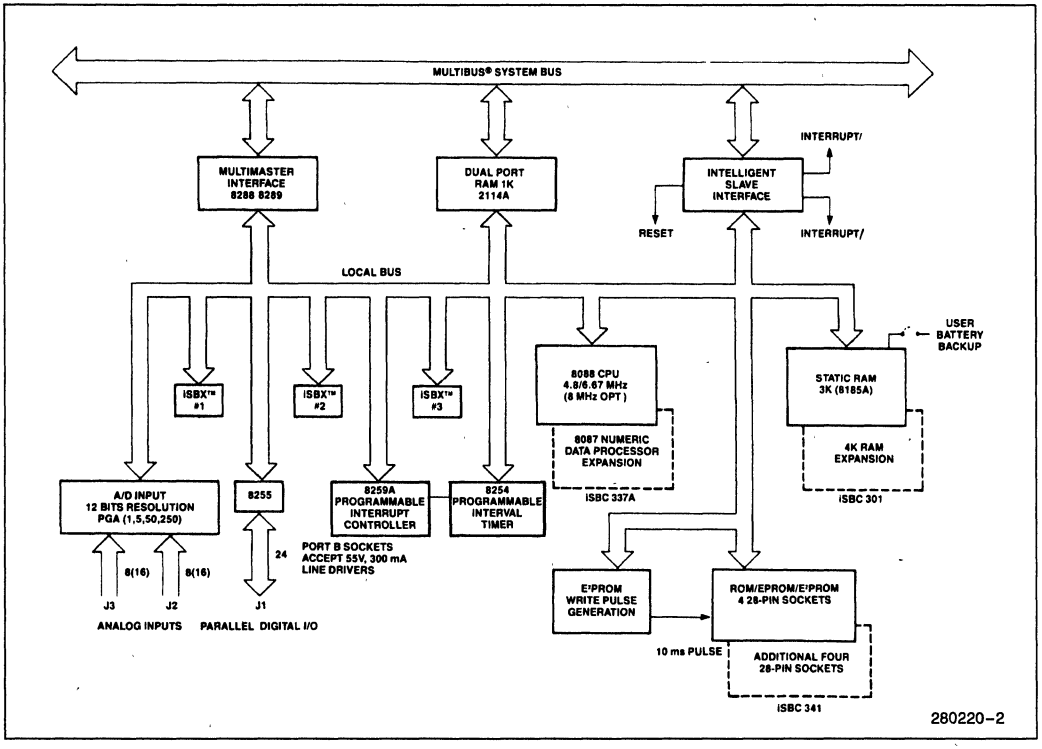


Figure 1. iSBC® 88/40A Measurement and Control Computer Block Diagram

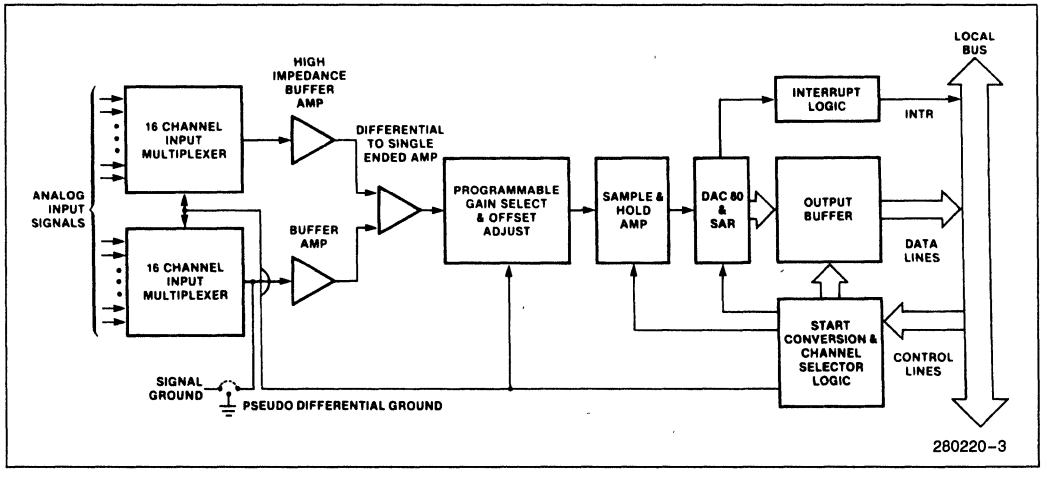


Figure 2. iSBC® 88/40 Analog Input Section

SPECIFICATIONS

Parallel I/O Port Operation Modes

Port	Lines (qty)	Mode of Operation					
		Unidirectional				Bidirectional	Control
		Input		Output			
		Latched	Latched & Strobed	Latched	Latched & Strobed		
1	8	X	X	X	X	X	
2	8	X	X	X	X		
3	4	X		X			X(1)
	4	X		X			X(1)

NOTE:
1. Part of port 3 must be used as a control port when either port 1 or port 2 are used as a latched and strobed input or a latched and strobed output port or port 1 is used as a bidirectional port.

Word Size

Instruction—8, 16, or 32 bits
Data—8 bits

Instruction Cycle Time (minimum)

Instruction	8088 Clock Rate			Number of Clock Cycles
	4.8 MHz	6.67 MHz	8.0 MHz	
In Queue	417 ns	300 ns	250 ns	2
Not in Queue	1.04 ns	750 ns	625 ns	5

MEMORY CAPACITY

On-Board ROM/EPROM/E²PROM

Up to 64K bytes; user installed in 2K, 4K, 8K or 16K byte increments or up to 128K if iSBC 341 MULTIMODULE EPROM option installed. Up to 8K bytes of E²PROM using Intel 2816As or 2817As may be user-installed in increments of 2, 4, or 8 bytes.

On-Board RAM

4K bytes or 8K bytes if the iSBC 301 MULTIMODULE RAM is installed. Integrity maintained during power failure with user-furnished batteries. 1K bytes are dual-ported.

Off-Board Expansion

Up to 1 megabyte of user-specified combination of RAM, ROM, and EPROM.

MEMORY ADDRESSING

On-Board ROM/EPROM

FE000-FFFFF (using 2716 EPROMs)
FC000-FFFFF (using 2732 EPROMs)
F8000-FFFFF (using 2764 EPROMs)
F0000-FFFFF (using 27128 EPROMs)

On-Board ROM/EPROM (With iSBC 341 MULTIMODULE EPROM option installed)

FC000-FFFFF (using 2716 EPROMs)
F8000-FFFFF (using 2732 EPROMs)
F0000-FFFFF (using 2764 EPROMs)
E0000-FFFFF (using 27128 EPROMs)

On-Board RAM (CPU Access)

00000-00FFF
00000-01FFF (if iSBC 301 MULTIMODULE RAM option installed)

On-Board RAM

Jumpers allow 1K bytes of RAM to act as slave RAM for access by another bus master. Addressing may be set within any 1K boundary in the 1-megabyte system address space.

Slave RAM Access

Average: 350 ns

INTERVAL TIMER

Output Frequencies

Function	Single Timer		Dual Timers (Two Timers Cascaded)
	Min	Max	
Real-Time Interrupt Interval	0.977 μs	64 ms	69.9 minutes maximum
Rate Generator (Frequency	15.625 Hz	1024 KHz	0.00024 Hz minimum

CPU CLOCK

4.8 MHz ±0.1% or 6.67 MHz ±0.1%. (User selectable via jumpers);

8.0 MHz (with user installed 24 MHz oscillator)

I/O Addressing

All communications to parallel I/O ports, iSBX bus, A/D port, timers, and interrupt controller are via read and write commands from the on-board 8088 CPU.

Interface Compatability

Parallel I/O—24 programmable lines (8 lines per port); one port includes a bidirectional bus driver. IC sockets are included for user installation of line drivers and/or I/O terminators and/or peripheral drivers as required for interface ports.

Interrupts

8088 CPU includes a non-maskable interrupt (NMI). NMI interrupt is provided for catastrophic events such as power failure. The on-board 8259A PIC provides 8-bit identifier of interrupting device to CPU. CPU multiplies identifier by four to derive vector address. Jumpers select interrupts from 26 sources without necessity of external hardware. PIC may be programmed to accommodate edge-sensitive or level-sensitive inputs.

Analog Input

16 differential (bipolar operation) or 32 single-ended (unipolar operation).

Full Scale Voltage Range—−5 to +5 volts (bipolar), 0 to +5 volts (unipolar).

NOTE:
Ranges of 0 to 10V and ±10V achievable with externally supplied ±15V power.

Gain—Program selectable for gain of 1, 5, 50, or 250.

Resolution—12 bits (11 bits plus sign for ±5, ±10 volts).

Accuracy—Including noise and dynamic errors.

Gain	25°C
1	±0.035% FSR*
5	±0.06% FSR*
50	±0.07% FSR*
250	±0.12% FSR*

NOTE:
FSR = Full Scale Range ±½ LSB. Figures are in percent of full scale reading. At any fixed temperature between 0°C and 60°C, the accuracy is adjustable to ±0.05% of full scale.

Gain TC (at gain = 1)—30 PPM (typical), 56 PPM (max) per degree centigrade, 40 PPM at other gains.

Offset TC— (in % of FSR/°C)	Gain	Offset TC (typical)
	1	0.0018%
	5	0.0036%
	50	0.024%
	250	0.12%

Sample and Hold-Sample Time: 15 μs
Aperature-Hold Aperature Time: 120 ns
Input Overvoltage Protection: 30 volts
Input Impedance: 20 megohms (min.)
Conversion Speed: 50 μs (max.) at gain = 1
Common Mode Rejection Ratio: 60 dB (min.)

Physical Characteristics

Width: 30.48 cm (12.00 in.)

Length: 17.15 cm (6.75 in.)

Height: 1.78 cm (0.7 in.)

2.82 cm (1.13 in.) with iSBC Memory Expansion, MULTIMODULES, iSBX Numeric Data Processor or iSBX MULTIMODULES.

Electrical Requirements

Power Requirements

Voltage	Current	
	Maximum	Typical
+5V	5.5A	4A
+5V Aux	150 mA	100 mA
+12V	120 mA	80 mA
−12V	40 mA	30 mA

NOTES:
1. The current requirement includes one worst case (active-standby) EPROM current.
2. If +5V Aux is supplied by the iSBC 88/40A board, the total +5V current is the sum of the +5V and the +5V Aux.

Environmental Requirements

Operating Temperature: 0° to +60°C with 6 CFM min. air flow across board

Relative Humidity: to 90% without condensation

Equipment Supplied

iSBC 88/40A Measurement and Control Computer Schematic diagram

REFERENCE MANUALS

147049— SBC 88/40A Measurement and Control Computer Hardware Reference Manual (Order Separately).

Manuals may be ordered from an Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

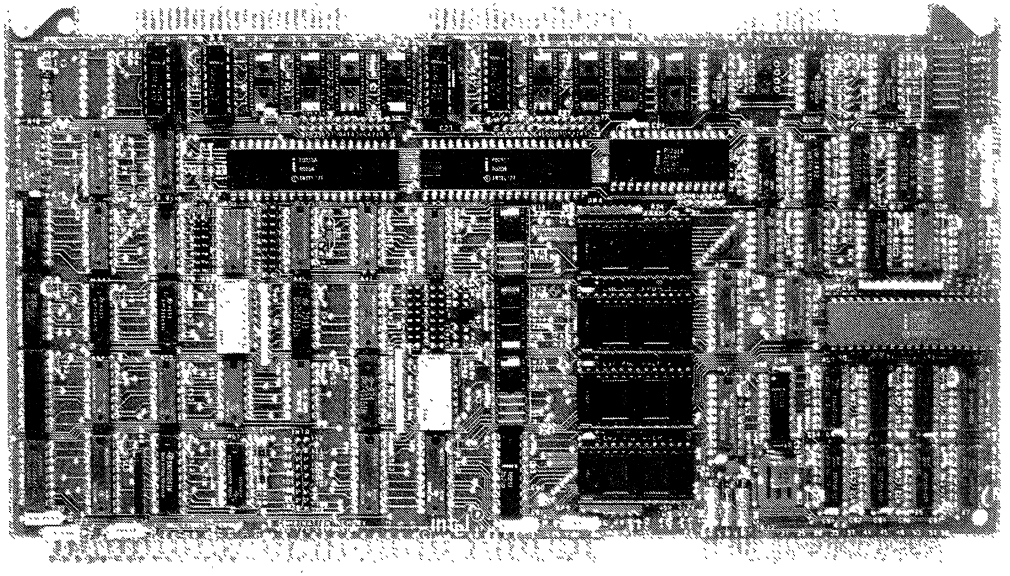
ORDERING INFORMATION

Order Code	Description
SBC8840A	Measurement and Control Computer

iSBC® 108A/116A COMBINATION MEMORY AND I/O EXPANSION BOARDS

- **8K or 16K Bytes of Read/Write Memory (iSBC 108A, iSBC 116A Boards Respectively) Sockets for up to 32K Bytes of EPROM**
- **Auxiliary Power Bus and Memory Protect Control Logic Provided for Battery Backup RAM Requirements**
- **RAM and EPROM Assignable Anywhere within a One Megabyte Address Space**
- **48 Programmable I/O Lines with Sockets for Interchangeable Line Drivers and Terminators**
- **Synchronous/Asynchronous Communications Interface with RS232C Drivers and Receivers**
- **Eight Maskable Interrupt Request Lines with a Pending Interrupt Register**
- **1 ms Interval Timer**

The iSBC 108A and iSBC 116A Combination Memory and I/O Boards are members of Intel's complete line of MULTIBUS memory and I/O expansion boards. Both boards interface directly to a host single board computer via the MULTIBUS interface to expand RAM, EPROM serial I/O and parallel I/O capacity. This mixture makes the iSBC 108A and 116A combination boards ideal for small microcomputer systems where the on-board resources of a single board computer are insufficient for incrementing the memory and I/O capacities of larger multiple board systems.



281011-1

October 1989
Order Number: 281011-001

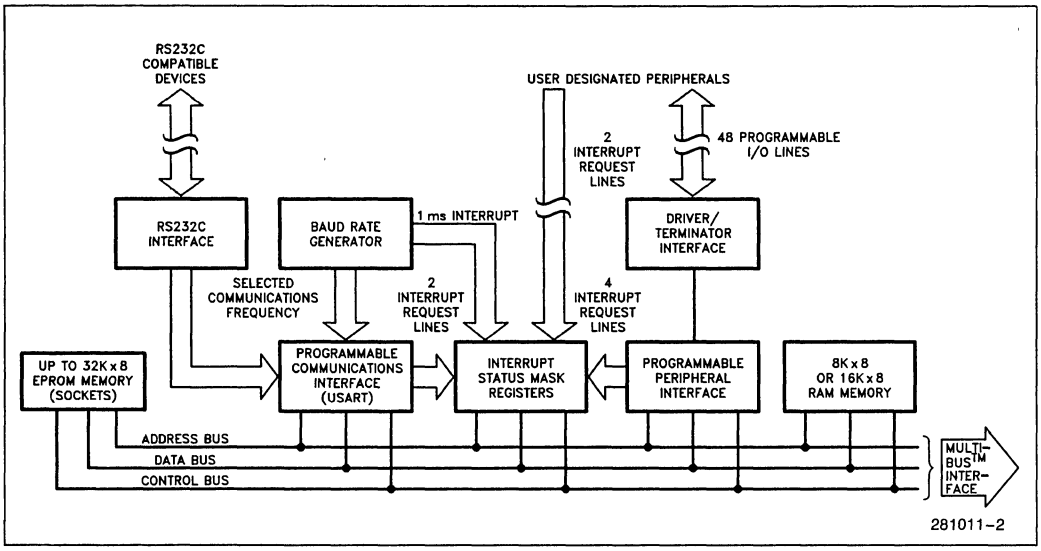


Figure 1. iSBC® 108A/116A Combination Memory and I/O Expansion Board Block Diagram

I/O Port Operation Modes

Port	Lines (qty)	Mode of Operation					
		Unidirectional				Bidirectional	Control
		Input		Output			
		Unlatched	Latched & Strobed	Latched	Latched & Strobed		
1	8	X	X	X	X	X	
2	8	X	X	X	X		
3	4	X		X			X[1]
	4	X		X			X[1]
4	8	X	X	X	X	X	
5	8	X	X	X	X		
6	4	X		X			X[2]
	4	X		X			X[2]

NOTES:
1. Part of port 3 must be used as a control port when either port 1 or port 2 are used as a latched and strobed input or a latched and strobed output or port 1 is used as a bidirectional port.
2. Part of port 6 must be used as a control port when either port 4 or port 5 are used as a latched and strobed input or a latched and strobed output or port 4 is used as a bidirectional port.

Memory Word Size

8 bits only. 16-bit single board computers may use this memory only for the storage of 8-bit data.

Memory Addressing

EPROM—Up to 4K, 8K, 16K or 32K bytes of read-only-memory may be located anywhere within a one megabyte address range. The base address must be located on a 4K byte boundary. EPROM addresses may not cross 32K byte boundaries.

RAM—8K (iSBC 108A) or 16K (iSBC 116A) bytes of RAM may be located anywhere in a one megabyte address range. The base address must be located on a 4K byte boundary. RAM addresses may not cross 32K byte boundaries.

Memory Response Time

Memory	Access (ns)	Cycle (ns)
RAM	450 Max*	580 Max*
EPROM/ROM	450 Max	635 Max

*Without refresh contention.

I/O Transfer Rate

Parallel—Read or write acknowledge time 575 ns max.

Serial—(USART)

Frequency (kHz) (Jumper Selectable)	Baud Rate (Hz)		
	Synchronous	Asynchronous (Program Selectable)	
		÷ 16	÷ 64
307.2	—	19200	4800
153.6	—	9600	2400
76.8	—	4800	1200
38.4	38400	2400	600
19.2	19200	1200	300
9.6	9600	600	150
4.8	4800	300	75
6.98	6980	—	110

Serial Communications Characteristics

Synchronous—5–8 bit characters; internal or external character synchronization; automatic sync insertion.

Asynchronous—5–8 bit characters; break characters generation; 1, 1½, or 2 stop bits; false start bit detectors.

Interrupts

Eight interrupt request lines may originate from the programmable peripheral interface (4 lines), the USART (2 lines) or user specified devices via the I/O edge connector (2 lines), or interval timer.

Interrupt Register Address

XX1 Interrupt mask register

XX0 Interrupt status register

NOTE:
XX is any two hex digits assigned by jumper selection.

Timer Interval

1.003 ms ±0.1% when 110 baud rate is selected.

1.042 ms ±0.1% for all other baud rates.

I/O Addressing

Port	1	2	3	4	5	6	8255A No. 1 Control	8255A No. 2 Control	USART Data	USART Control
Address	XX4	XX5	XX6	XX8	XX9	XXA	XX7	XXB	XXC or XXE	XXD or XXF

NOTE:
XX is any two hex digits assigned by jumper selection.

Auxiliary Power

An auxiliary power bus is provided to allow separate power to RAM for systems requiring battery backup or read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

Memory Protect

An active-low TTL compatible memory protect signal is brought out on the auxiliary connector which, when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during system power-down sequences.

Line Drivers and Terminators

I/O Drivers—The following line drivers and terminators are all compatible with the I/O driver sockets on the iSBC 108A/116A board. Ports 1 and 4 have 25 mA totem-pole drivers and 1 kΩ terminators.

Driver	Characteristic	Sink Current (mA)
7438	I,OC	48
7437	I	48
7432	NI	16
7426	I,OC	16
7409	NI,OC	16
7408	NI	16
7403	I,OC	16
7400	I	16

NOTE:
I = Inverting, NI = Non-Inverting, OC = Open Collector.

I/O Terminators—220Ω/330Ω divider or 1 kΩ pull-up user supplied.

Bus Driver

Function	Characteristic	Sink Current (mA)
Data	Tri-State	32
Commands	Tri-State	32

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.50 in. (1.27 cm)
Weight: 14 oz. (397.3 gm)

Electrical Characteristics Average DC Current

	V_{DD} = +12 ±5%	V_{CC} = +5 ±5%	V_{BB} = −5 ±5%	V_{AA} = −12 ±5%
No EPROM or Terminators	250 mA	2.9A	—	70 mA
4 2708s and 8 Terminators	520 mA	3.6A	180 mA	70 mA
4 2716s and No Terminators	250 mA	3.3A	—	70 mA
4 2732s and No Terminators	250 mA	3.5A	—	70 mA
Aux. Power RAM Accessed	175 mA	0.45A	3 mA	—
Aux. Power No RAM Access	20 mA	0.45A	3 mA	—

Environmental Characteristics

Operating Temperature: 0°C to +55°C.

Reference Manuals

9800862: iSBC 108A/116A Board Hardware Reference Manual (NOT SUPPLIED)

Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

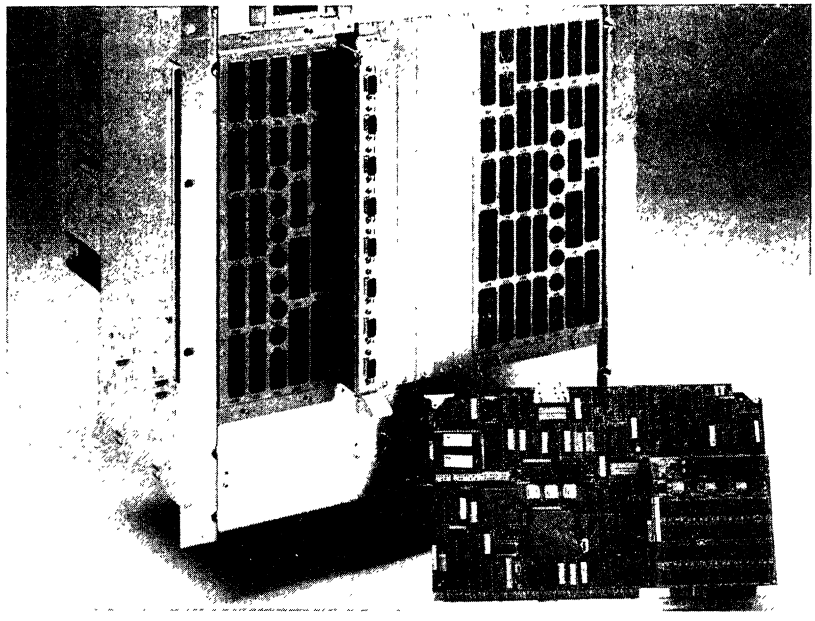
ORDERING INFORMATION

Order Code	Description
SBC108A	Combination Memory and I/O Expansion Board with 8 Kbytes RAM
SBC116A	Combination Memory and I/O Expansion Board with 16 Kbytes RAM

MULTIBUS® I System Packaging and Power Supplies

13

SYP341 CARD CAGE MODULE



A 16-SLOT MULTIBUS®I CARD CAGE MODULE FOR FLEXIBLE, EXPANDABLE SYSTEMS CONFIGURATIONS

Intel's SYP341 Card Cage is a standard module designed to provide, along with the companion SYP342 Peripheral Module, a basic platform for the integration of large capacity systems. Intel's modular packaging scheme allows for integration into standard 19 inch rack-mount cabinets or NEMA-type enclosures.

FEATURES

- 16-slot MULTIBUS I backplane with integrated priority and interrupt circuitry.
- Accepts standard 7 x 12 inch MULTIBUS I boards and up to seven 10 x 12 inch boards.
- Meets EIA, 19 inch rack standard.
- 4-layer backplane construction. Interleaved bus signal traces. Dedicated power and ground layers.
- 24-bit addressing supported on all slots.
- Extended gold pins for all P2 signals. Supports iLBX bus cables.
- Backplane generated bus clock.
- MULTIBUS reset and interrupt switches with power-on and status indicators.
- 750 watt multiple output switching power supply. Switch selectable 110/220 VAC.
- Forced air cooling. Provides 300 lfm across boards.

September, 1989
Order Number 280641-002

ENVIRONMENTALS

Ambient Temperature	
Operating	0 to 55°C
Non-Operating	−40 to 80°C
Relative Humidity	
Operating	80% at 40°C
Non-operating	95% at 55°C
Altitude	
Operating	Sea Level to 10,000 feet
Non-Operating	Sea Level to 40,000 feet

ELECTRICAL

DC Power Output	750 watt maximum
+5v	100.0 A maximum
+12v	10.0 A maximum
−12v	10.0 A maximum
AC Power Input	90-132 VAC or 180-264 VAC 47-63 Hz

REGULATIONS

Meets the following safety requirements:

US	UL478 5th Edition recognized
Canada	CSA C22.2 No. 220 certified
Europe	IEC 380 and IEC 950

Power Supply meets the following EMI/RFI requirements:

US	FCC Class B Conducted emissions
Europe	VDE Limit Class B Conducted emissions

PHYSICAL CHARACTERISTICS

Dimensions

Standard Rear Mount Power Supply

Height	488.1 mm (19.22 in)
Width	482.7 mm (19.00 in)
Depth	501.6 mm (19.75 in)
Weight	23.9 kilograms (53 lbs)

Optional Mounting:
Side Mounted Power Supply

Width	597.0 mm (23.50 in)
Depth	355.7 mm (14.00 in)

Backplane Slot Spacing

Slots 5, 13 — 1.8″
Slot 1 — 1.4″
Slots 2-4, 6-12, 14-16 — 0.8″
Slots 6-12 accommodate 10 x 12 inch boards

ORDER CODES

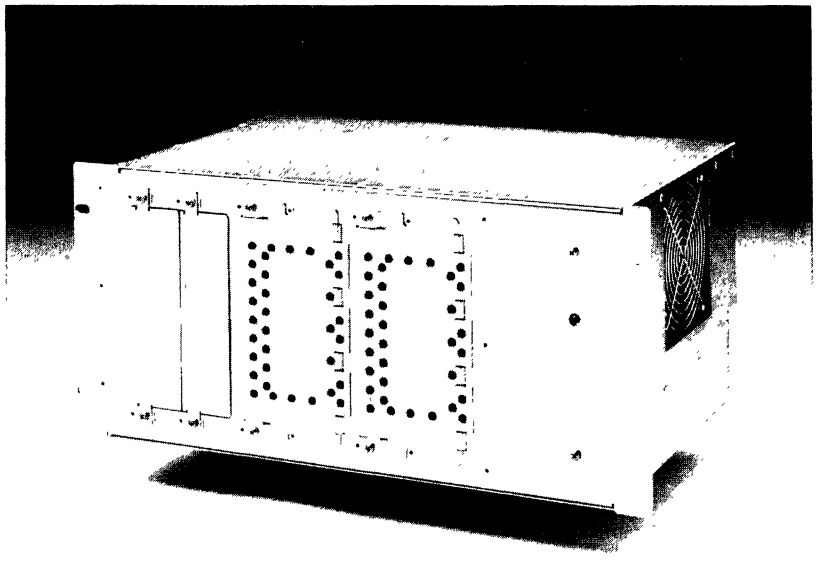
SYP341V1—Configured 110 VAC
SYP341V2—Configured 220 VAC

WORLDWIDE SERVICE AND SUPPORT

Intel provides support for Intel and non-Intel boards and peripherals as well as on-site service. Development support options include phone support, subscription service, on-site consulting, and customer training.

QUALITY AND RELIABILITY

The SYP341 is designed, tested and manufactured in accordance with Intel's industry leading quality and reliability standards.



A PERIPHERAL MODULE FOR FLEXIBLE, EXPANDABLE SYSTEM CONFIGURATIONS

Intel's SYP342 Peripheral Module is a standard module designed to provide, along with the companion SYP341 Card Cage Module, a basic platform for the integration of large capacity systems. Intel's modular packaging scheme allows for integration into standard 19-inch rack-mount cabinets or NEMA-type enclosures.

FEATURES

- Houses up to 3 full-height or 6 half-height 5¼ inch peripherals
- Meets EIA, 19-inch rack standard
- 175 Watt multiple output switching power supply
- Auto selectable 110/220 VAC
- Forced air cooling
- Selectable exposed or recessed peripheral mounting positions
- Peripheral carriers, filler panels and ESD panels provided

September, 1989
Order Number 280642-002

ENVIRONMENTALS

Ambient Temperature	
Operating	0° to 50°C
Non-Operating	−40° to 80°C
Relative Humidity	
Operating	80% at 40°C
Non-Operating	95% at 55°C
Altitude	
Operating	Sea Level to 10,000 feet
Non-Operating	Sea Level to 40,000 feet

ELECTRICAL

DC Power Output	175 watt maximum continuous (220 watt maximum peak)
+5v	15 A maximum continuous
+12v	12A maximum continuous (18 A maximum peak)
AC Power Input	90-130 VAC or 180-260 VAC 47-63 Hz

REGULATIONS

Meets the following safety requirements:

US	UL478 5th Edition recognized
Canada	CSA C22.2 No. 220 certified
Europe	IEC 380 and IEC 950

Power Supply meets the following EMI/RFI requirements:

US	FCC Class B Conducted emissions
Europe	VDE Limit Class B Conducted emissions

PHYSICAL CHARACTERISTICS

Height	220.9 mm (8.7 in)
Width	482.7 mm (19.0 in)
Depth	292.1 mm (11.5 in)
Weight	12.7 kilograms (28 lbs)

ORDER CODE

SYP342E

WORLDWIDE SERVICE AND SUPPORT

Intel provides support for Intel and non-Intel boards and peripherals as well as on-site service. Development support options include phone support, subscription service, on-site consulting, and customer training.

QUALITY AND RELIABILITY

The SYP342 is designed, tested and manufactured in accordance with Intel's industry leading quality and reliability standards.

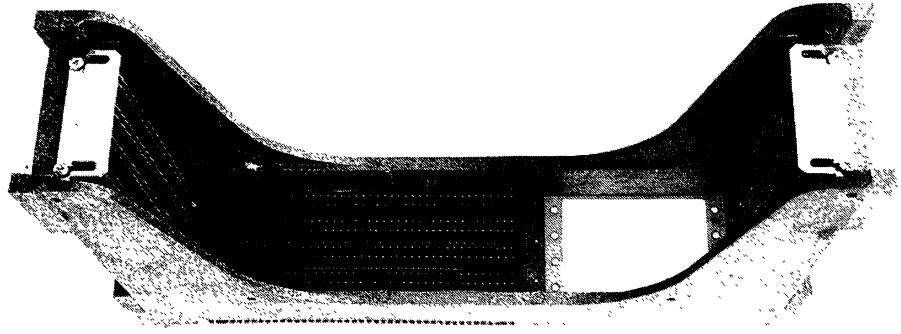
ORDERING INFORMATION

For more information or the number of your nearest sales office call 800-548-4725 (good in the U.S. and Canada).

iSBC® 604/614
MODULAR CARDCAGE ASSEMBLIES

- **Interconnects and Houses up to Four MULTIBUS® Boards per Cardcage**
- **Connectors Allow Interconnection of up to Four Cardcage Assemblies for 16 Board Systems**
- **Strong Cardcage Structure Helps Protect Installed Boards from Warping and Physical Damage**
- **Cardcage Mounting Holes Facilitate Interconnection of Units**
- **Compatible with 3.5-Inch RETMA Rack Mount Increments**
- **Interleaved Grounds on Backplane Minimize Noise and Crosstalk**
- **Up to 3 CPU Boards per System for Multiprocessing Applications**

The iSBC 604 and iSBC 614 Modular Cardcage Assemblies units provide low-cost, off-the-shelf housing for OEM products using two or more MULTIBUS boards. Each unit inerconnects and houses up to four boards. The base unit, the iSBC 604 Cardcage Assembly, contains a male backplane PC edge connector and bus signal termination circuits, plus power supply connectors. It is suitable for applications requiring a single unit, or may be interconnected with up to three iSBC 614 cardcage assemblies for a four cardcage (16 board) system. The iSBC 614 contains both male and female backplane connectors, and may be interconnected with iSBC 604/614 units. Both units are identical, with the exception of the bus signal terminator feature. A single unit may be packaged in a 3.5 inch RETMA rack enclosure, and two interconnected units may be packaged in a 7 inch enclosure. The units are mountable in any of three planes.



280205-1

November 1989
Order Number: 280205-001

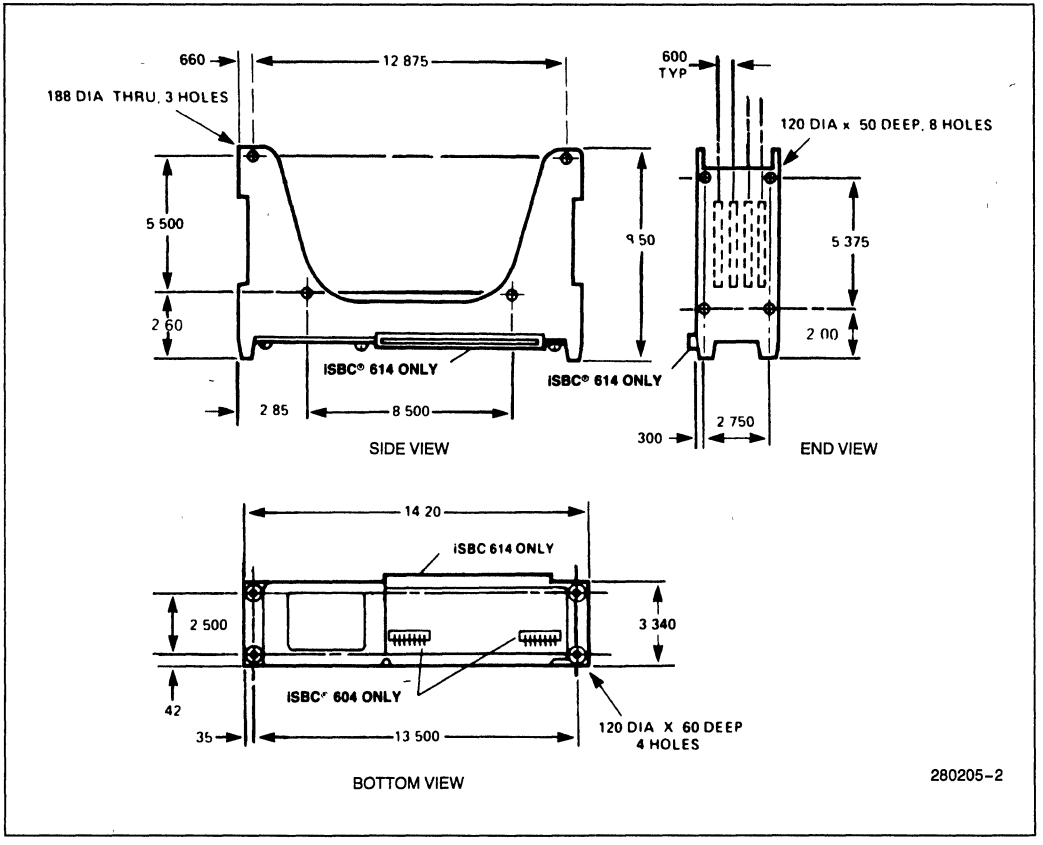


Figure 1. iSBC® 604/614 Cardcage Assembly Dimensions

SPECIFICATIONS

Backplane

Bus Lines—All MULTIBUS system bus address, data, and command bus lines are bussed to all four connectors on the printed circuit backplane

Power Connectors—G for ground, +5, −5, +12V, −12V, and −10V power supply lines

ISBC 604—Bus signal terminators, backplane male PC edge connector only, and power supply headers

ISBC 614—Backplane male and female connectors and power supply headers

Mating Power Connectors

AMP	Connector	87159-7
	Pin	87023-1
	Polarizing Key	87116-2
Molex	Connector	09-50-7071
	Pin	08-50-0106
	Polarizing Key	15-04-0219

NOTE:
1. Pins from a given vendor may only be used with connectors from the same vendor.

ORDERING INFORMATION

Part Number	Description
SBC 604	Modular Cardcage Assembly (Base Unit)

Bus Arbitration: Serial; up to 3 CPU masters

Equipment Supplied: iSBC 604 or iSBC 614 Cardcage Schematic

Physical Dimensions

Height: 8.5 in. (21.59 cm)

Width: 14.2 in. (36.07 cm)

Depth: 3.34 in. (8.48 cm)

Weight: 35 oz. (992.23 gm)

Card Slot Spacing: 0.6 in.

Environmental Characteristics

Operating Temperature: 0°C to 55°C

Reference Manual

9800708—iSBC 604/614 Cardcage Hardware Reference Manual (ORDER SEPARATELY)

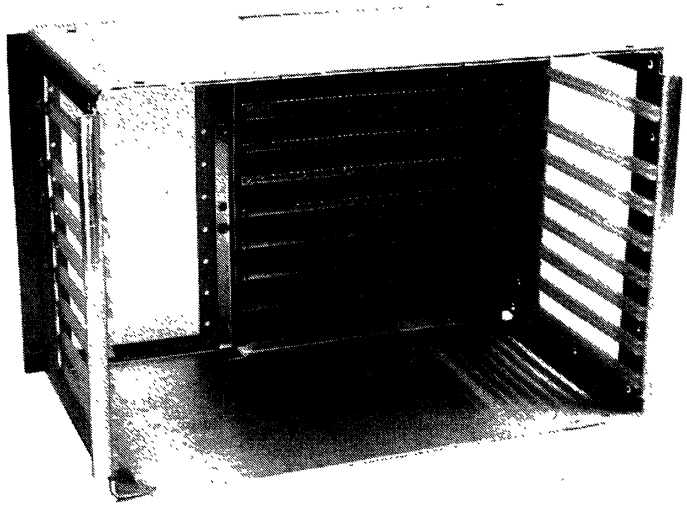
Part Number	Description
SBC 614	Modular Cardcage Assembly (Expansion Unit)

iSBC® 608/618 CARDCAGES

- **Houses Eight MULTIBUS® iSBC® Boards in an Aluminum Package**
- **Board-to-Board Clearance for iSBC® MULTIMODULE™ Boards on All Slots**
- **Board-to-Board Clearance for iSBX™ MULTIMODULE™ Boards on Two Slots**
- **Parallel Priority Circuitry for up to Eight Multimaster iSBC® Boards**
- **Enhanced Bus Noise Immunity for High Speed Systems**
- **Plug on iSBC 618 Unit for up to Sixteen Board Systems**
- **NEMA-Type Backwall or 19-Inch Rack Mount Hardware Included**
- **Signal Line Termination Circuitry on iSBC® 608 Cardcage**

Intel's iSBC 608/618 Cardcages are matched to the latest generation of iSBC/iSBX boards which mount in the MULTIBUS system bus. These products provide several features which make them the industry's leading price/performance cardcage product. MULTIMODULE board clearance, parallel priority circuitry, enhanced backplane noise immunity, and precision fit card guides are a few of the distinctions which make this the industry's better product.

The iSBC 608 Cardcage is the base unit, housing up to eight iSBC boards and their MULTIMODULE boards. Additionally, this base unit includes mounting hardware and fan mounting bracketry. The iSBC 618 is the expansion unit, providing eight additional iSBC board slots to the iSBC 608 Cardcage for a total of sixteen board slots which can be NEMA-type backwall or 19-inch rack mounted. This is accomplished with the mounting hardware of the iSBC 608 Cardcage. The iSBC 618 expansion unit also includes fan mounting bracketry.



210373-1

November 1986
Order Number: 210373-001

FUNCTIONAL DESCRIPTION

Mechanical Aspects

The iSBC 608/618 Cardcages provide housing and a MULTIBUS system bus for up to sixteen single board computers and their MULTIMODULE boards. The iSBC 608 unit and iSBC 618 unit offer board-to-board clearance (0.8 inches or greater) on all eight slots for iSBC MULTIMODULE boards. Two slots provide clearance (1.2 inches or greater) for iSBX MULTIMODULE boards as shown in Figure 1. Each cardcage includes precision fitted nylon cardguides for secure board fit and accurate MULTIBUS board pin alignment. Fan mounting bracketry is also included with each cardcage. This bracketry allows the mounting of several industry standard fans. The iSBC 608 Cardcage base unit includes aluminum mounting hardware for NEMA-type backwall mounting, or anchoring a sixteen slot iSBC 608/618 combination in a standard 19-inch rack.

Electrical Aspects

The iSBC 608/618 Cardcages implement a parallel priority resolution scheme by using plug-in jumper connections. There are six different priority schemes allowed, each requiring a different jumper configuration. In systems where an iSBC 618 Cardcage is attached to the base unit, the base unit will have lower priority overall. That is, master boards in the iSBC 608 base unit bay gain control of the MULTIBUS lines only when no boards in the iSBC 618 expansion unit are asserting the bus request (BREQ/) signal.

Noise-minimizing ground traces are strategically interleaved between signal and address lines on these backplanes. This provides the enhanced noise immunity and minimized signal-to-signal coupling which is important in high speed, high board count microcomputer systems.

The iSBC 608/618 Cardcages provide power connector lug bolts for +5 VDC and ground. The lug bolts, compared to other power connection methods, help transfer higher amounts of current. Other voltages (±12 VDC, −5 VDC) are connected via a mating power connector plug as shown in Figure 2.

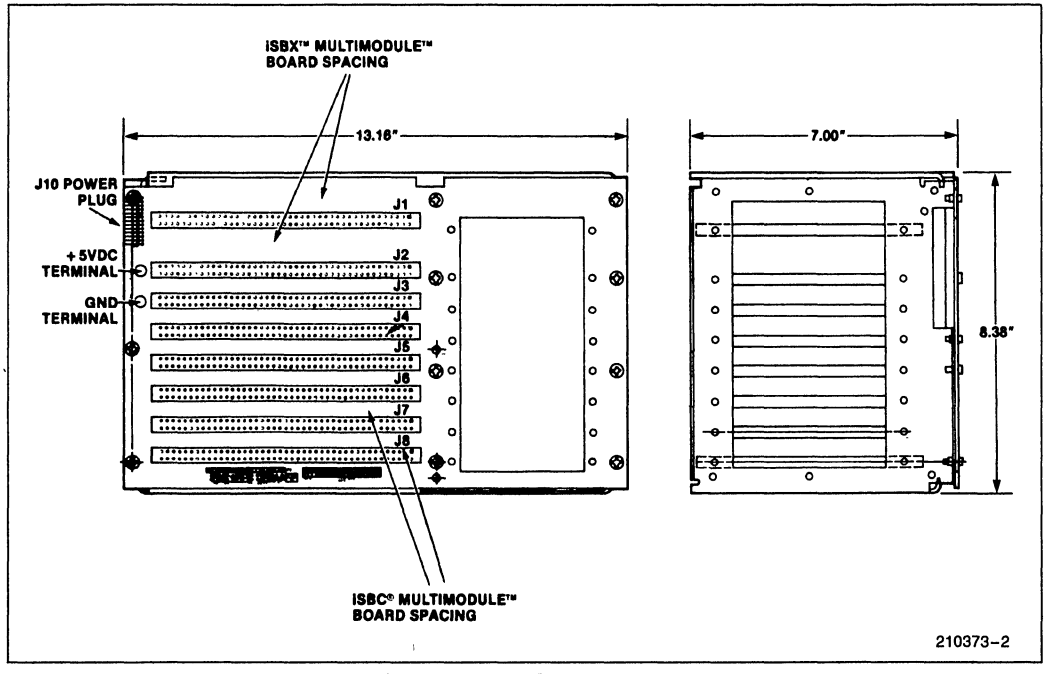


Figure 1. iSBC® 608/618 Cardcages Dimensions

SPECIFICATIONS

Bus Lines

All MULTIBUS (IEEE 796) system bus address and command lines are bussed to each of the eight MULTIBUS connectors on the backplane. Ground traces are interleaved among these signal lines and bussed to the backplane edge connector for interconnection of the iSBC 608 and iSBC 618 backplane.

Power Connectors

Ground (0V), +5V, −5V, +12V, −12V power supply header stakes and power lug bolts are provided on the iSBC 608/618 Cardcages as shown in Figure 2.

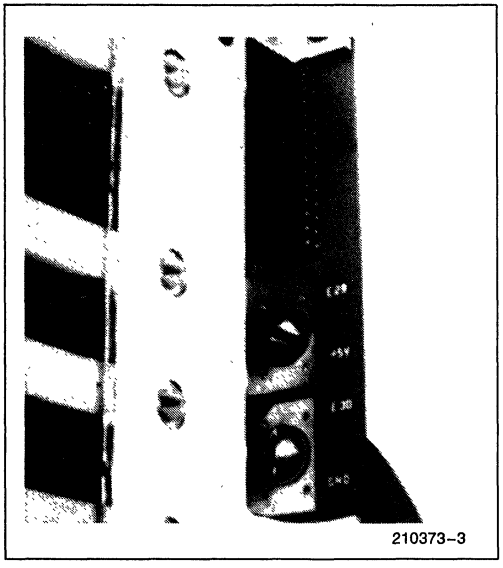


Figure 2. Power Header Stakes and Lugs

Environmental Characteristics

Operating Temperature:	0°C to 55°C
Storage Temperature:	−40°C to +85°C
Humidity:	50% to 95% non-condensing at 25°C to 40°C.
Vibration and Shock:	2G max. through 50 Hz

Physical Characteristics

SLOT-TO-SLOT DIMENSIONS (See Figure 1)

Top-J1:	1.200 in. (to center)
J1-J2:	1.300 in. (center to center)
J8-Bottom:	0.700 in. (to center)
All Others:	0.800 (center to center)

Physical Dimensions

Height:	8.38 in. (21.29 cm)
Length:	13.16 in. (33.43 cm)
Width:	7.50 in. (19.05 cm)
Weight:	3.50 lbs (1.59 kg)
Shipping Weight:	5.75 lbs (2.61 kg)

Equipment Supplied

iSBC® 608 BASE UNIT

Eight Slots:	Two at greater than 1.2 inches; six at 0.8 inches
Male Backplane Connector:	For expansion with iSBC 618 cardcage
Parallel Priority Circuitry:	Eight slots are configurable via the use of jumper stakes. Six priority schemes allowed
Construction Materials:	Aluminum card housing
	Nylon card guides
	Power connector header stakes and lug bolts

Accessories

iSBC® 618 EXPANSION UNIT

Eight-Slots:	Two at greater than 1.2 inches; six at 0.8 inches
Female Backplane Connector:	For expansion to iSBC 608 base unit
Parallel Priority Circuitry:	Eight slots are configurable via the use of jumper stakes. Six priority schemes allowed.
Construction Materials:	Aluminum card housing
	Nylon card guides
	Power connector header stakes and lug bolts
	Fan Mounting Hardware
	Schematic

User-Supplied Equipment

MATING POWER CONNECTORS

Vendor	Part Number
3M	3399-6026
Ansley	609-2600M
Berg	65485-009

MOUNTABLE FANS

Vendor	Part Number
Rotron	SU2A1-028267
Torin	TA300-A30473-10
Pamotor	8506D

iSBC® 661
SYSTEM CHASSIS

- **Eight-Slot MULTIBUS® Chassis with Parallel Priority Circuitry**
- **UL, FCC and CSA Approved for Data Processing Equipment**
- **230 Watt Power Supply with Power Fail Warning**
- **Designed for Slide Rack Mounting or Table-Top Use**
- **Extra-Wide Cardcage Slot Spacing for iSBX™ MULTIMODULE™ Board Clearance**
- **Configurable for Front or Rear Access to MULTIBUS® Circuit Boards**
- **Five Connector Ports for I/O Cabling**
- **Operational from 47 Hz to 63 Hz, 100/120/220/240 VAC ±10%**

The iSBC 661 System Chassis is an advanced MULTIBUS (IEEE) 796 chassis which incorporates unique usability and service features not found on competitive products. This chassis is designed or rack-mount or table-top applications and reliably operates up to an ambient temperature of 50°C. Additionally, this sytem chassis is certified by UL, CSA and FCC for data processing equipment.

An application requiring multiprocessing will find this eight-slot MULTIBUS chassis particularly well suited to its needs. Parallel priority bus arbitration circuiry has been integrated into the backplane. This permits a bus master to reside in each slot. Extra-wide inter-slot spacing on the cardcage allows the use of plug-on MULTIMODULE boards without blocking adjacent slots. For this reason, the iSBC 661 System Chassis provides the slot-functionality of most 16-slot chassis. Standard logic recognizes a system AC power failure and generates a TTL signal for use in powerdown control. Additionally, current limiting and over-voltage protection are provided at all outputs.

210866-1

October 1986
Order Number: 210866-002

FUNCTIONAL DESCRIPTION

Mechanical Features

The iSBC 661 System Chassis houses, cools, powers, and interconnects up to eight iSBC single board computers and their MULTIMODULE boards for the MULTIBUS System Bus. Based on Intel's iSBC 608 Cardcage, the chassis provides 0.8 inches of board center-to-center clearance on six slots, and 1.2 inches or more of center-to-center clearance on two slots. This permits the users of standard MULTIMODULE boards and custom wire-wrap boards to plug into the MULTIBUS System Bus without blocking adjacent slots. All slots provide enough clearance for iSBC MULTIMODULE boards, and two slots can accommodate iSBX MULTIMODULE boards.

High-technology MULTIBUS applications requiring rack-mount, or laboratory table-top use will find the iSBC 661 System Chassis ideal. Standard 19″ slide-rack mounting is possible with user-provided slides attached to the side panels. Slide mounting holes are provided in the chassis for the slide-rails listed under User Supplied Options. Rubber feet are included on the chassis for convenient table-top use.

The chassis is constructed of burnished aluminum which has been coated with corrosion-resistant chromate. It contains a system control module which presents the front panel control switches to the user, and holds the I/O cabling bulkhead to the rear. The chassis has the unique feature of being configurable for either front or rear access to MULTIBUS circuit boards.

This is accomplished by a simple procedure involving removal of the system control module, reversing it end-for-end, and re-securing it to the chassis. The system chassis is shipped in a configuration such that the MULTIBUS boards are installed from the front.

Electrical Features

The iSBC 661 System Chassis is powered by the iSBC 640 power supply. This is a standard Intel power supply which has been adopted by several MULTIBUS vendors throughout the industry. It sup-

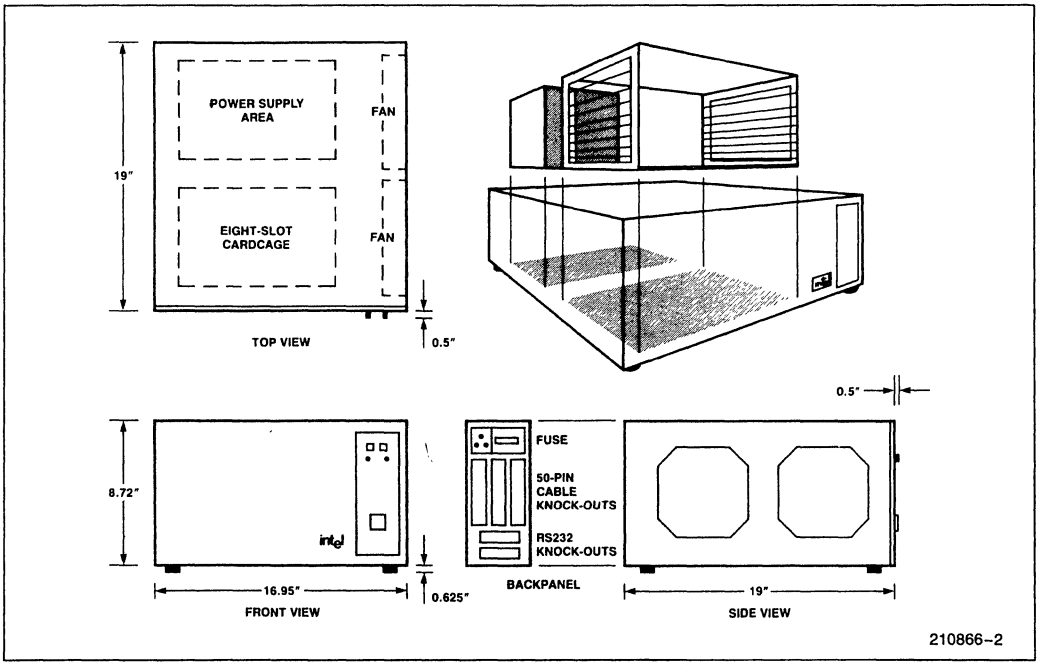


Figure 1. iSBC® 661 System Chassis Dimensions

plies 230 watts of power, power fail warning, and remote sensing of +5 volts. Its electrical and operational parameters are listed under Specifications.

The cardcage of the iSBC 661 System Chassis implements a user-changeable parallel priority bus arbitration scheme by using plug-in jumper connections. Six different priority schemes are allowed, each scheme fixing the priority to the eight MULTIBUS board slots. Bus contention among eight busmasters in a multiprocessing environment can be managed using this approach.

Noise minimizing ground traces are strategically interleaved between signal and address lines on the system bus. This provides the enhanced noise immunity and minimized signal-to-signal coupling which is particularly important in high speed, high board count microcomputer systems.

SPECIFICATIONS

Electrical Parameters

OUTPUT POWER

Table 1. Output Power Levels iSBC® 661-1

Voltage	Output Current (max.)	Current Limits (amps)	Over-Voltage Protection
+12V	4.5A	4.7–6.8	15V ±1V
+5V	30.0A	31.5–45.0	6.2V ±0.4V
−5V	1.75A	1.8–3.2	−6.2V ±0.4V
−12V	1.75A	1.8–3.2	−15V ±1V

Operational Parameters

Input AC Voltage—100/120/220/240 VAC ±10% (User selects via external switch), 47–63 Hz

Power-Fail Indication and Hold-Up Time (triggered at 90% of VAC in)—TTL O.C. High 3 msec. (min.)

Output Ripple and Noise—1% Peak-to-Peak output nominal (DC to 0.5 MHz)

Operational Temperature—0°C to 50°C

Storage Temperature—−40°C to 70°C

Operational Humidity—10% to 85% relative, noncondensing

Remote Sensing—Provided for +5 VCD

Output Transient Response—50 μs or less for ±50% load change

Physical Characteristics

Width: 16.95 inches (43.05 cm)

Height: 8.72 inches (22.2 cm)

Depth: 19.00 inches (48.3 cm)

Weight: 41 pounds (21 kg)

Shipping Weight (approx.): 50 pounds (25 Kg)

Equipment Supplied

iSBC® 661-1—Eight-slot MULTIBUS system chassis with parallel priority arbitration circuitry and 230 watt linear power supply

Reference Manual

(Not included: order separately)

145340-001—iSBC 661 System Chassis Hardware Reference Manual

User Supplied Options

Compatible Rack-Mount Slides—Chassis Trak, Inc., P. O. Box 39100, Indianapolis, IN 46239; Part No. C300 S 122

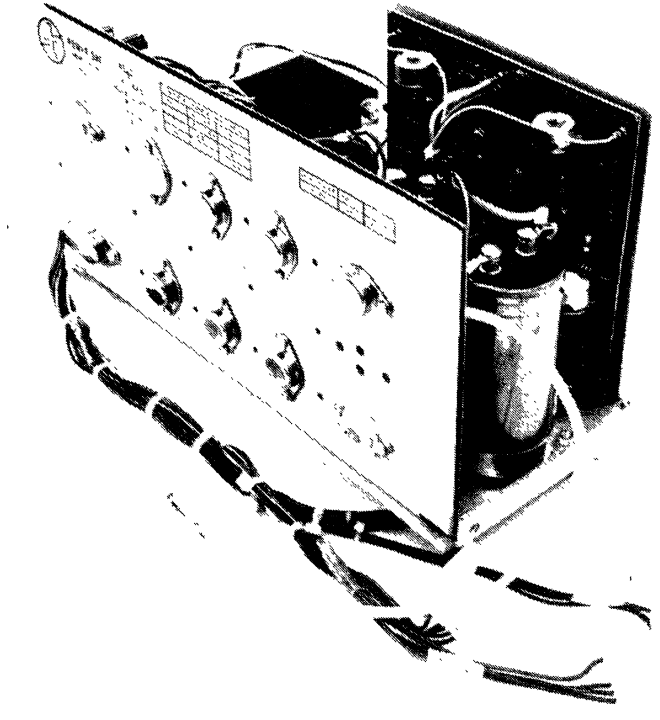
ORDERING INFORMATION

Part Number	Description
SBC 6611	Eight-slot MULTIBUS system chassis with parallel priority arbitration circuitry and 230 watt Linear Power Supply

iSBC® 640
POWER SUPPLY

- ±5V and ±12V Output Voltage
- Sufficient Power for 8–12 MULTIBUS® Computer, Memory, and Peripheral Boards
- Current Limiting and Overvoltage Protection on All Outputs
- UL Listed and CSA Certified
- "AC Low" Power Failure TTL Logic Level Output Provided for System Power-Down Control
- DC Power Cables and Connectors Mate Directly to iSBC 604/614 and iSBC 608/618 Modular Cardcage/Backplane Assemblies
- 100, 120, 220, and 240V AC Operation,
- 50 Hz or 60 Hz Input

The iSBC 640 Power Supply provides low cost, off-the-shelf, single chassis power generation for OEM and industrial system products using Intel single board computers. The iSBC 640 supply provides regulated DC output power at +12V, +5V, and −5V and −12V levels. The current capabilities of each of these output levels has been chosen to provide power over a 0°C to +55°C temperature range for one fully loaded Intel single board computer, plus residual capability for most combinations of up to eleven iSBC memory, I/O, or combination expansion boards. Current limiting and overvoltage protection is provided on all outputs. Access for AC input is provided via a standard 4-pin keyed connector. DC output power levels are provided on cables with keyed connectors directly compatible with the iSBC 604/614 and iSBC 608/618 Modular Backplane/Cardcage assemblies. The iSBC 640 supply includes logic whose purpose is to sense system AC power failure and generate a TTL signal for clean system power-down control.



281017-1

October 1989
Order Number: 281017-001

SPECIFICATIONS

Electrical Characteristics

Input Power
Frequency: 50 Hz ±5%, 60 Hz ±5%
Voltage: 100/120/220/240 VAC ±10%
via user configured wiring options

Output Power

Nominal Voltage	Current (Amps)(Max)	Current Limit Range (Amps)	Short Circuit (Amps)(Max)	Overvoltage Protection
+12V	4.5A	4.7–6.8	2.3	15V ±1V
+5V	30A	31.5–45.0	15.0	6.2V ±0.4V
−5V	1.75A	1.8–3.2	0.9	−6.2V ±0.4V
−12V	1.75A	1.8–3.2	0.9	−15V ±1V

Combined Line/Load Regulation—±1% at ±10% static line change and ±50% static load change, measured at the output connector (±0.2% measured at the power supply under the same conditions).

Remote Sensing—Provided for +5 V_{DC} output line regulation.

Output Ripple and Noise—10 mV peak-to-peak maximum (DC to 500 kHz)

Output Transient Response—Less than 50 μs for ±50% load change.

Output Transient Deviation—Less than ±10% of initial voltage for ±50% load change.

Power Failure Indication (AC Low)—A TTL open collector high signal is provided when the input voltage drops below 90% of its nominal value. DC voltages will remain within 5% of their nominal values for 3.0 ms (minimum, 7.5 ms typical) after AC Low goes true.

The "AC Low" signal will reset to a TTL low level when the AC input voltage is restored and after all output voltages are within specified regulation.

The "AC Low" threshold is adjustable for optimum powerdown performance at other input combinations (i.e., 100 VAC, 220 VAC, 50 Hz).

Mating Connectors(1)

AC Input

Housing	Molex	03-09-2042 or equivalent
Pin	Molex	02-09-2118 or equivalent (18 to 22 gauge wire)

DC Output(2)

Housing	Molex	26-03-3071
	Amp	3-87025-3
Pins	Molex	08-50-0187 or 08-50-0189
	Amp	87023-1
Key	Molex	15-04-9209
	Amp	87116-2

Compatible with Molex 09-66-1071 Header

NOTES:
1. Pins from given vendor may only be used with connectors from the same vendor.
2. iSBC 640 DC output connectors are directly compatible with input power connectors on iSBC 604/614 and iSBC 608/618 Modular Cardcage/Backplane assemblies. Four connectors are provided.

Physical Characteristics

Height: 6.66″ max. (16.92 cm)
Width: 8.19″ max. (20.80 cm)
Depth: 12.65″ max. (32.12 cm)
Weight: 30 lbs. max. (13.63 kg)

Environmental Characteristics

Temperature: 0°C to 55°C with 55 CFM moving air
Non-Operating: 40°C to +85°C

Equipment Supplied

iSBC 640 Power Supply with AC and DC cables with keyed connectors.

Reference Manuals

9800803— iSBC 640 Power Supply Hardware Reference Manual (order separately)

Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

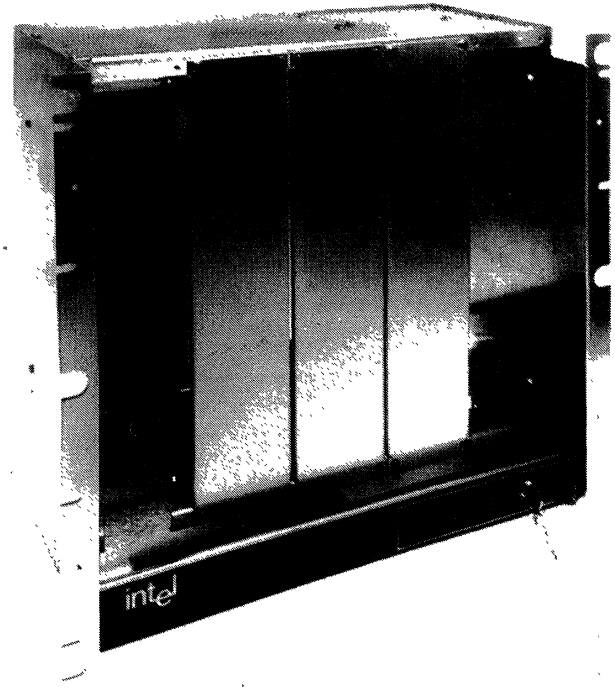
ORDERING INFORMATION

Part Number	Description
SBC640	Power Supply

iCS™ 80
INDUSTRIAL CHASSIS
KIT 635, KIT 640

- **Available with iSBC® 640 Power Supply**
- **Accommodates from 1 to 3 iSBC® 604/ 614 Cardcage Assemblies for 4–12 MULTIBUS® Board Capacity**
- **Vertical Board Orientation and Four Fans for High Efficiency Cooling**
- **Front Access to iSBC® Boards, Power Supply, and Signal Conditioning Panels**
- **19″ Wide RETMA Rack Mounting or NEMA Type Backwall Mounting Brackets**
- **UL and CSA Approved**
- **Multi-Voltage Operation**
- **Lockable Service Panel**
- **Recessed Mounting Space for Signal Conditioning/Wire Termination Panels**

The iCS 80 Industrial Chassis provides industrially oriented mounting space for Intel single board computer (iSBC) products, associated iSBC power supplies, and related analog and digital conditioning/termination panels. The base unit provides a 4-slot MULTIBUS backplane (iSBC 604) with expansion space and cabling to expand to 12 MULTIBUS backplane slots by adding additional 4-slot iSBC 614s as needed (up to two). Full MULTIBUS compatibility in the iCS 80 chassis allows configuration of multiple single board computers to share system tasks through communication over the bus (through multimaster bus arbitration built on the multiple iSBC processors).



281016–1

October 1989
Order Number: 281016-001

FUNCTIONAL DESCRIPTION

iCS™ 80 Kit 640

This chassis uses the higher power iSBC 640 power supply, and is designed to power higher board count systems. By installing one or two additional iSBC 614 cardcages, this chassis will accommodate up to 8 or 12 MULTIBUS boards. Signal conditioning panels may attach directly in the iCS 80.

Engineered for Industrial Applications

The MULTIBUS slots are mounted vertically to improve convection cooling and the top, bottom and sides are engineering to allow maximum air flow over the boards. Four fans are provided to increase air flow, allowing users to eliminate or minimize the need for supplementary fans or air conditioning.

Power Supply Flexibility

The power supplies are mounted on slide in/out mounting rails, and quick disconnect cabling and connectors are provided for rapid service replacement. An A.C. wiring barrier strip allows simple wiring connections for integration into larger systems (see Figure 4).

Industrial Rack Mounting

The chassis mounts directly into 19″ standard width RETMA (Radio-Electronics-Television Manufacturers Association) customer provided rack. Alternately, mounting brackets and power cabling access are provided for mounting directly on a backwall, such as the backwall panel of a NEMA-type (National Electrical Manufacturers Association), front-access-only cabinet.

Front Access Serviceability

To simplify serviceability, front access is provided for all iSBC boards, the power supply, operation indicator lights, interrupt and reset buttons, and the A.C. power fuse.

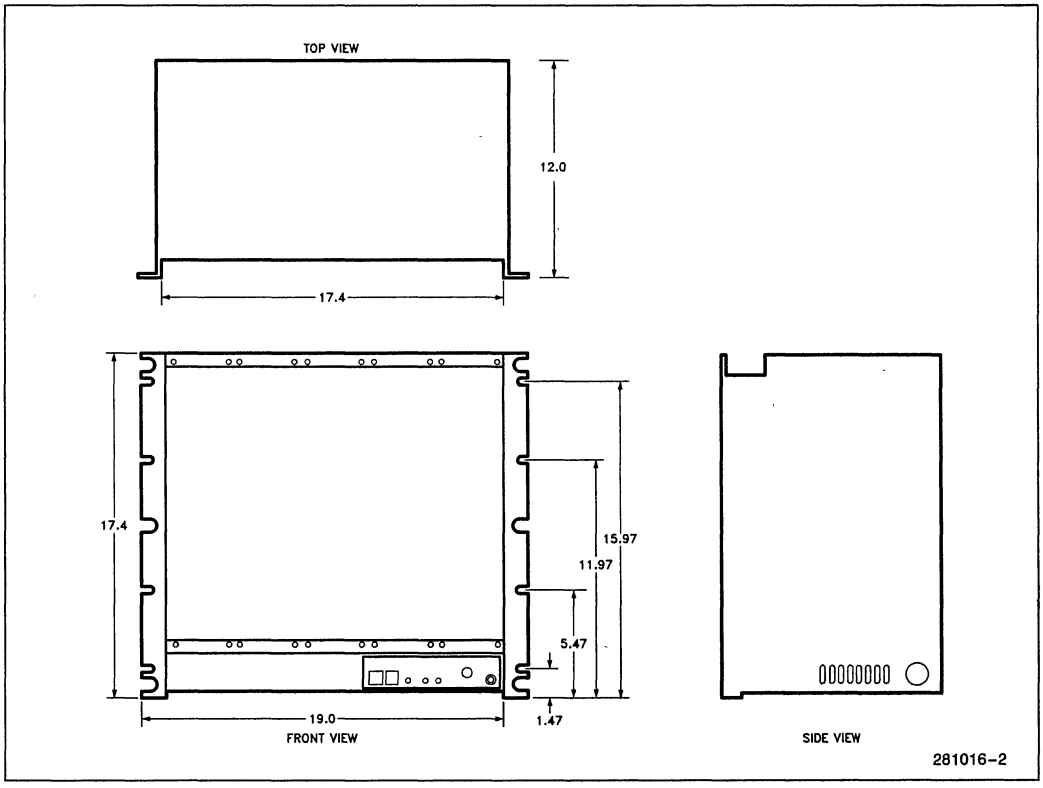


Figure 1. iCS™ 80 Chassis Dimensions

Lockable Service Panel

To assist in development, checkout and service, two pushbuttons are provided. The RESET button pulls low the initialize line (INIT) on the MULTIBUS backplane. The INTERRUPT button pulls low one interrupt line on the MULTIBUS backplane (INT1). Logic within the iCS 80 ensures that these buttons function with all versions of Intel single board computers. From the front of the iCS 80 chassis, without a CRT or other panel, an operator or service person can reset or interrupt on-going iCS 80 system operations to get attention, signal an alarm, or start a self-test operation.

A front panel key provides three positions: OFF (A.C. power off and key removable), ON (A.C. power on, pushbuttons enabled, key unremovable) and LOCK (A.C. power on, pushbuttons disabled, key removable).

Three indicator light emitting diodes record basic chassis status. POWER ON (GREEN); RUN (GREEN); and HALT (RED); the RESET or INTERRUPT buttons will remove the HALT state.

U.L. Approved

The iCS 80 chassis has received full Underwriters Laboratory approval (F.6 #E70842) as a U.L. listed component under the Underwriters Laboratories Safety Standard for Process Control Equipment, UL1092. When installed as described in the iCS 80 Hardware Reference Manual, the iCS 80 chassis provides adequate protection against shock, fire and casualty hazards, and should comply with most local and regional requirements for installation in ordinary locations. In addition, the iCS 80 chassis was designed to comply with the UL requirements for Data Processing Equipment, UL478. The iCS 80 has also been approved by the Canadian Standards Association under CSA category C22.2 No. 142, the Canadian Standard for Safety for Process Control Equipment and C22.2 No. 154 for Data Processing Equipment.

Mounting Space for Signal Conditioning/Wire Terminations

The cardcages and power supplies in the iCS 80 chassis are recessed behind the front edge of the rack mounting ears to provide mounting space for the iCS 9XX series signal conditioning/termination panels and field wiring. For smaller systems with only one or two iSBC 604/614 cardcages (4 to 8 slots), up to two iCS 910, iCS 920, or iCS 930 signal conditioning/termination panels can be mounted vertically over the area where the second or third cardcage would mount (see Figure 2). The benefit of this design is a completely self-contained industrial chassis with iSBC cards, power supply, signal conditioning and field wiring terminations, all in one enclosure.

SPECIFICATIONS

Capacity

Four slots for MULTIBUS compatible single board computers, memory, I/O or other expansion boards
Expandable to 12 slots using two iSBC 614 cardcages
(Order Separately)

Front Panel Controls

Pushbuttons

RESET: Connected to Initialize/ on MULTIBUS backplane

INTERRUPT: Connected to Interrupt 1/ line on MULTIBUS backplane.

Panel Indicator Lights (LEDs)

POWER ON (green): +5V power exists on the MULTIBUS backplane

RUN (green): CPU is executing an instruction. Light goes out if CPU is in WAIT or HALT state

HALT (red): CPU has executed a HALT instruction

Keylock

OFF: A.C. power off, key removable

ON: A.C. power on, pushbuttons enabled, key unremovable

LOCK: A.C. power on, pushbottons disabled, key removable

Fuse—A.C. power (6A)

Equipment Supplied

iCS 80 industrial chassis, three fans for cardcages, one fan for power supply, 4-slot cardcage with MULTIBUS backplane, control panel with switches, indicators, keylock, power distribution barrier strip, A.C. power fuse, line filter, 115V power cable, and logic for interrupt and reset buttons. An installation package is also provided, including a NEMA cabinet

mounting kit, power supply extension cables, and RETMA cabinet mounting screws, 100/120/220/240 VAC operation.

Software

See the RMX/80 Real-time Multi-tasking Executive specifications for industrial related applications. In addition, system monitors for most of the Intel single board computers are available in the INSITE (Intel's Software Index and Technology Exchange) User's Program Library.

Physical Characteristics

Height: 39.3 cm (15.7")

Width: 48.5 cm (19.0") at front panel
43.5 cm (17.4") behind front panel

Depth: 30.0 cm (12.0") with all protrusions

Weight: 16.8 kg (37.0 lb) without power supplies

Environmental Characteristics

(Ambient at iCS 80 air intake, bottom of chassis)

Temperature (Ambient)

Operating: 0°C to 50°C (32°F to 122°F)

Non-Operating: −40°C to +85°C

Humidity: Up to 90% relative, noncondensing at 40°C

Electrical Characteristics

The iCS 80 chassis provides mounting space for the iSBC 640 power supply. Unless otherwise stated, electrical specifications apply to both power supplies when installed by user in iCS 80 chassis.

Input Power

Frequency: 47 Hz to 63 Hz Voltage (Nominal)
Voltage (Single Phase, Jumper Selectable)
iCS 80 Kit 640: 100, 120, 220, 240 VAC (±10%)

Current (Including Fans)	with iSBC 640	Input Voltage
	5.6A max	103 VAX
	2.8A max	206 VAX
Power, Max:	580W	

Output Power

Voltage	Output Current (Max)	Overvoltage Protection
	iSBC 640	iSBC 640
+12V	4.5A	+14V to +16V
+5V	30.0A	+5.8V to +6.6V
−5V	1.75A	−5.8V to −6.6V
−12V	1.75A	−14V to −16V

Combined Line/Load Regulation—±1% at ±10% static line change and ±50% static load change, measured at the output connector (±0.2% measured at the power supply under the same conditions).

Remote Sensing—Provided for +5 V_{DC} output line regulation.

Output Ripple and Noise—10 mV (iSBC 640 supply) peak-to-peak, maximum (D.C. to 500 kHz)

Output Transient Response—Less than 50 μs for ±50% load change

Maximum Watts Dissipation (load plus losses)—500W (iSBC 640 supply)

Installation

Complete instructions for installation are contained in the iCS 80 Site Planning and Installation Guide, including RETMA and NEMA cabinet mounting, and field signal, ground wiring and cooling suggestions.

Warranty

The iCS 80 Industrial Chassis is warranted to be free from defects in materials and workmansip under normal use and service for a period of 90 days from date of shipment.

ORDERING INFORMATION

Part Number	Description
iCS 80 Kit 640	iCS 80 system consisting of:
	iCS 80 Industrial Chassis
	iSBC 640 Power Supply

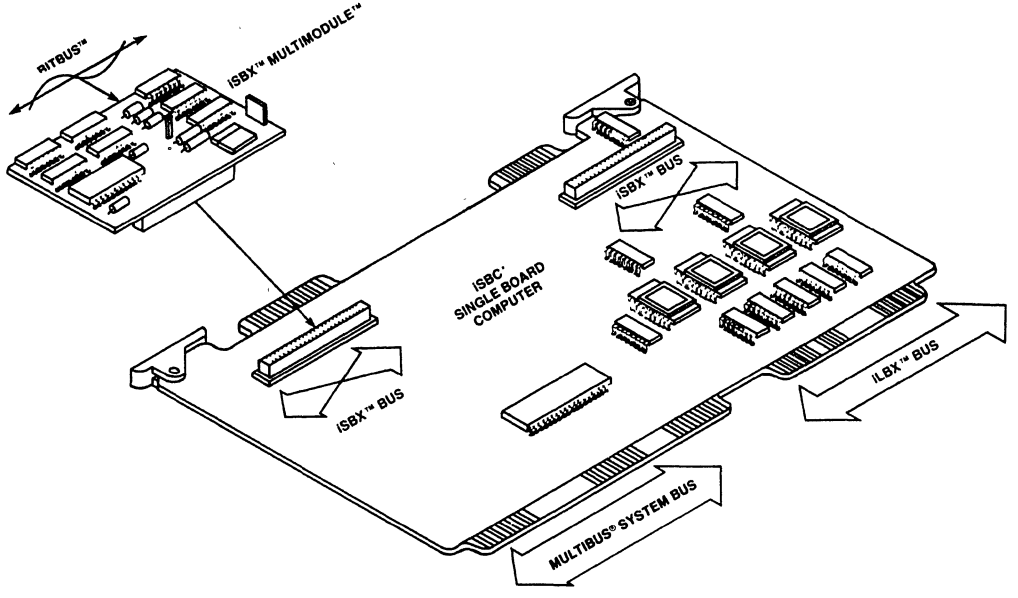
MULTIBUS® I Architecture

14

MULTIBUS® SYSTEM BUS

- **IEEE 79688 Industry Standard System Bus**
- **Supports Multiple Processor Systems with Multi-Master Bus Structure**
- **8-Bit, 16-Bit, and 32-Bit Devices Share the Same MULTIBUS® System Resources**
- **Foundation of Intel's Total System Architecture: MULTIBUS®, iLBX™, BITBUS™ and iSBX™ Buses**
- **16 Mbyte Addressing Capability**
- **Bus Bandwidth of Up to 10 Megabytes Per Second**
- **Supported by a Complete Family of Single Board Computers, Memory, Digital and Analog I/O, Peripheral Controllers, Graphics and Speech Recognition, Packaging and Software**
- **Supported by Over 200 Vendors Providing Over 2000 Compatible Products**

The MULTIBUS® System bus is one of a family of standard bus structures resident within Intel's total system architecture. The MULTIBUS interface is a general purpose system bus structure containing all the necessary signal lines to allow various system components to interact with one another. This device interaction is built upon the master-slave concept. The "handshaking" between master and slave devices allows modules of different speeds to use the MULTIBUS interface and allows data rates of up to 5 million transfers per second. The MULTIBUS system bus can support multiple master devices (16) on a 18 inch backplane and can directly address up to 16 megabytes of memory. As a non-proprietary, standard system bus, the MULTIBUS interface has become the most prominent 8/16-bit microcomputer system bus in the industry with over 200 vendors supplying over 2000 MULTIBUS compatible products. Its success as the industry standard has been reinforced by adoption of the MULTIBUS specification by the Institute of Electrical and Electronic Engineers—(IEEE 79688 System Backplane Bus). MULTIBUS-based systems have been designed into applications, such as, industrial automation and control, office systems and word processing, graphics systems and CAD/CAM, telecommunications systems and distributed processing.



280294-1

July 1989
Order Number: 280294-001

FUNCTIONAL DESCRIPTION

Architectural Overview

The MULTIBUS® system bus is the physical framework and the conceptual foundation of Intel's total system architecture. It is a general purpose system bus used in conjunction with the single board computer concept to provide a flexible mechanism for inter-module processing, control and communication. The MULTIBUS interface supports modular CPU, memory and I/O expansion in flexible, cost effective microcomputer system configurations. These configurations implement single board computers and expansion modules in a multiple processor approach to enhance system performance. This enhanced performance is achieved through partitioning of overall system functions into tasks that each of several processors can handle individually. When new system functions are added (peripherals) more processing power can be applied to handle them without impacting existing processor tasks.

Structural Features

The MULTIBUS interface is an asynchronous, multiprocessing system bus designed to perform 8-bit and 16-bit transfers between single board computers, memory and I/O expansion boards. Its interface structure consists of 24 address lines, 16 data lines, 12 control lines, 9 interrupt lines, and 6 bus exchange lines. These signal lines are implemented on single board computers and a mating backplane in the form of two edge connectors resident on 6.75" × 12.00" form factor PC boards. The primary 86-pin P1 connector contains all MULTIBUS signal lines except the four address extension lines. The auxiliary 60-pin P2 connector contains the four MULTIBUS address extension lines, and reserves the remaining 56 pins for implementing the iLBX Execution Bus into the MULTIBUS system architecture.

Bus Elements

The MULTIBUS system bus supports three device categories: 1) Master, 2) Slave, 3) Intelligent Slave.

A bus master device is any module which has the ability to control the bus. This ability is not limited to only one master device. The MULTIBUS interface is capable of supporting multiple masters on the same system through bus exchange logic. Once access has been acquired by a master device, it has a period of exclusive control to affect data transfers through a generation of command signals, address signals and memory or I/O addresses.

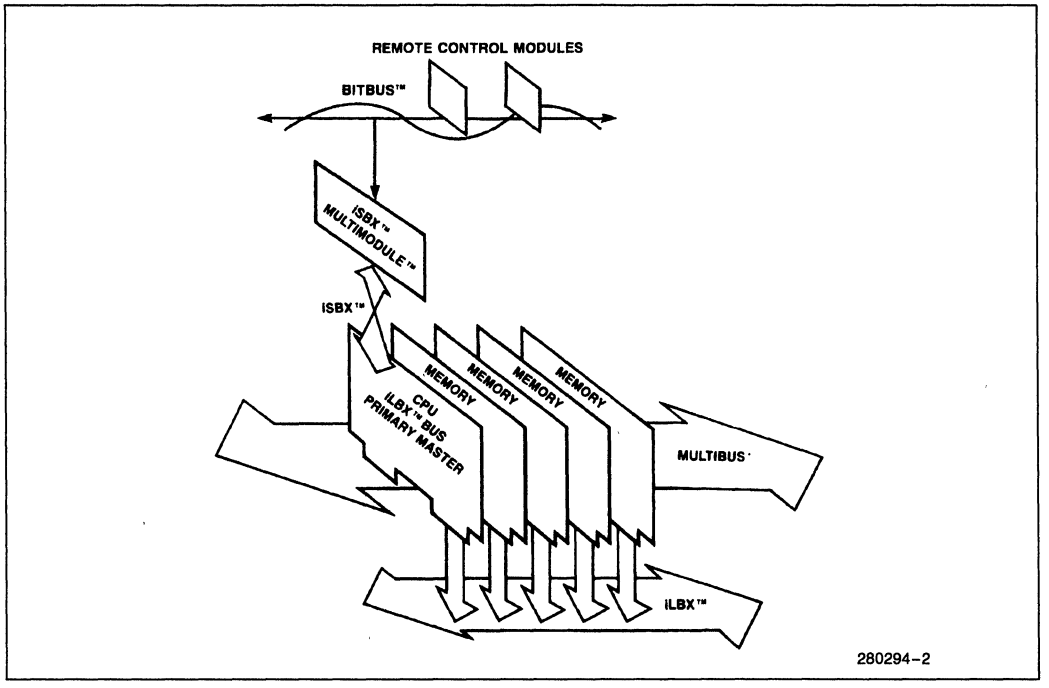


Figure 1. MULTIBUS® System Architecture

A bus slave device is a module that decodes the address lines on the MULTIBUS and acts upon the command signals from the bus masters. Slave devices are not capable of controlling the MULTIBUS interface.

The intelligent slave has the same bus interface attributes as the slave device but also incorporates an on-board microprocessor to control on-board memory and I/O tasks. This combination of on-board processor, memory and I/O allow the intelligent slave to complete on-board operations without MULTIBUS access.

Bus Interface/Signal Line Descriptions

The MULTIBUS system bus signal lines are grouped into five classes based on the functions they perform: 1) control lines, 2) address and inhibit lines, 3) data lines, 4) interrupt lines, 5) bus exchange lines. Figure 2 shows the implementation of these signal lines.

The MULTIBUS control lines are broken down into five sub-groups: clock signals (2), commands (4), acknowledge (1), initialize (1), and lock (1). The two clock signals provide for the generation of a master

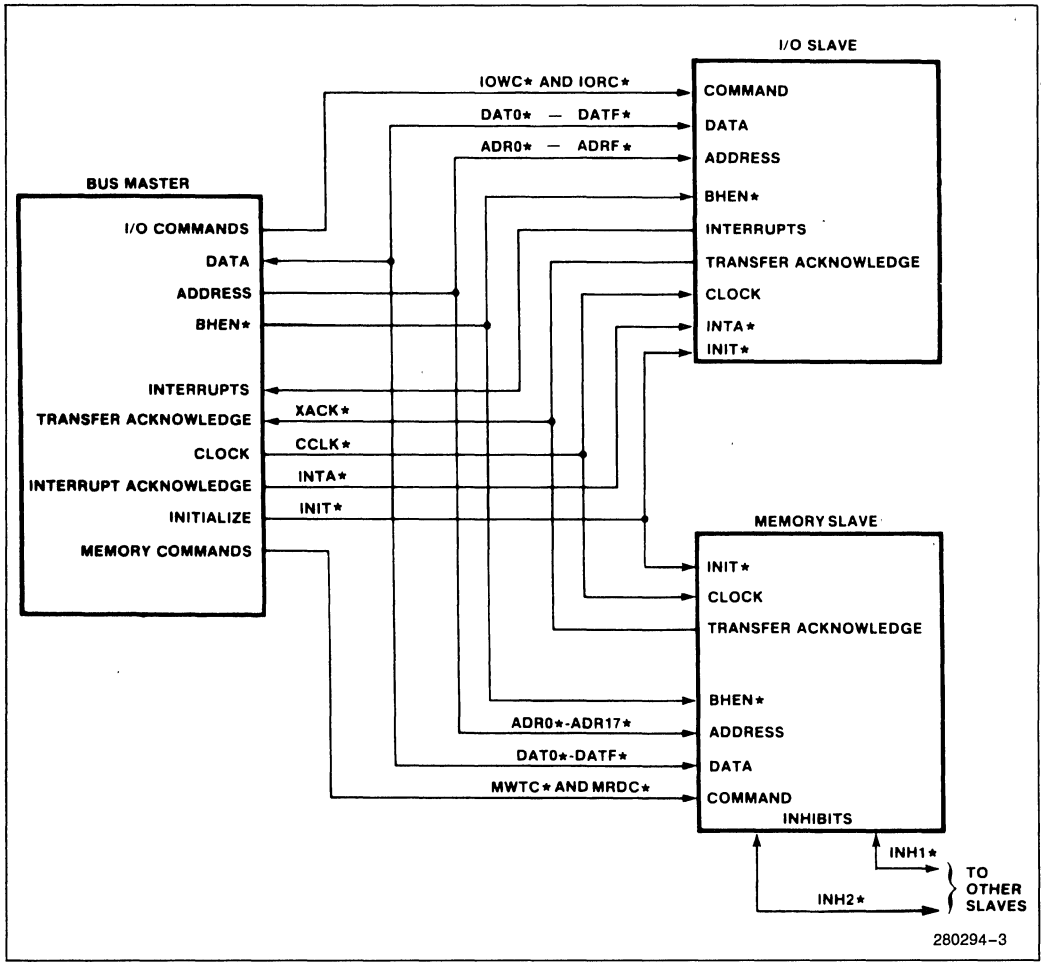


Figure 2. MULTIBUS® Interface Signal Lines

clock for the system and the synchronization of bus arbitration logic. The four command lines are the communication links between the bus masters and bus slaves, specifying types of operations to be performed such as reads or writes from memory or I/O. The transfer acknowledge line is the slave's acknowledgement that a requested action of the master is complete. The initialize signal is generated to reset the entire system to a known state. The lock signal is used by an active bus master to lock dual-ported for mutual exclusion.

The address and inhibit lines are made up of 24 address lines, two inhibit lines, and one byte control line. The 24 address lines are signal lines used to carry the address of the memory location or the I/O device that is being referenced. These 24 lines allow a maximum of 16 million bytes of memory to be accessed. When addressing an I/O device, sixteen address lines are used to address a maximum of 64 thousand devices. The two inhibit lines are used to allow different types of memory (RAM, ROM, etc.) having the same memory address to be accessed in a preferred priority arrangement. The byte control line is used to select the upper byte of a 16-bit word in systems incorporating 16-bit memory and I/O modules.

The MULTIBUS interface supports sixteen bi-directional data lines to transmit or receive information to or from a memory location or an I/O port.

The MULTIBUS interrupt lines consist of eight interrupt request lines and one interrupt acknowledge line. Interrupts are requested by activating one of the eight interrupt request lines. The interrupt acknowledge signal is generated by the bus master when an interrupt request is received. It effectively freezes interrupt status and requests the placement of the interrupt vector address onto the data lines. There are six bus exchange lines that support two bus arbitration schemes on the MULTIBUS system bus. A bus master gains control of the bus through the manipulation of these signals. The bus request, bus priority, bus busy, and bus clock signals provide for a slot dependent priority scheme to resolve bus master contention on the MULTIBUS interface. Use of the common bus request signal line can save arbitration time by providing for a higher priority path to gain control of the system bus.

Bus Operation Protocol

DATA TRANSFER OPERATION

The data transfer operation of the MULTIBUS system bus is a straight-forward implementation of an asynchronous master-slave handshaking protocol. Figures 3 and 4 show the basic timing for a read and write data transfer operation. A MULTIBUS data transfer begins by having the bus master place the memory or I/O port address on the address bus. If the operation is a write, the data is also placed on the data lines at this time. The bus master then generates a command (I/O read or write, or memory read or write) which activates the appropriate bus slave. The slave accepts the data if it is a write operation, or places data on the data bus if it is a read. A transfer acknowledge is then sent to the bus master by the bus slave, allowing the bus master to complete its cycle, removing the command from the command line, and then removing the address and data from the MULTIBUS interface.

INTERRUPT OPERATIONS

The MULTIBUS interface supports two types of interrupt implementation schemes, Non-Bus Vectored and Bus Vectored. Non-Bus vectored interrupts are interrupts handled on the bus master which do not require the MULTIBUS interface for transfer of the interrupt vector address. The interrupt vector address is generated by the interrupt controller on the master and transferred to the processor over the local bus when an interrupt request line is activated by a slave module over the MULTIBUS interface. Bus vectored interrupts are interrupts which transfer the interrupt vector address along the MULTIBUS data lines from the slave to the bus master using the interrupt acknowledge command signal for synchronization. When an interrupt request occurs, the interrupt control logic on the bus master interrupts the processor, generating an interrupt acknowledge command that freezes the interrupt logic on the bus for priority resolution and locks the MULTIBUS system bus. After the bus master selects the highest priority active interrupt request lines, a set of interrupt sequences allow the bus slave to put its interrupt vector address on the data lines. This address is used as a pointer to interrupt the service routine.

BUS EXCHANGE TECHNIQUES

The MULTIBUS system bus can accommodate several bus masters on the same system, each one taking control of the bus as it needs to affect data transfers. The bus masters request bus control through a bus exchange sequence.

The MULTIBUS interface provides for two bus exchange priority techniques: a serial technique and a parallel technique. In a serially arbitrated MULTIBUS system, requests for system bus access are ordered by priority on the basis of bus slot location. Each master on the bus notifies the next lower priority master when it needs to use the bus, and it monitors the bus request status of the next higher priori-

ty-master. Thus, the masters pass bus requests along from one to the next in a daisy chain fashion. The parallel bus arbitration technique resolves system bus master priorities using external hardware in the form of a priority resolution circuit. This parallel arbitration logic is included in many commercially available cardcages.

Mechanical Implementation

BUS PIN ASSIGNMENTS

Printed circuit boards (6.75" x 12.00") designed to interface to the MULTIBUS system bus have two connectors which plug into the bus backplane. These connectors, the 86-pin P1 (Primary) and the 60-pin P2 (Auxiliary), have specific pin/signal assignments. Because of this, the designer must insure that the MULTIBUS backplane being designed is compatible (pin-for-pin) with these two connectors. Tables 1 and 2 show the pin/signal assignments for the P1 and P2 edge connectors. The MULTIBUS interface connection is accomplished via a rigid backplane that has connectors that mate to the P1 (43/86-pin) board edge connector and allows for connectors that mate to the P2(30/60-pin) board edge connector. Figure 5 shows a typical MULTIBUS backplane. Figure 6 displays the connector and pin numbering convention. Figure 7 shows the standard MULTIBUS form-factor printed wiring board outline.

Please refer to Intel's MULTIBUS specification and iLBX bus specification for more detailed information.

Table 1. MULTIBUS® Pin/Signal Assignment—(P1)

	Pin	(Component Side)		Pin	(Circuit Side)	
		Mnemonic	Description		Mnemonic	Description
Power Supplies	1	GND	Signal GND	2	GND	Signal GND
	3	+5V	+5Vdc	4	+5V	+5 Vdc
	5	+5V	+5Vdc	6	+5V	+5 Vdc
	7	+12V	+12Vdc	8	+12V	+12 Vdc
	9		Reserved, bussed	10		Reserved, bussed
	11	GND	Signal GND	12	GND	Signal GND
Bus Controls	13	BCLK*	Bus Clock	14	INIT*	Initialize
	15	BPRN*	Bus Pri. In	16	BPRO*	Bus Pri. Out
	17	BUSY*	Bus Busy	18	BREQ*	Bus Request
	19	MRDC*	Mem Read Cmd	20	MWTC*	Mem Write Cmd
	21	IORC*	I/O Read Cmd	22	IOWC*	I/O Write Cmd
	23	XACK*	XFER Acknowledge	24	INH1*	Inhibit 1 (disable RAM)
Bus Controls and Address	25	LOCK*	Lock	26	INH2*	Inhibit 2 (disable PROM or ROM)
	27	BHEN*	Byte High Enable	28	AD10*	
	29	CBRQ*	Common Bus Request	30	AD11*	Address
	31	CCLK*	Constant Clk	32	AD12*	Bus
	33	INTA*	Intr Acknowledge	34	AD13*	
Interrupts	35	INT6*	Parallel	36	INT7*	Parallel
	37	INT4*	Interrupt	38	INT5*	Interrupt
	39	INT2*	Requests	40	INT3*	Requests
	41	INT0*		42	INT1*	
Address	43	ADRE*		44	ADRF*	
	45	ADRC*		46	ADRD*	
	47	ADRA*	Address	48	ADRB*	Address
	49	ADR8*	Bus	50	ADR9*	Bus
	51	ADR6*		52	ADR7*	
	53	ADR4*		54	ADR5*	
	55	ADR2*		56	ADR3*	
	57	ADR0*		58	ADR1*	

Table 1. MULTIBUS® Pin/Signal Assignment—(P1) (Continued)

	Pin	(Component Side)		Pin	(Circuit Side)	
		Mnemonic	Description		Mnemonic	Description
Data	59	DATE*		60	DATF*	
	61	DATC*		62	DATD*	
	63	DATA*	Data	64	DATB*	Data
	65	DAT8*	Bus	66	DAT9*	Bus
	67	DAT6*		68	DAT7*	
	69	DAT4*		70	DAT5*	
	71	DAT2*		72	DAT3*	
	73	DAT0*		74	DAT1*	
Power Supplies	75	GND	Signal GND	76	GND	Signal GND
	77		Reserved, bussed	78		Reserved, bussed
	79	−12V	−12 Vdc	80	−12V	−12 Vdc
	81	+5V	+5 Vdc	82	+5V	+5 Vdc
	83	+5V	+5 Vdc	84	+5V	+5 Vdc
	85	GND	Signal GND	86	GND	Signal GND

NOTES:
All Reserved pins are reserved for future use and should not be used if upwards compatibility is desired.
*The Reserved MULTIBUS P2 connector pin/signal assignments are contained in Intel's iLBX Bus Specification.

Table 2. MULTIBUS® Pin/Signal Assignment—(P2)

	Pin	(Component Side)		Pin	(Circuit Side)	
		Mnemonic	Description		Mnemonic	Description
	1		Reserved	2		Reserved
	3		Reserved	4		Reserved
	5		Reserved	6		Reserved
	7		Reserved	8		Reserved
	9		Reserved	10		Reserved
	11		Reserved	12		Reserved
	13		Reserved	14		Reserved
	15		Reserved	16		Reserved
	17		Reserved	18		Reserved
	19		Reserved	20		Reserved
	21		Reserved	22		Reserved
	23		Reserved	24		Reserved
	25		Reserved	26		Reserved
	27		Reserved	28		Reserved
	29		Reserved	30		Reserved
	31		Reserved	32		Reserved
	33		Reserved	34		Reserved
	35		Reserved	36		Reserved
	37		Reserved	38		Reserved
	39		Reserved	40		Reserved

Table 2. MULTIBUS® Pin/Signal Assignment—(P2) (Continued)

	Pin	(Component Side)		Pin	(Circuit Side)	
		Mnemonic	Description		Mnemonic	Description
	41		Reserved	42		Reserved
	43		Reserved	44		Reserved
	45		Reserved	46		Reserved
	47		Reserved	48		Reserved
	49		Reserved	50		Reserved
	51		Reserved	52		Reserved
	53		Reserved	54		Reserved
Address	55	ADR16*	Address Bus	56	ADR17*	Address Bus
	57	ADR14*		58	ADR15*	
	59		Reserved, Bussed	60		Reserved, Bussed

NOTES:
All Reserved Pins are reserved for future use and should not be used if upwards compatibility is desired.
*The Reserved MULTIBUS P2 connector pin/signal assignments are contained in Intel's iLBX Bus Specification.

SPECIFICATION

Word Size

Data: 8- and 16-bit

Memory Addressing

24-bits: 16 megabyte-direct access

I/O Addressing

16-bit: 64 Kbytes

Maximum Bus Backplane Length

18 inches

Electrical Characteristics

BUS POWER SUPPLY SPECIFICATIONS

Table 3

Standard(1)				
Parameter	Ground	+5	+12	−12
Mnemonic	GND	+5V	+12V	−12V
Bus Pins	P1-1,2,11,12, 75,76,85,86	P1-3,4,5,6, 81,82,83, 84	P1-7,8,	P1-79,80
Tolerance	Ref.	±1%	±1%	±1%
Combined Line & Load Reg	Ref.	0.1%	0.1%	0.1%
Ripple (Peak to Peak)	Ref.	50 mV	50 mV	50 mV
Transient Response (50% Load Change)		100 μs	100 μs	100 μs

NOTE:
1. Point of measurement is at connection point between motherboard and power supply. At any card edge connector a degradation of 2% maximum (e.g. voltage tolerance ±2%) is allowed.

Bus Devices Supported

16 total devices—(Master, Slave, Intelligent Slave)

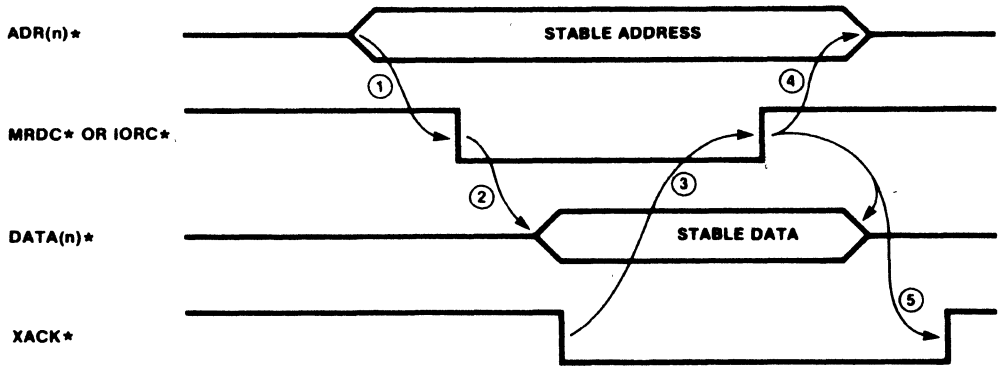
Bus Bandwidth

10 megabytes/sec: 16-bit
5 megabytes/sec: 8-bit

Bus Exchange Cycle

200 ns—Best Case; 300 ns—Worst Case (assuming no bus master is currently active on the bus.)

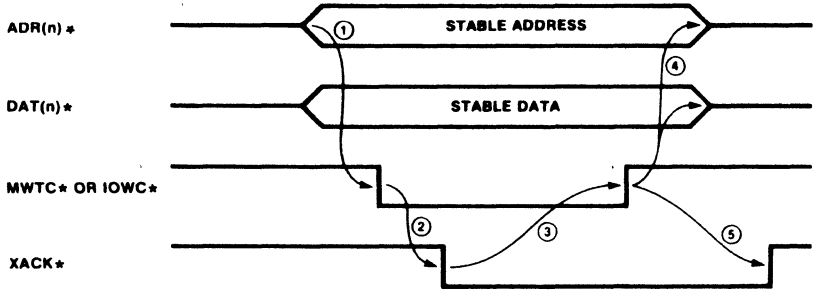
BUS TIMING



280294-4

NOTES:
1. Address Setup Time: 50 Nanoseconds Minimum.
2. Time Required for Slave to Get Data Onto Bus in Accordance with Setup Time Requirement. XACK* can be Asserted as soon as Data is on Bus.
3. Time Required for Master to Remove Command.
4. Address and Data Hold Time; 50 Nanoseconds Minimum.
5. XACK* and Data Must be Removed from the Bus a Maximum of 65 Nanoseconds after the Command is Removed.

Figure 3. Memory or I/O Read Timing

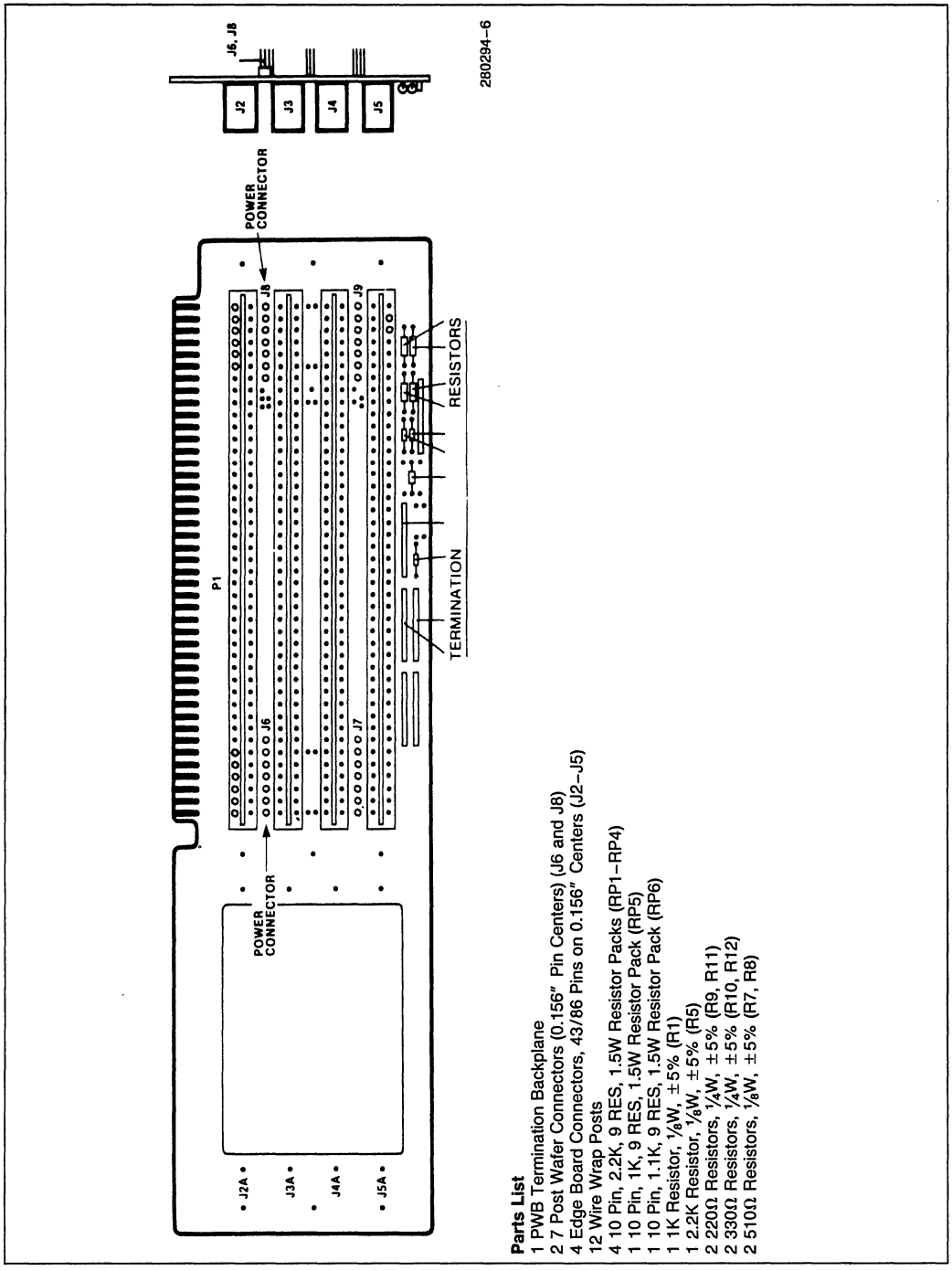


280294-5

NOTES:
1. Address and Data Setup; 50 Nanoseconds Minimum.
2. Time Required for Slave to Accept Data.
3. Time Required for Master to Remove Command from Bus.
4. Address and Data Hold Time; 50 Nanoseconds Minimum.
5. XACK* Must be off the Bus 65 Nanoseconds after Command.

Figure 4. Memory or I/O Write Timing

PHYSICAL CHARACTERISTICS



Parts List
1 PWB Termination Backplane
2 7 Post Wafer Connectors (0.156" Pin Centers) (J6 and J8)
4 Edge Board Connectors, 43/86 Pins on 0.156" Centers (J2–J5)
12 Wire Wrap Posts
4 10 Pin, 2.2K, 9 RES, 1.5W Resistor Packs (RP1–RP4)
1 10 Pin, 1K, 9 RES, 1.5W Resistor Pack (RP5)
1 10 Pin, 1.1K, 9 RES, 1.5W Resistor Pack (RP6)
1 1K Resistor, 1/8W, ±5% (R1)
1 2.2K Resistor, 1/8W, ±5% (R5)
2 220Ω Resistors, 1/4W, ±5% (R9, R11)
2 330Ω Resistors, 1/4W, ±5% (R10, R12)
2 510Ω Resistors, 1/8W, ±5% (R7, R8)

Figure 5. MULTIBUS® System Backplane Example

Physical Characteristics (Continued)

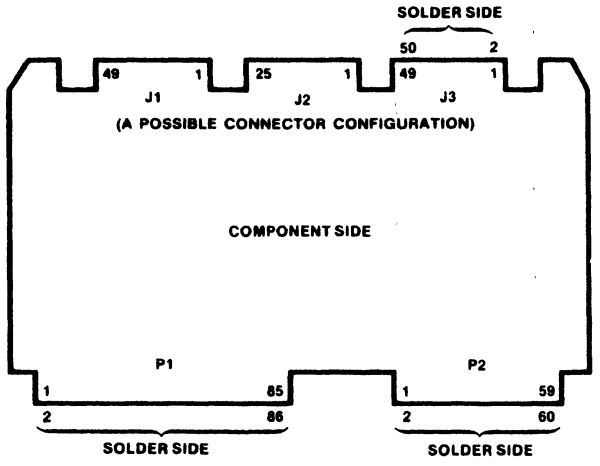


Figure 6. Connector and Pin Numbering

PHYSICAL CHARACTERISTICS (Continued)

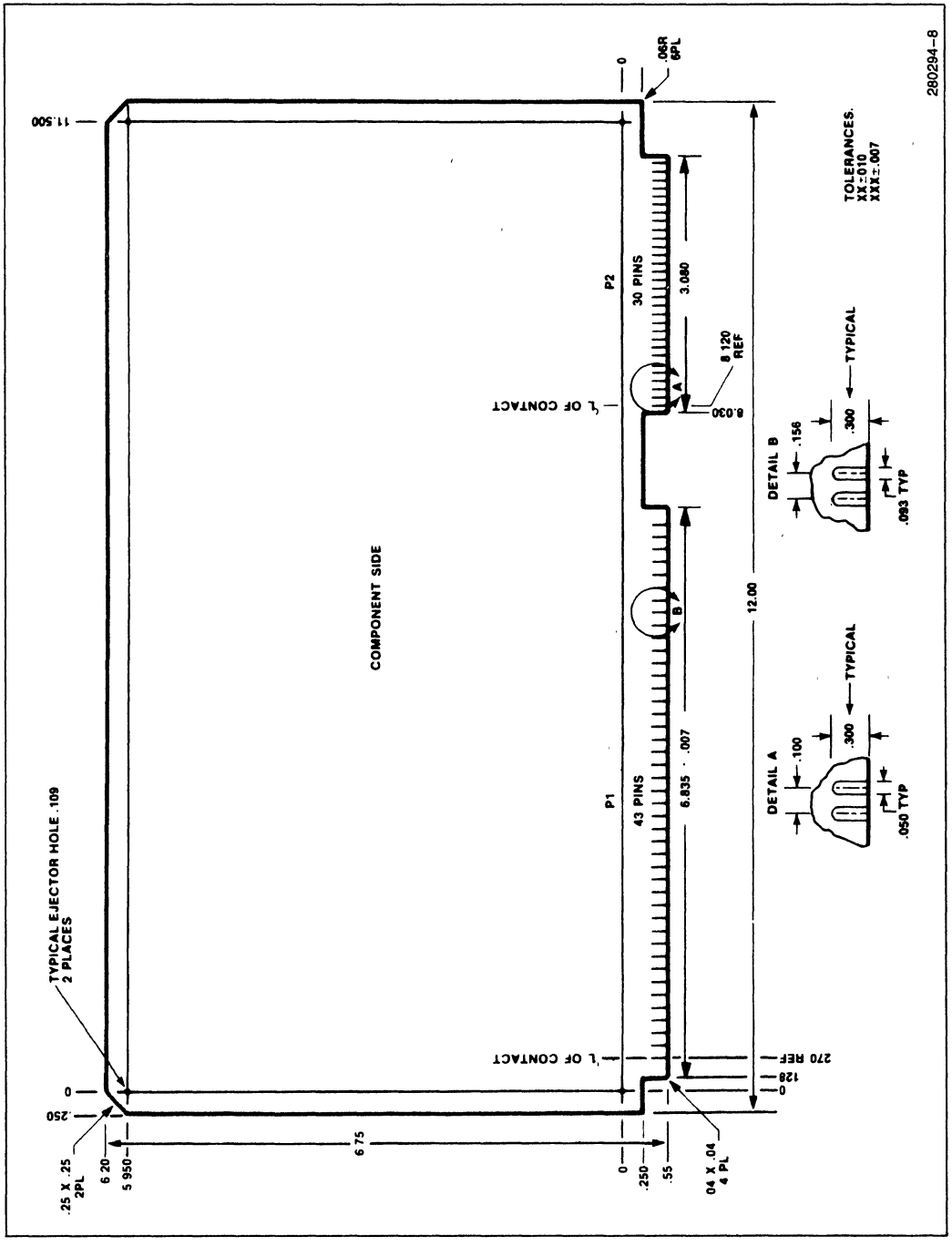


Figure 7. Standard Printed Wiring Board Outline

Backplane Connectors

Table 4. Connector Vendors

Function	# Of Pins	Centers Inches	Connector Type	Vendor	Vendor #	Intel #
Multibus Connector (P1)	43/86	0.156	Soldered(1)	VIKING ELFAB	2KH43/9AMK12 BS1562D43PBB	102247-001
Multibus Connector (P1)	43/86	0.156	Wire wrap(1, 2)	ELFAB ELDAC	BW1562D43PBB 3370860540201	102248-001
				ELFAB EDAC	BW1562A43PBB 337086540202	102273-001(3)
Auxiliary Connector (P2)	30/60	0.1	Soldered(1)	ELFAB EDAC	BS1020A30PBB 345060524802	102238-001
Auxiliary Connector (P2)	30/60	0.1	Wire wrap(1, 2)	TI VIKING	H421121-30 3KH30/9JNK	N/A(3)
				EDAC ELFAB	345060540201 BW1020D30PBB	102241-001

NOTES:
1. Connector heights are not guaranteed to conform to Intel packaging equipment.
2. Wirewrap pin lengths are not guaranteed to conform to Intel packaging equipment.
3. With mounting ears with 0.128 mounting holes.

Environmental Characteristics

Operating Temperature: 0°C to 60°C; free moving air across modules and bus

Humidity: 90% maximum (no condensation)

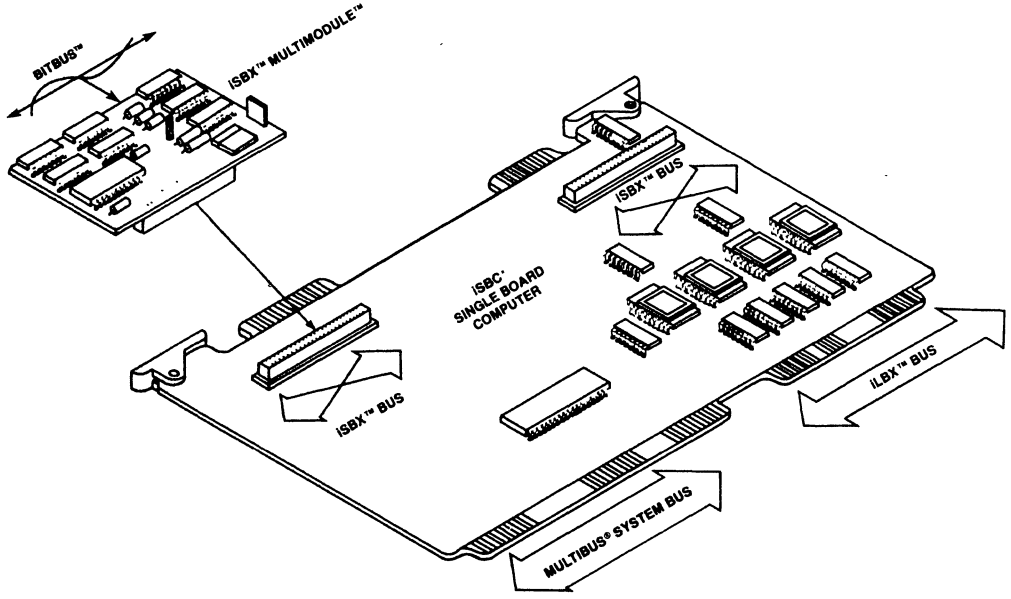
Reference Manuals

210883-002— MULTIBUS Architecture Reference Book

iLBX™ EXECUTION BUS

- **High Bus Bandwidth**
  - **— 9.5 Mbytes/sec. for 8-Bit Transfers**
  - **— 19 Mbytes/sec. for 16-Bit Transfers**
- **16 Mbyte Addressing Range**
- **8 and 16-Bit Data Transfers**
- **Supports up to 5 iLBX™ Compatible Devices Per Bus**
- **Primary and Secondary Master Bus Exchange Capabilities**
- **Standard 60-Pin MULTIBUS® P2 Connector**

The iLBX™ Execution Bus is one of a family of standard bus structures resident within Intel's total system architecture. The Local Bus Extension (iLBX) Bus is a dedicated execution bus capable of significantly increasing system performance by extending the processor board's on-board local bus to off-board resources. This extension provides for arbitration-free, direct access to high-performance memory. Acting as a "virtual" iSBC®, up to 16 megabytes of processor addressable memory can be accessed over the iLBX bus and appear as though it were resident on the processor board. The iLBX Bus preserves advantages in performance and architecture of on-board memory, while allowing memory configurations larger than possible on a single board computer. High throughput and independence from MULTIBUS® activities make the iLBX bus an ideal solution for "working store" type program memory and data processing applications requiring large amounts of high performance memory. Such applications include graphics systems, robotics, process control, office systems, and CAD/CAM.



280215-1

September 1988
Order Number: 280215-001

FUNCTIONAL DESCRIPTION

Architectural Overview

The iLBX bus is an architectural solution for supporting large amounts of high performance memory. It is the first structure that allows the CPU board selection to be decoupled from the on-board memory requirement, and still maximizes the processor's performance potential. It eliminates the processor's need to access its off-board memory resources solely over the MULTIBUS system bus. Architectural consistency with the single board computer approach including iLBX memory can be maintained by dual port access of memory resources between the iLBX bus and the MULTIBUS system bus. This allows for global access by other processors and I/O devices while still providing high speed local CPU operations. This sub-system created by the iLBX bus of a single board computer and a maximum of 4 memory cards can be perceived architecturally as a "virtual single board computer". The implementation of iLBX bus "virtual modules" makes it possible to create functional modules with a new level of flexibility and performance in implementing a wide range of memory capabilities. With future needs in mind, the iLBX bus has the capability of accessing a full 16 megabytes of memory.

Structural Features

The iLBX bus uses a non-multiplexed 16-bit configuration capable of 8 and 16-bit transfers. Used in conjunction with the MULTIBUS interface, the iLBX bus resides on the MULTIBUS form factor P2 connector and supercedes the MULTIBUS interface definitions for the P2 signals. The iLBX bus uses the standard 60-pin MULTIBUS P2 connector and occupies 56 of the P2 connector pins with 16 data lines, 24 address lines plus control, command access, and parity signals. The four MULTIBUS address extension lines on the MULTIBUS/iLBX P2 connector retain the standard MULTIBUS interface definition.

Bus Elements

The iLBX bus supports three distinct device categories: 1) Primary Master, 2) Secondary Master, 3) Slave. These three device types may be combined to create several iLBX local busses ranging (in size) from a minimum of two to a maximum of five devices per iLBX bus. There is only one Primary Master in any given implementation of iLBX bus, and its presence is required along with the attachment of at least one Slave device. To provide alternate access over an iLBX bus, one optional Secondary Master

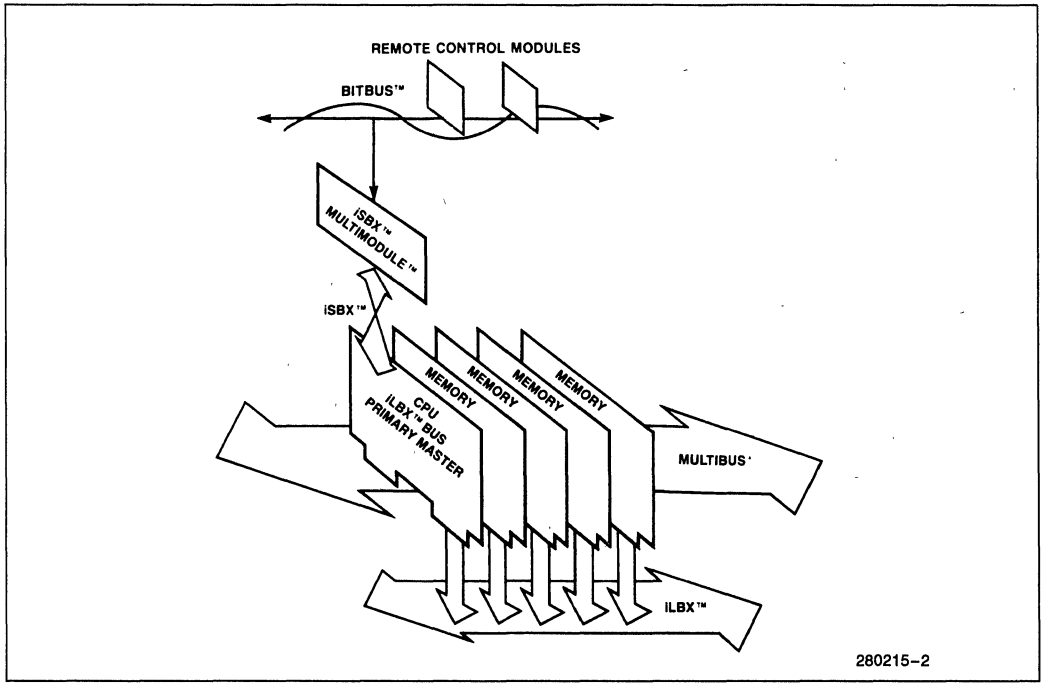


Figure 1. MULTIBUS® System Architecture

may be incorporated to create a "two-master" local bus subsystem. By limiting the iLBX bus to two masters (a Primary and a Secondary), bus arbitration is reduced to a simple request and acknowledge process, with privileged use of the bus maintained by the Primary Master, and limited access granted to the Secondary Master when needed.

The Primary Master executes the role of iLBX bus "supervisor" by controlling the general operation of the bus and managing Secondary Master accesses to the Slave memory resources.

The Secondary Master Device is an option providing alternate access to the Slave resources on the iLBX bus. Secondary master devices are typically DMA driven. This feature is provided for implementation flexibility when occasional DMA transfers in and out of iLBX memory resources can optimize the overall system performance. The Secondary Master essentially duplicates the Primary Master's data transfer capability, but must rely on the Primary Master to grant access. Once access is granted, the Secondary Master controls the bus, and drives all signal lines until the operation is complete and control is passed back to the Primary Master.

The Slave devices contain the memory resources used by the Primary Master and the optional Secondary Master. Each iLBX implementation can contain a maximum of four Slave devices. Using 64K RAM technology on four slave devices with ECC can provide for over 2 megabytes of "on-board" high performance memory. With 256K RAM chips, each iLBX bus could contain slave devices with memory totalling 8 megabytes. As memory technology increases, the iLBX bus is designed to incorporate it in rapid fashion because it is capable of directly accessing a full 16 megabytes of memory on its high-performance Slave devices.

Bus Interface/Signal Line Descriptions

The iLBX bus interface is divided into four functional classes of signal lines: address and data lines, control lines, command lines, and bus access lines. The 40 address and data lines defined by the iLBX Bus Specification consist of 16 data lines and 24 address lines.

There are 16 bi-directional data lines exclusively used to handle 8-bit and 16-bit data transfers between the active bus master and the selected slave device. The iLBX bus uses these data lines for all data transfers, and are driven by tri-state drivers.

The 24 address lines on the iLBX bus provide the ability to directly address 16 megabytes of memory. These single-direction address lines are exclusively driven by the active bus master. The iLBX bus master uses them to select a specific slave device. Three control lines specify the type of data transfer between master and slave devices, while the three command lines initiate, control, and terminate the transfer. There are also three bus access lines used to transfer bus control between master devices.

Bus Pin Assignments

The iLBX bus uses the standard 60-pin MULTIBUS P2 connector. The physical location of each pin assignment and its corresponding function is listed in Table 1. The four MULTIBUS address extension lines (pins 55–58 on the P2 connector) retain the standard MULTIBUS interface functions.

Bus Operation Protocol

The operation protocol for the iLBX bus is a straightforward set of procedures consisting of three basic operations: bus control access, write data to memory, read data from memory. These operations use asynchronous protocol with positive acknowledgment.

Bus Access

The iLBX bus is shared by at most two masters; one Primary Master and one optional Secondary Master, each providing an alternate access path to iLBX bus memory resources. The mechanism for obtaining bus access is a simple request and acknowledge process communicated between masters. Each master is a bus controller of similar capabilities, responsible for data transfer operations between devices, but the Primary Master has the added responsibility of controlling iLBX bus accesses.

The Primary Master has default control of the iLBX bus. If the Secondary Master needs access to the bus, it must initiate a request and wait for acknowledgment from the Primary Master. The choice of when to surrender control of the bus rests with the Primary Master, but if no data transfer is in progress, the Primary Master normally relinquishes control immediately to the Secondary Master.

Data Transfer Operation

The iLBX bus supports two types of data transfer operations: write data to memory and read data from memory. These data transfer operations facilitate the passing of information between the active bus master and the selected slave device. The operation of these two transfer types is very similar; the only differences being the direction of the data transfer and the device driving the data lines.

For either type of data transfer, the active bus master first initiates the transfer operation by placing the memory address on the address lines (AB23–AB0) and a control configuration on the control lines to select the slave device. Once the slave device is selected, the type of data transfer becomes the key factor. With the write operation, the active master maintains control of the data lines and provides valid data within the specified time. Upon accepting a data element, the slave sends a receipt acknowledgment signal to the master which completes the data transfer operation.

With the read operation, the slave device drives the data lines and places valid data on the data lines before sampling by the active master. The slave acknowledges the master to signal the end of the data transfer, and the master completes the operation.

The iLBX Bus Specification includes provisions for both optimized and non-optimized data transfers. Optimized operation uses pipelining and signal overlapping techniques to manage the data transfer timing relationships between the active bus master and the selected slave. The use of signal overlapping requires that every device attached to the iLBX bus provide a means of varying the timing of the slave request and acknowledge signals. The non-optimized operation uses fixed signal sequences, instead of signal overlapping, to assure a valid data transfer, and a device does not need a variable request or acknowledge to read data-valid timing on the iLBX bus. Please refer to the iLBX Bus Specification for detailed descriptions of these transfer operations.

Table 1. iLBX™ Bus Pin Assignments, P2 Edge Connector

Component Side			Solder Side		
16-Bit Pin	**Mnemonic**	**Signal Name**	**16-Bit Pin**	**Mnemonic**	**Signal Name**
1	DB0	Data Line 0	2	DB1	Data Line 1
3	DB2	Data Line 2	4	DB3	Data Line 3
5	DB4	Data Line 4	6	DB5	Data Line 5
7	DB6	Data Line 6	8	DB7	Data Line 7
9	GND	Ground	10	DB8	Data Line 8
11	DB9	Data Line 9	12	DB10	Data Line 10
13	DB11	Data Line 11	14	DB12	Data Line 12
15	DB13	Data Line 13	16	DB14	Data Line 14
17	DB15	Data Line 15	18	GND	Ground
19	AB0	Address Line 0	20	AB1	Address Line 1
21	AB2	Address Line 2	22	AB3	Address Line 3
23	AB4	Address Line 4	24	AB5	Address Line 5
25	AB6	Address Line 6	26	AB7	Address Line 7
27	GND	Ground	28	AB8	Address Line 8
29	AB9	Address Line 9	30	AB10	Address Line 10
31	AB11	Address Line 11	32	AB12	Address Line 12
33	AB13	Address Line 13	34	AB14	Address Line 14
35	AB15	Address Line 15	36	GND	Ground
37	AB16	Address Line 16	38	AB17	Address Line 17
39	AB18	Address Line 18	40	AB19	Address Line 19
41	AB20	Address Line 20	42	AB21	Address Line 21
43	AB22	Address Line 22	44	AB23	Address Line 23
45	GND	Ground	46	ACK*	Slave Acknowledge
47	BHEN	Byte High Enable	48	R/$\overline{W}$	Read Not Write
49	ASTB*	Address Strobe	50	DSTB*	Data Strobe
51	SMRQ*	Secondary Master Request	52	SMACK*	Secondary Master Acknowledge
53	LOCK*	Access Lock	54	GND	Ground
55	ADR22*	MULTIBUS® Address Extension Line 22	56	ADR23*	MULTIBUS® Address Extension Line 23
57	ADR20*	MULTIBUS® Address Extension Line 20	58	ADR21*	MULTIBUS® Address Extension Line 21
59	RES	Reserved	60	TPAR*	Transfer Parity

Mechanical Implementation

Because the iLBX bus uses the P2 connector of the MULTIBUS form factor, the iLBX bus "shares" a MULTIBUS chassis with the MULTIBUS backplane system bus in the system design. The iLBX mechanical specifications are synonymous with the MULTIBUS specifications for board-to-board spacing, board thickness, component lead length, and component height above the board. The iLBX bus interconnection can use either flexible ribbon cable or a rigid backplane. The iLBX bus interconnect maximum length is limited to 10 cm (approximately 4 inches); that is sufficient to span 5 card slots across two connected chassis. Figure 2 shows an iLBX bus cable assembly.

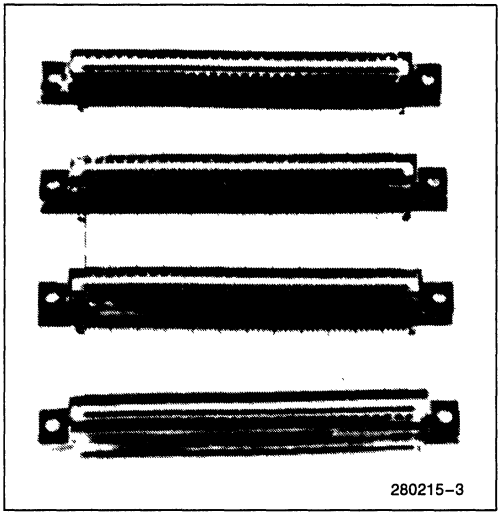


Figure 2. Typical iLBX™ Bus Interface Cable Assembly

Electrical Characteristics

DC SPECIFICATIONS

Table 2

Signal Name	Driver Type	Termination (to +5 Vdc At Master	Min Driver Requirements			Max Receiver Requirements		
			High	Low	Load Cap.	High	Low	Load Cap.
DB15–0	TRI-STATE	10 KΩ	0.6 mA	9 mA	75 pF	0.15 mA	2 mA	18 pF
TPAR*	TRI-STATE	10 KΩ	0.6 mA	9 mA	75 pF	0.15 mA	2 mA	18 pF
AB23–0	TRI-STATE	None	0.4 mA	20 mA	120 pF	0.10 mA	5 mA	30 pF
R/$\overline{W}$	TRI-STATE	None	0.2 mA	8 mA	75 pF	0.05 mA	2 mA	18 pF
BHEN	TRI-STATE	None	0.2 mA	8 mA	75 pF	0.05 mA	2 mA	18 pF
LOCK*	TRI-STATE	None	0.2 mA	8 mA	75 pF	0.05 mA	2 mA	18 pF
SMRQ*	TTL	10 KΩ	0.05 mA	2 mA	20 pF	0.05 mA	2 mA	18 pF
SMACK*	TTL	None	0.05 mA	2 mA	20 pF	0.05 mA	2 mA	18 pF
†ASTB*	TRI-STATE	10 KΩ	0.2 mA	9 mA	75 pF	0.05 mA	2 mA	18 pF
†DSTB*	TRI-STATE	10 KΩ	0.2 mA	9 mA	75 pF	0.05 mA	2 mA	18 pF
ACK*	Open Coll.	330 Ω	N.A.	20 mA	45 pF	0.05 mA	2 mA	18 pF

†At slave, additional series RC termination to GND (100 Ω, 10 pF).

SPECIFICATIONS

Word Size

Data: 8 and 16-bit

Memory Addressing

24-bits—16 megabyte—direct access

Bus Bandwidth

9.5 megabytes/sec: 8-bit
19 megabytes/sec: 16-bit

BUS TIMING

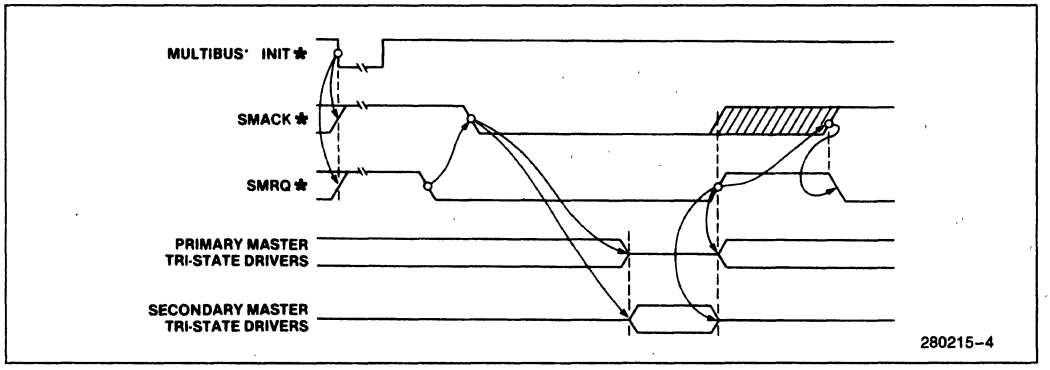


Figure 3. iLBX™ Bus Granting Timing Chart

16-Bit Transfer Timing

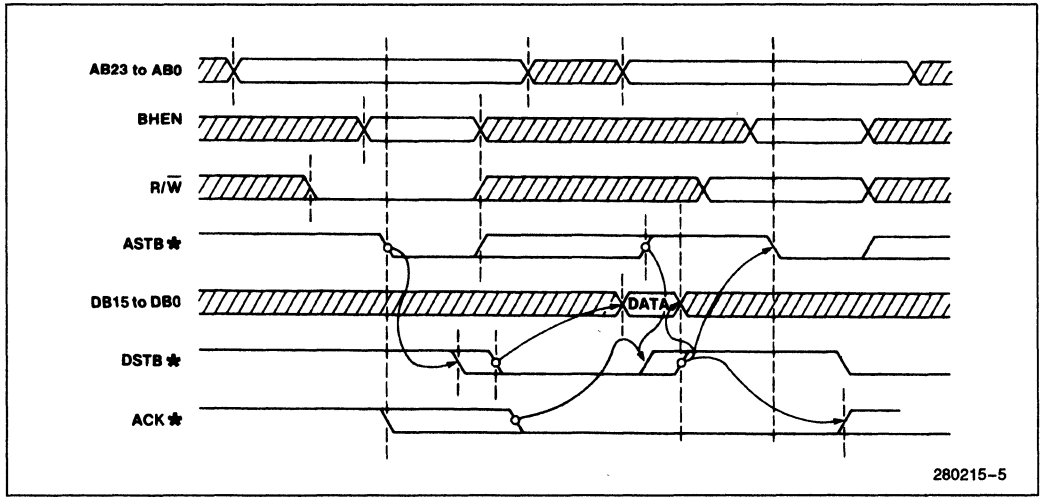


Figure 4. Write Data-to-Memory

BUS TIMING (Continued)

16-Bit Transfer Timing (Continued)

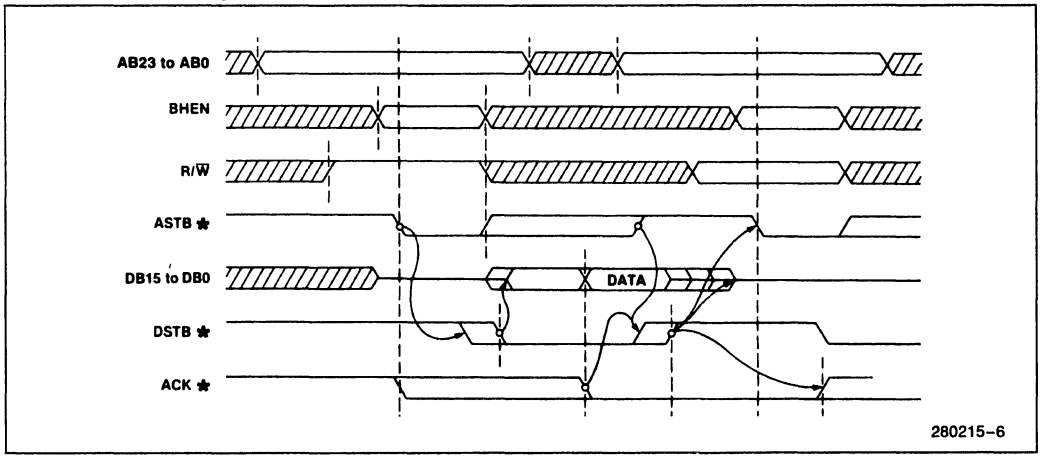


Figure 5. Read Data-From-Memory

Physical Characteristics

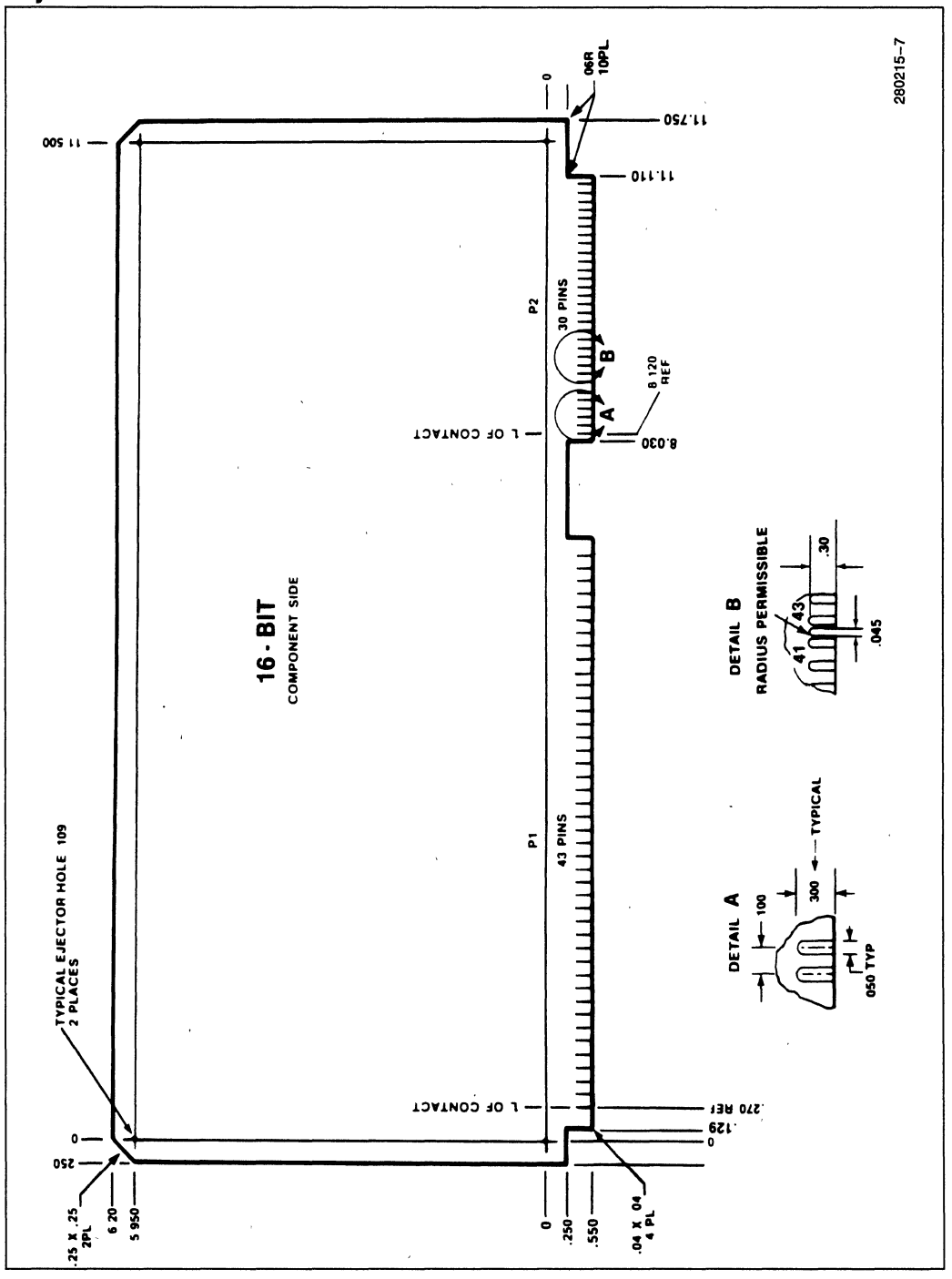


Figure 6. iLBX™ Bus Standard Printed Circuit Board Outline

Cables and Connectors

Table 3. Cable and Receptacle Vendors

iLBX™ Bus Compatible Cable		
Vendor	**Vendor Part No.**	**Conductors**
T & B Ansley	171-60	60
T & B Ansley	173-60	60
3M	3365/60	60
3M	3306/60	60
Berg	76164-060	60
Belden	9L28060	60
Spectrastrip	455-240-60	60
iLBX™ Bus Compatible Receptacles		
Vendor	**Vendor Part No.**	**Pins**
Kelam	RF30-2803-5	60
T & B Ansley	A3020 (609-6025 Modified)	60

Environmental Characteristics

OPERATING

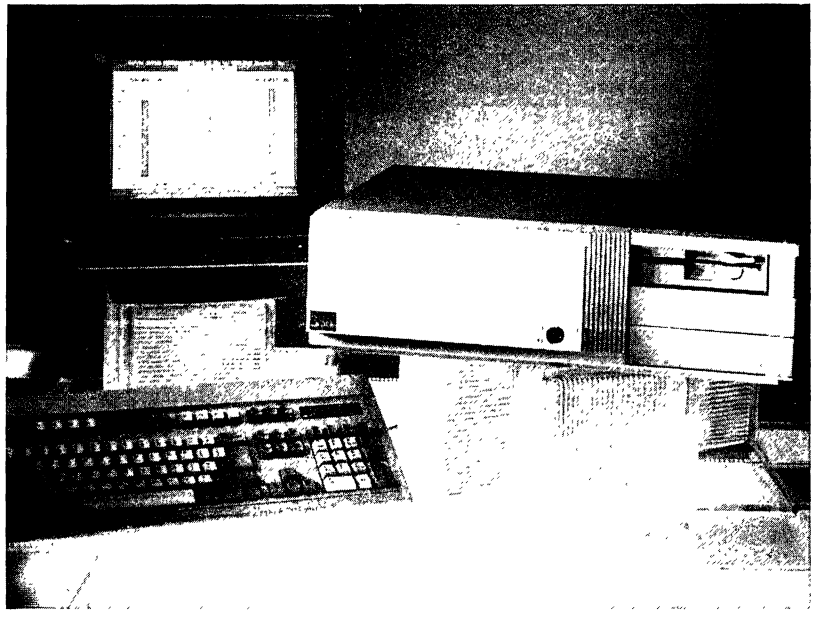
Temperature: 0°C to 60°C

Relative Humidity: 0% to 85%; non-condensing

Reference Manuals

210883-002—MULTIBUS Architecture Reference Book

ISA Boards and Systems 15



INTEL 25 MHZ 386™ PERFORMANCE IN AN ISA COMPATIBLE

Running at 25 MHz, the Intel 386™ MicroComputer Model 302 offers OEMs state-of-the-art performance in an ISA-compatible design. A 64KByte cache provides effective 0 wait state execution, without the high cost of fast-access main memory. Memory capacity is extensive, beginning with 4MB on-board, expandable to 24MB via two 32-bit expansion slots. Additionally, the Model 302 is designed to pass FCC B and VDE B levels of EMI/RFI regulations, a significant test at 25 MHz.

Based on the ISA architecture, the Model 302 is compatible with such software products as MS-DOS, OS/2, and UNIX*. Furthermore, ISA hardware products from a multitude of vendors plug into eight I/O expansion slots.

STANDARD FEATURES:

- Intel 386 microprocessor running at 25 MHz
- 64Kbyte cache (0 w.s. performance)
- 0, 2, 4, or 8MB main memory
- Phoenix Technologies ROM BIOS
- High reliability chassis
- 8 I/O expansion slots
- 220-watt power supply
- 2 32-bit I/O expansion slots
- 2 serial ports
- 1 Centronics parallel port
- 5 half-height, 5¼″ peripheral bays
- FCC-class B/VDE Level B
- UL/CSA/TUV

OPTIONS:

- Intel 387 math coprocessor running at 25 MHz
- 1.2MB floppy drive
- 8-16MB extended memory
- 40MB Winchester drive
- 4 MB and 8 MB add-in memory cards

intel

May, 1988
Order Number 280984-001

SPECIFICATIONS

BASE SYSTEM

Central Processor	Intel 386 microprocessor, 25 MHz
Floating-Point Processor	Intel 387 microprocessor, 25 MHz
Main Memory	
RAM	0, 2, 4, or 8 MB, on-board
Extended Memory	8 to 16 MB
Maximum RAM	24 MB
Cycle Time	80 ns
Data Bus Width	32-Bits
Error Detection	Bit Parity
Cache	64K Bytes, Direct map with write through
I/O	2 serial ports (asynch, RS232C, 9-pin connector) AT Compatible) 1 parallel port (Centronics compatible, 25-pin connector, AT Compatible) 8 expansion slots 2 32-bit, 16-bit, or 8-bit slots 1 8-bit slot 5 16-bit or 8-bit slots
Floppy Disk Option	5.25″ footprint 1.2 MB high density
Step Rate	3 milliseconds
Head Settling Time	15 ms max
Winchester Disk Option	5.25″ half height 40.8 MB formatted
Access Time	28 ms typ

REGULATIONS

Meets or exceeds the following requirements:

Safety	
US	UL 478 5th Edition
Canada	CSA C22.2 No. 154
Europe	IEC 435 & VDE 0806
EMI/RFI	
US and Canada	FCC 47 CFR Part 15 Subpart J, Class B
Europe	VDE 0871 Level B

ENVIRONMENT

Ambient Temperature	
System On:	15.6 to 40°C
System Off:	−34 to 60°C
Relative Humidity	
System On:	To 85%
System Off:	To 95%
Altitude	
Operating:	To 10,000 feet
Static Discharge:	Kv max

ELECTRICAL

AC Voltage/Frequency	Switching power supply, 115 V/60Hz or 230 V/50 Hz
DC Power	220 W
+5v	23.0 A maximum continuous
+12v	8.0 A maximum continuous; 12.0 A maximum surge 15 seconds
−12v	0.5 A maximum continuous
−5v	0.5 A maximum continuous

PHYSICAL CHARACTERISTICS

Length	18.7 inches (47.5 cm)
Width	21.3 inches (54.1 cm)
Height	6.4 inches (16.3 cm)
Approximate Weight (Base system)	35 pounds (15.9 kg)

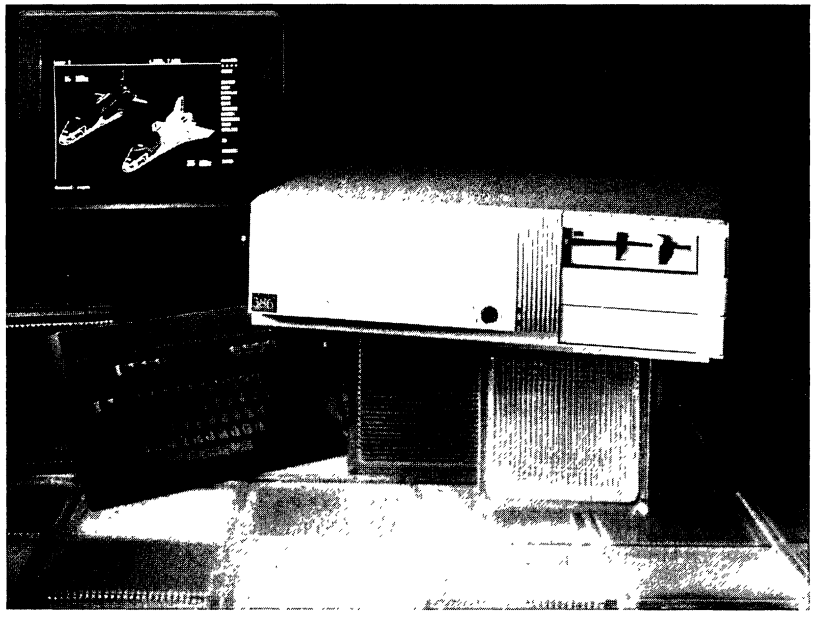
WORLDWIDE SERVICE AND SUPPORT

1 year warranty.

Field services include product installation, configuration, and maintenance.

Factory services include product repair or exchange, spare part sales, and advanced service diagnostics.

INTEL386™ MICROCOMPUTER MODEL 302-20



THE HIGH-PERFORMANCE 20 MHZ 386™ STANDARD

The Intel386™ MicroComputer Model 302-20 provides an excellent price-performance mix for the OEM building high-performance computer-based products. The Model 302-20 mother- board contains a 20 MHz 386™ microprocessor—the industry-standard workhorse of 386 computing—a socket for an Intel 387™ math coprocessor or Weitek 1167 math coprocessor, and 2 MB of interleaved main memory, expandable to 16 MB. For fast time to market, the Model 302-20 is available as an FCC certified system product.

FEATURES

- 20 MHz 386 microprocessor
- Zero wait state performance
- Socket for 387 math coprocessor or Weitek 1167
- Phoenix ROM BIOS
- Eight standard ISA I/O slots
- Two serialports, one parallel port
- Five half-height 5¼″ peripheral bays

intel

*Same product sold by Intel Puerto Rico, Inc

September, 1989
Order Number 280977-002

BOARD

Central Processor	Intel 386 microprocessor, 20 MHz
Floating-Point Processor	Intel 387 math coprocessor, 20 MHz Weitek 1167, 20 MHz
Memory	
Standard	2 MB on-board SIMM
Maximum	16 MB on-board SIMM
Cycle Time	80 ns
Data Bus Bandwidth	32 bits
Error Detection	Parity
I/O	2 serial ports (asynch, RS232C, 9-pin connector) 1 parallel port (Centronics compatible, 25-pin connector) 8 expansion slots 7 16-bit or 8-bit slots 1 8-bit slot
Options	4 MB SIMM memory 8 MB SIMM memory

SYSTEM

Peripherals	Floppy drive, 5¼", HH Four spare HH, 5¼" bays
Power Supply	220 W
Options	Customer installed hard drive

REGULATIONS

Meets or exceeds the following requirements:

Safety	
US	UL 478, 5th Edition
Canada	CSA C22.2 No. 220-1986
Europe	IEC 950
EMI/RFI	
US and Canada	FCC 47 CFR Part 15 Subpart J, Class B
Europe	VDE 0871 Level B

ENVIRONMENT

Ambient Temperature	
Operating:	10°C to 35°C
Relative Humidity	
Operating:	To 85%
Altitude	
Operating:	To 10,000 feet
Static Discharge:	7.5 Kv max

ELECTRICAL

AC Voltage/Frequency	Switching power supply, 115 V/60Hz or 230 V/50 Hz
DC Power	220 W
+5V	23.0 A maximum continuous
+12V	8.0 A maximum continuous; 12.0 A maximum surge 15 seconds
−12V	0.5 A maximum continuous
−5V	0.5 A maximum continuous

PHYSICAL CHARACTERISTICS

Length	18.7 inches (47.5 cm)
Width	21.3 inches (54.1 cm)
Height	6.4 inches (16.3 cm)
Approximate Weight (Base system)	35 pounds (15.9 kg)

WORLDWIDE SERVICE AND SUPPORT

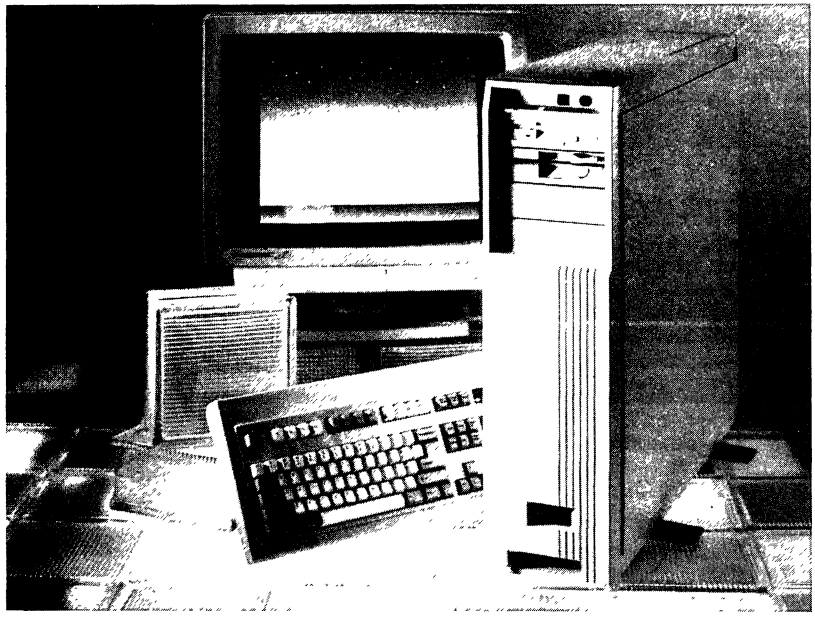
Multiply your sales potential in new markets throughout the world using Intel's worldwide service organization to install and maintain your system at your customer's site.

Field services include product installation, on-site maintenance, including third party peripherals, per-call or carry-in repair. Network installation and configuration services are also available.

Factory services include system-level, board or peripheral repair or exchange. Spare part sales and advanced service diagnostics are also available.

ORDERING INFORMATION

For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada).



33 MHZ AT-BUS PLATFORM FOR BUILDING HIGH-PERFORMANCE 386™ SYSTEMS

Based on the 33 MHz 386™ microprocessor, the Intel386™ MicroComputer Model 303 combines state-of-the-art performance, ISA compatibility, and unparalleled expansion capability to deliver a microcomputer platform ideally suited for file server and other high-performance applications.Available in either board or system configurations, the Model 303 features 33 MHz performance, 10 I/O expansion slots, and full FCC emission compliance.

FEATURES

- 33 MHz 386 motherboard with 4 MB RAM
- 10 I/O expansion slots
  - —Two 8/16/32 bit
  - —Seven 8/16 bit
  - —One 8-bit
- Full FCC Class B emission compliance
- 33 MHz 387™ math coprocessor socket
- 64 Kbyte cache with 0 wait states
- Eight half-height 5.25″ peripheral bays
- 300 watt power supply

ADDITIONAL SYSTEM-LEVEL FEATURES

- 170 MB SCSI hard drive
- 150 MB SCSI tape drive
- 1.44 MB 3.5″ floppy drive
- 1.2 MB 5.25″ floppy drive
- Power sequencing board

intel

products are manufactured by Intel Puerto Rico, Inc.

July, 1989
Order Number 280976-002

THE FASTEST 386™ ENGINE AROUND

At 33 MHz, the Intel386 MicroComputer Model 303 is the fastest 386-based compute platform on the market today. The high-speed 386 CPU can be augmented by an 387 math coprocessor, also running at 33 MHz. Performance is further enhanced by a 64 Kbyte cache memory that provides zero wait state execution without the cost of fast-access main memory.

EXPANSION FLEXIBILITY

The Model 303 motherboard was designed from the ground up for OEM customization. Standard features include 4 MB of main memory, ten I/O expansion slots, two serial ports, one parallel port, one AT-style keyboard connector, and one PS/2-style mouse connector.

On-board memory can be expanded to 8 MB using SIMM memory technology. Additional add-in memory—up to 32 MB—is available utilizing Intel's proprietary 32-bit memory bus and Intel add-in memory cards. The maximum memory configuration is 40 MB of high-speed memory.

The high-speed CPU easily supports heavy peripheral I/O traffic. The Model 303 system configuration contains eight half-height peripheral bays to support the increased storage demands of high-performance applications such as servers, CAD/CAM, and graphics. A power sequencing board supports smooth simultaneous power-up of multiple peripherals. And, the 303.5 watt power supply powers the loading of all eight peripheral bays, as well as the ten I/O slots on the baseboard.

FULL FCC EMISSION COMPLIANCE

The Model 303 baseboard has been designed for emission suppression and complies fully with FCC Class B emission requirements, a significant accomplishment at 33 MHz. The Model 303 system chassis also helps contain emissions. Emissions reduction facilitates product integration where stringent FCC-B/VDE-B compliance is required.

WORLDWIDE SERVICE AND SUPPORT

Multiply your sales potential in new markets throughout the world using Intel's worldwide service organization to install and maintain your system at your customer's site.

Field services include product installation, on-site maintenance, including third-party peripherals, per-call or carry-in repair. Network installation and configuration services are also available.

Factory services include system-level, board or peripheral repair or exchange. Spare parts sales and advance service diagnostics are also available.

SPECIFICATIONS

BOARD

CPU	386 microprocessor at 33 MHz
Floating point	387 MHz socket
Motherboard Memory	
Standard RAM	4 MB SIMM
Maximum RAM	8 MB SIMM

32-bit Add-in Memory (via 4, 8, 16 MB add-in cards)

Maximum add-in memory	32 MB
Maximum system memory	40 MB
Cycle Time	100 nsec
Error Detection	Byte Parity

64 KByte cache, 0 wait state execution on read hit (direct mapped, posted write through)

I/O EXPANSION SLOTS

- 2 8/16/32 bit slots (AT-32/ISA)
- 7 8/16 bit slots (ISA)
- 1 8 bit slot (ISA)

I/O

- 2 serial ports (9 pin)
- 1 parallel port (Centronics compatible, 25 pin)
- 1 AT style keyboard connector
- 1 PS/2 style mouse connector

BOARD DIMENSIONS

13″ × 13.6″

ELECTRICAL

Input:

AC Voltage/Frequency	115 V/60 Hz 230 V/50 Hz (externally configurable)

Output DC Voltage:

+5V	35.0 A maximum continuous
+12V	10.0 A maximum continuous (14A peak for 15 seconds)
−12V	0.5 A maximum continuous
−5V	0.5 A maximum continuous
Total Power Output:	303.5 Watts (switching)

SYSTEM

Floppy Disk Options

Footprint	3.5″
Capacity	1.44 MB
Footprint	5.25″
Capacity	1.2 MB

SCSI Winchester Disk Option

Footprint	5.25″
Capacity	170 MB
Average Seek Time	14 ms

SCSI Tape Drive Option

Footprint	5.25″
Capacity	150 MB

Eight half height 5.25″ peripheral bays (4 internal, 4 external)

Front mounted recessed reset switch

SYSTEM DIMENSIONS

Height	24.4 inches
Width	6.8 inches
Depth	27.75 inches
Base System weight	67 pounds (without peripherals)

REGULATIONS

Meets or exceeds the following requirements:

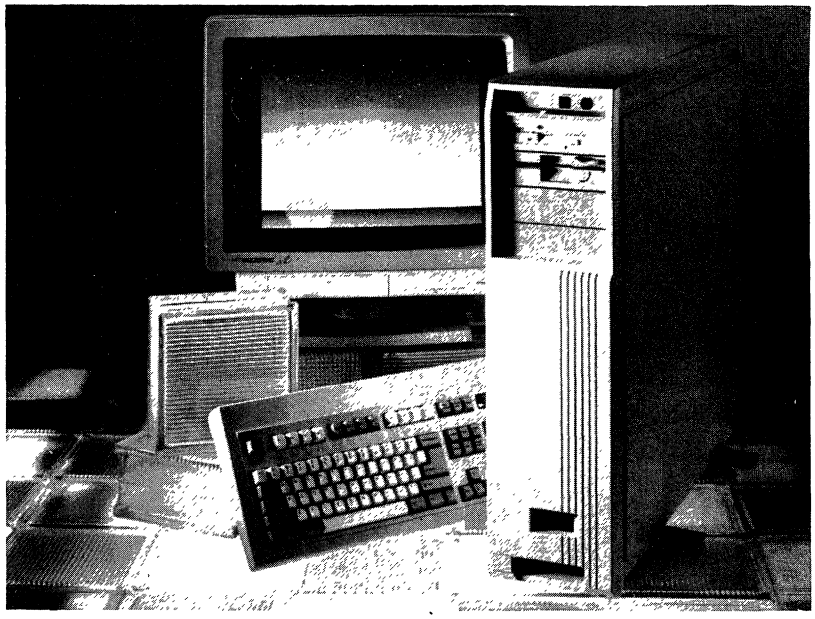
Safety

U.S.	UL 478 5th edition
Canada	CSA C22.2 No. 154
Europe	IEC 435 and VDE 0806

EMI/RFI

U.S. and Canada	FCC47 CFR Part 15 Subpart J Class B
Europe	VDE 0871 Level B

INTEL 486™ MICROCOMPUTER MODEL 401



BE THE FIRST IN YOUR MARKET WITH A 25 MHZ 486™ MICROPROCESSOR-BASED SYSTEM

Intel's 486™ Microcomputer Model 401 is the fastest way to be the first to market with a 486 microprocessor-based compute platform. The Model 401 features ISA (Industry Standard Architecture) compatibility, flexible expansion and customization, 386™ software compatibility, state-of-the-art 25 MHz 486 microprocessor performance, and Intel's world-class quality, service and support backing you up after the sale.

BOARD-LEVEL FEATURES

- 486 microprocessor running at 25 MHz
- 8 Kbytes of 4-way set-associative on-chip cache memory with zero wait-states
- High-performance main memory structure with 8 MB of interleaved 80 nanosecond DRAMs
- 8 expansion slots (4 32-bit)
- On-board floppy disk controller
- Phoenix Technologies ROM BIOS

SYSTEM-LEVEL FEATURES

- Eight half-height 5.25″ peripheral bays (4 internal, 4 external)
- 1.44 MB 3.5″ flexible disk drive
- 1.2 MB 5.25″ flexible disk drive
- 170 MB SCSI hard disk drive
- 150 MB SCSI tape drive
- Full FCC Class B emission compliance

OPTIONS

- 8 MB expansion memory boards (expandable to 32 MB)
- 101-key enhanced keyboard
- Intel worldwide service/maintenance/network support
- MS-DOS*, MS-OS/2*, and UNIX* V.3.2 software

September, 1989
Order Number 280986-001

FEATURES

A 486™ ENGINE PACKED WITH POWER.

The 486 Microcomputer Model 401 features all the configuration flexibility an OEM could want. The motherboard includes eight expansion slots, an on-board floppy disk controller, two serial ports, one parallel port, a keyboard port, and a PS/2 mouse port.

The power of the 25 MHz 486 microprocessor is enhanced by a high-speed memory structure that features interleaved 80-nanosecond DRAMs supporting zero wait-state burst mode reads.

A COMPLETELY CONFIGURABLE TOWER CHASSIS.

You can buy the Model 401 motherboard only; or, you can buy a complete Model 401 system, partially or fully integrated, ready for resale to your customers.

The Model 401 system product comes in a tower chassis that measures only 24.4″ high and 6.8″ wide—short enough to fit under the most restrictive table and slim enough to nest multiple 401s side by side in a powerful network.

The cabinet design allows for hidden peripheral and network cabling connections at the top rear of the chassis with an easy access door. Ease-of-use features include a power switch on the front bezel, and recessed keylock and reset switches.

The 401 tower has eight half-height 5.25″ peripheral bays to support the massive storage demands of high-performance applications such as servers, workstations, CAD/CAM and graphics. The high-speed CPU easily supports the heavy I/O traffic, and the 303.5 Watt power supply powers the loading of all eight peripheral bays, as well as the eight I/O slots on the baseboard.

SOFTWARE COMPATIBILITY.

The Model 401 runs industry-standard operating systems like MS-DOS, MS-OS/2, and UNIX, preserving your existing software investment and guaranteeing a smooth growth path from 386 to 486 architectures. Intel further reduces your software development and support costs through extensive hardware and software compatibility testing.

WORLDWIDE SERVICE AND SUPPORT.

Multiply your sales potential in new markets throughout the world using Intel's worldwide service organization to install and maintain your system at your customer's site.

Support contracts are available for hardware/software engineering assistance; repair and maintenance for Intel and non-Intel systems and peripherals; network design, installation and maintenance; and training.

Factory services include system-level, board or peripheral repair or exchange. Spare parts sales and advance service diagnostics are also available.

SPECIFICATIONS

BOARD

CPU	486 microprocessor at 25 MHz
Motherboard memory	
Standard	8 MB interleaved
Maximum	32 MB
Burst mode	zero wait-state reads
Data bus resolution	32 bits
Cache memory	8 Kbytes on-chip 4-way set associative cache

I/O EXPANSION SLOTS

Eight add-in expansion slots
- Four 8/16/32-bit
- Three 8/16-bit
- One 8-bit

I/O

Two serial ports (9-pin)
One parallel port (Centronics-compatible, 25-pin)
One AT-style keyboard connector
One PS/2*-style mouse connector

BOARD DIMENSIONS

12.0″ x 13.0″ (30.4 cm x 33.0 cm)

ELECTRICAL

Input:	
AC Voltage/Frequency	115V/60 Hz 230V/50 Hz (externally configurable)
Output DC Voltage:	
+5V	35.0 A maximum continuous
+12V	10.0 A maximum continuous (14 A peak for 15 seconds)
−12V	0.5 A maximum continuous
−5V	0.5 A maximum continuous
Total Power Output	303.5 Watts (switching)

SYSTEM

Floppy Disk Options	
Footprint	3.5″ half-height
Capacity	1.44 MB
Footprint	5.25″ half-height
Capacity	1.2 MB
Hard Disk Drive	
Footprint	5.25″ half-height
Capacity	170 MB
Tape Drive	
Footprint	5.25″ half-height
Capacity	150 MB

SYSTEM DIMENSIONS

Height	24.4 inches (62.0 cm)
Width	6.8 inches (17.3 cm)
Depth	27.75 inches (70.5 cm)
Base system weight	67 pounds (30.5 kg) (without peripherals)

REGULATIONS

Meets or exceeds the following requirements:

Safety	
USA	UL 478, Edition 5
Canada	CSA C22.2 No. 220
Europe	IEC 950 & IEC 380
EMI/RFI	
USA	FCC Class B; CFR 47 Part 15 Subpart J
Europe	VDE 0871 Level B
Canada	DOC; CRC c.1374, Class B

INTEL386™ MICROCOMPUTER MODEL 300SX

LOW COST 32-BIT COMPUTE PLATFORM BASED ON 386SX™ MICROPROCESSOR TECHNOLOGY

The Intel386™ MicroComputer Model 300SX is a cost-effective 32-bit compute platform based on the low-cost 386SX™ microprocessor. Available in several configurations at either the board or system level, the Model 300SX provides excellent integration flexibility for OEMs building custom 386SX systems. The Model 300SX features four slots for OEM customization, 2 MB of on-board RAM, and a high-performance disk subsystem.

FEATURES

- 16 MHz 386SX
- 2MB on-board memory
- On-board floppy controller
- Four 16-bit ISA slots
- Two AT-style serial ports
- VGA/EGA/CGA/Hercules graphics support
- Complete 32-bit software compatibility
- 387SX socket for math-intensive operations
- Small footprint chassis (system)
- Worldwide Intel service and support

intel

*300 SX products are manufactured by Intel Puerto Rico, Inc and Singapore

September, 1989
Order Number 280955-002

BROAD CONFIGURATION FLEXIBILITY

Intel offers two board-level and three different system-level configurations of the Model 300SX, so you can select the platform best suited to your needs. Board or system, with or without peripherals or chassis, the Intel386 MicroComputer Model 300SX is an excellent foundation on which to build your high-performance 386SX product.

LOW-COST BOARD-LEVEL INTEGRATION

The powerful Model 300SX compute engine is available as a standalone motherboard for integration into your custom system. The 300SX single-board computer contains the following standard features:

- 2 MB SIMM memory
- 387SX socket for math-intensive operations
- VGA/EGA/CGA Mono./Hercules graphics interface
- PS/2 mouse port
- Two AT-style serial ports
- Parallel port
- TTL and analog video connectors

LOW-COST 386SX™ TECHNOLOGY IN A HIGH-PERFORMANCE SYSTEM

The Intel386 MicroComputer Model 300SX provides more configuration options and high-performance system features than any other 386SX platform. All system configurations feature four 16-bit slots available for OEM customization, 2 MB of on-board RAM for running large applications, a high-performance disk subsystem, built-in graphics support, and a small footprint chassis.

HIGH-PERFORMANCE DISK SUBSYSTEM

The Model 300SX frees a slot for use by the OEM by providing an on-board floppy controller and an embedded Winchester controller interface right on the motherboard. A look-ahead cache boosts hard disk access times to 12 msec. Optional peripherals include a 3.5″ 1.44 MB floppy and a 3.5″ 40 MB high-performance Winchester disk.

BUILT-IN GRAPHICS SUPPORT

The Model 300SX contains on-board support for all standard color graphics monitors—VGA, EGA, CGA, Monochrome and Hercules—saving another slot you don't have to use for a graphics board. Both analog and TTL connector hardware are included on the board.

WORLDWIDE SERVICE AND SUPPORT

Multiply your sales potential in new markets throughout the world using Intel's worldwide service organization to install and maintain your system at your customer's site.

Field services include product installation, on-site maintenance, including third party peripherals, per-call or carry-in repair. Network installation and configuration services are also available.

Factory services include system-level, board or peripheral repair or exchange. Spare part sales and advance service diagnostics are also available.

SPECIFICATIONS

BOARD

CPU	386SX microprocessor at 16 MHz
Floating point math	387SX socket
Memory	
Standard RAM	2 MB SIMM
Maximum RAM	4 MB SIMM
Cycle Time	125 ns
Error Detection	Byte Parity
I/O	
2 serial ports	Async, RS 232 C, 9-pin
1 parallel port	Centronics compatible, 25-pin
1 mouse port	PS/2 compatible
4 slots	16-bit ISA compatible
Physical Characteristics	
Width	12″
Depth	10″
Weight	3.3 lbs

SYSTEM

Floppy Disk Option	
Footprint	3.5″, ⅓ Height
Capacity	1.44 MB
Winchester Disk Option	
Footprint	3.5″, ½ Height
Capacity	40 MB
Average access	19 msec
Effective access with cache	12 msec
Physical Characteristics	
Height	6″
Width	14″
Weight	27 lbs

ELECTRICAL

AC Voltage/Frequency	Switching power supply, 115 V/60 Hz or 230V/ 50 Hz; convenience outlet
DC Power	145 W
+5 V	18.0 A maximum continuous
+12 V	4.2 A maximum continuous 6.0A maximum continuous for 15 seconds
−12 V	0.3 A maximum continuous
−5 V	0.2 A maximum continuous

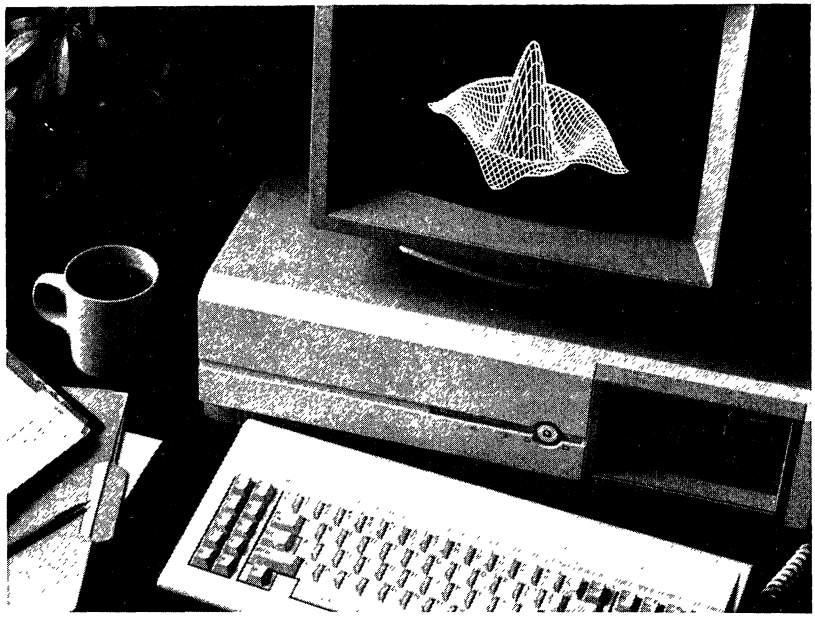
REGULATIONS

Meets or exceeds the following requirements:

Safety	
U.S.	UL 478 5th edition
Canada	CSA C22.2 No. 220
Europe	IEC 435 and VDE 0806
EMI/RFI	
U.S. and Canada	FCC47 CFR Part 15 Subpart J Class B
Europe	VDE 0871 Level B

ORDERING INFORMATION

For more information or the number of your nearest sales office, call 800-548-4725 (good in the U.S. and Canada).



HIGH-PERFORMANCE 32-BIT COMPUTE PLATFORM WITH ISA COMPATIBILITY

The Intel386™ MicroComputer Model 301Z offers the power of the 386™ microprocessor with the flexibility of the Industry Standard Architecture (ISA). This combination produces a board or system platform suitable for building high-performance applications like computer-aided design (CAD), computer-aided engineering (CAE), and advanced financial analysis, which require greater processing and memory capability. The Model 301Z features eight slots, so you can customize the system using off-the-shelf boards, operating systems, and application software.

STANDARD FEATURES:

- Intel386 processor running at 16 MHz
- 2 MB zero wait state main memory
- Eight 16-bit ISA slots
- One serial, one parallel port
- 387™ socket for math-intensive operations
- Phoenix Technologies ROM BIOS
- Worldwide Intel service and support

intel

June, 1988
Order Number 280827-005

BROAD CONFIGURATION FLEXIBILITY

Intel offers several configurations of the Model 301Z, so you can select the platform best suited to your needs. Board or system, with or without peripherals or chassis, the Intel386 MicroComputer Model 301Z is an excellent foundation on which to build your high-performance 16 MHz 386 product.

COST-EFFECTIVE BOARD-LEVEL INTEGRATION

The Model 301Z compute engine is available as a standalone motherboard for integration into your custom system. Two megabytes of on-board memory running at zero wait states, and the ability to download Phoenix BIOS into RAM, provide excellent performance. For maximum configuration flexibility, the 301Z board offers 32-bit memory, expandable to 16 MB, and eight I/O expansion slots.

The 301Z single-board computer captures the full 32-bit capabilities of the powerful 386 CPU without sacrificing compatibility with the industry-standard 8 MHz ISA bus. Exhaustive testing of numerous add-in boards, operating systems, and software assures broad compatibility across a range of applications.

QUICK TIME-TO-MARKET SYSTEM PLATFORM

The Intel386 MicroComputer Model 301Z provides a large number of configuration options and high-performance features. All system configurations of the Model 301Z feature eight slots (two 8-bit PC XT, two 8-bit PC XT or 32-bit memory expansion, and four 16-bit ISA), serial and parallel ports, and expansion capability for up to five half-height 5.25" peripheral devices.

WORLDWIDE SERVICE AND SUPPORT

Multiply your sales potential in new markets throughout the world using Intel's worldwide service organization to install and maintain your system at your customer's site.

Field services include product installation and on-site maintenance, including third-party peripherals and per-call or carry-in repair. Network installation and configuration services are also available.

Factory services include system-level, board or peripheral repair or exchange. We also offer spare part sales and advance service diagnostics.

SPECIFICATIONS

BOARD

CPU	386 microprocessor at 16 MHz
Floating point math	387 socket
Memory	
Standard RAM	2 MB on-board
Maximum RAM	16 MB
Cycle Time	125 ns
Error Detection	Byte Parity
I/O	
One serial port	Async, RS-232-C, 9-pin, AT-compatible
One parallel port	Centronics-compatible, 25-pin, AT-compatible
8 slots	2 32-bit or 8-bit slots 2 8-bit slots 4 16-bit or 8-bit slots
Physical Characteristics	
Width	13.8"
Depth	12.0"
Weight	36.8 oz

SYSTEM

Floppy Disk Option	
Footprint	5.25"
Capacity	1.6 MB unformatted
Winchester Disk Option	
Footprint	5.25"
Capacity	40.8 MB formatted
Average access	28 msec
Physical Characteristics	
Height	6.5"
Width	21.3"
Weight	44 lbs

ELECTRICAL

AC Voltage/Frequency	Switching power supply, 115 V/60 Hz or 230 V/50 Hz; convenience outlet
DC Power	220 W
+5v	23.0 A maximum continuous
+12v	8.0 A maximum continuous 11.0 A maximum total for 15 seconds
−12v	0.5 A maximum continuous
−5v	0.5 A maximum continuous

REGULATIONS

Meets or exceeds the following requirements:

Safety	
U.S.	UL 478 5th edition
Canada	CSA C22.2 No. 154
Europe	IEC 435 and VDE 0806
EMI/RFI	
US and Canada	FCC47 CFR Part 15 Subpart J Class A
Europe	VDE 0871 Level A

SOFTWARE PRODUCTS FOR FAST TIME-TO-MARKET

Intel has a wide range of software products and services to support the OEM. These products include MS® OS/2, MS-DOS,® and Diagsoft diagnostics. All have been tested and evaluated to meet Intel's high standards of compatibility and reliability. All are also supported by Intel's online Product Assistance Network (iPAN) and Intel's Phone Action Line Support (iPALS) services. Intel's OEM support program provides all the assistance you need to get your products to market quickly.

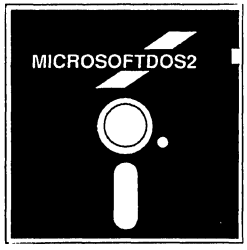
FEATURES:

- Industry-standard software
- Improves OEM time-to-market
- Proven software reduces risk
- Complete end user packages with documentation
- Intel-supplied device drivers for integrated systems
- End user break-the-seal license

intel

September, 1988
Order Number 280952-001

MICROSOFT MS® OS/2 OPERATING SYSTEM



PRESENTATION MANAGER INCLUDED

Microsoft MS OS/2 Release 1.1 is a single-user operating system for the Intel386™ MicroComputer family of products, giving users multitasking capabilities, freedom from MS-DOS memory constraints, a standardized user interface, interprocess communication and dynamic data exchange. MS OS/2 also gives you access to the wealth of MS-DOS software using the MS-DOS compatibility mode window.

The MS OS/2 Operating System has numerous features that make it a superior choice for large-memory applications. MS OS/2 provides an execution engine tailored to Intel's 386 MicroComputer family of products. It also eliminates the 640 Kbyte memory constraints by allowing you to address up to 16MB of user memory.

MS OS/2 also provides a standard human interface—presentation manager—that transcends Microsoft Windows/386® and the MS-DOS 4.01 visual shell. The MS OS/2 presentation manager represents the state of the art in graphical user interfaces for personal workstations.

MS OS/2 provides access to the full range of features of the programming environment. Users have the option of developing applications or device drivers under Intel's adaptation of MS OS/2. Users desiring this feature are encouraged to purchase the Microsoft Software Development Kit (SDK) or Microsoft Device Driver Development Kit (DDDK). Both are available through Microsoft Corporation.

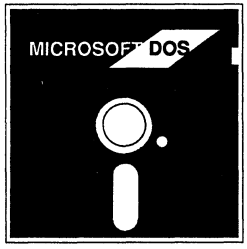
Intel's adaptation of MS OS/2 contains the basic operating system kernel, presentation manager and the basic set of MS OS/2 utilities to execute on Intel's 386 MicroComputer family of products. MS OS/2 supports the following features:

- Multitasking
- Presentation Manager
- MS-DOS 4.01 compatibility
- Intel386 architecture support
- Dynamic data exchange
- Extensive Microsoft documentation
- 16 MB addressability
- Intel-supplied device drivers
- LAN Manager support (user-supplied)
- Standard Applications Interface
- Increased disk performance
- Online Help facility

MS OS/2 CAPABILITIES

Capability	MS OS/2 1.1	MS-DOS 4.01
Multitasking	Yes	No
Multiuser	No	No
Application size	Virtual	640 Kbyte
Existing MS-DOS apps.	Yes	Yes
Shared files	Yes	Within appls.
Presentation Manager	Yes	No (Shell)
16MB addressability	Yes	No
EMS 4.0 support	Not req.	Yes

MICROSOFT MS-DOS® OPERATING SYSTEM



VISUAL SHELL INTERFACE INCLUDED

The Microsoft MS-DOS 4.01 Operating System is an updated version of Microsoft MS-DOS 3.3 that supports larger disk files/partitions, expanded memory support for 386 systems, and features an optional file directory manager. MS-DOS 4.01 is fully compatible with MS-DOS 3.3 and allows the user to migrate to a visual shell environment that is compatible with the look and feel of Microsoft Windows/386 and MS OS/2 Presentation Manager.

MS-DOS 4.01 includes support for hard disk files greater than 32MB, so the user is not required to divide large hard disk drives into smaller partitions. The new file management scheme increases performance over that of MS-DOS 3.3. The maximum amount of hard disk storage supported under MS-DOS 4.01 is 2 Gbytes.

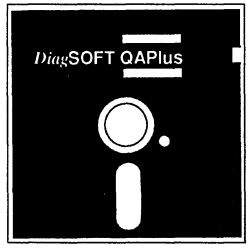
MS-DOS 4.01 emulates the expanded memory specification (EMS 4.0) developed by Lotus, Intel, Microsoft and AST Research, supporting up to 40 MB of memory. The expanded memory support is fully compatible with Intel's family of 386 MicroComputer platforms.

The visual shell interface allows MS-DOS 4.01 to experience the "look and feel" of the MS Windows/386 and MS OS/2 interfaces. Under MS-DOS 4.01 the user has the option of replacing the shell interface with MS Windows/386 (user-supplied) or the standard MS-DOS command prompt.

The MS-DOS 4.01 package contains all the necessary utilities and documentation to allow you to configure MS-DOS 4.01 on the Intel386 MicroComputer family of products. MS-DOS supports:

- Hard disk support to 2 Gbytes
- New system commands
- File caching
- GW-BASIC
- Extended Memory Support to 40MB
- Complete Microsoft documentation
- Intel-supplied device drivers
- MS-DOS 4.01 visual shell
- Enhanced utilities (FORMAT, GRAPHICS etc.)
- User installation/configuration utility

DIAGSOFT QAPLUS® SYSTEM DIAGNOSTICS



SYSTEM-LEVEL PERFORMANCE ANALYSIS

The DiagSoft Quality Assurance Advance Diagnostics System (QAPLUS) provides a complete diagnostic capability to assure the proper operation of Intel386 MicroComputer products. It contains a complete diagnostic capability that tests the CPU functionality and performs overall system analysis. It also allows you to isolate memory system faults to the component level.

The QAPLUS diagnostics system provides a complete performance analysis panel which reports on a system's operation. The performance characteristics of the system are continuously computed using dhrystone and whetstone benchmarking techniques. The QAPLUS diagnostic system also allows users to perform all standard IBM PC/AT system-level tests.

The QAPLUS diagnostic system also allows users to perform all standard IBM PC/AT system-level tests.

QAPLUS contains an exhaustive RAM test, hard disk analysis and a pre-formatting capability. With QAPLUS, the user will be able to identify any service problems and correct most common faults before initiating a service call.

The QAPLUS package includes the diagnostic program and user manual and is designed to run under MS-DOS on Intel's 386 MicroComputer family of products. The QAPLUS diagnostic system supports the following features:

- System Performance Panel
- Complete video analysis
- System configuration analysis
- Logging capability
- Complete RAM analysis
- Hard disk low-level formatting
- Multiple test capability
- Extensive system tests

SUPPORTED HARDWARE

Intel fully supports MS OS/2 Release 1.1, MS-DOS 4.01 and Diagsoft diagnostics on the Intel386 Micro-Computer product family. All Intel's software products are available in both 3.5" and 5.25" diskette media. The Intel386 MicroComputer supported products are:

- Intel386 MicroComputer Model 301
- Intel386 MicroComputer Model 301Z
- Intel386 MicroComputer Model 302

SERVICE/SUPPORT/TRAINING

Intel provides OEMs with complete technical support through the OEM Platforms Product Assistance Network (iPAN) and the OEM Platforms Phone Action Line Support (iPALS). OEM access to the iPAN and iPALS support systems require a signed OEMlicense.

For time and performance critical development projects, Intel offers MS OS/2 jointly with the Microsoft University program. The training courses include MS OS/2 system development kit (SDK) and the device driver development kit (DDDK). Intel also has a staff of factory trained software engineers which can be contracted by the hour or over several months as part of your project team. Call your local Intel sales office for course information or software engineering services.

ORDERING INFORMATION

Intel's MS OS/2 Release 1.1, MS-DOS 4.01 and Diagsoft diagnostic software may be ordered with any Intel386 MicroComputer platform. The system software products are available in two forms: as a sampler and in bulk. Each bulk unit contains five single-user copies. All bulk units require a license for purchase.

OEMLICENSE — OEM software distribution and support agreement. Each license provides for a site purchasing agreement for software products, and a single user access to iPAN and iPALS.

OEMSAMPLER — OEM software sampler package. Each sampler includes MS-DOS 4.01, MS OS/2 1.1 and Diagsoft diagnostics. Each package contains 1 copy of documentation, and both 3.5" and 5.25" diskette media. The sampler does not require a signed license for purchase.

5.25" diskette media

MSDOSF — MS-DOS release 4.01 on 5.25" diskette media (360 KB format). Each MSDOSF contains five single user copies of MS-DOS release 4.01. Each single user copy is shrink wrapped and contains MS-DOS user manuals and diskettes. MSDOSF requires a signed license for purchase.

MSOS2F — MS OS/2 release 1.1 on 5.25" diskette media (1.2 MB format). Each MSOS2F contains five single user copies of MS OS/2 release 1.1. Each single user copy is shrink wrapped and contains MS OS/2 user manuals and diskettes. MSOS2F requires a signed license for purchase.

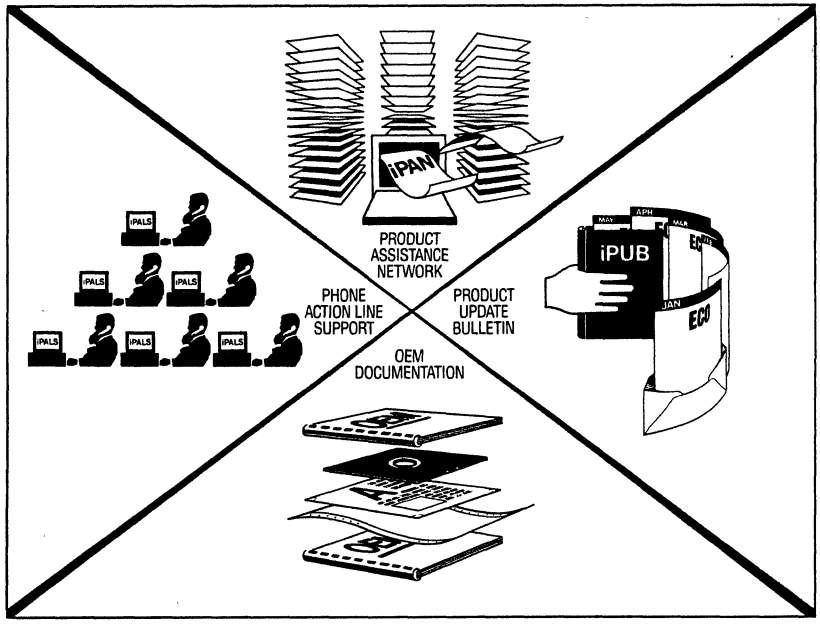
3.5" diskette media

MDOS2M — MS-DOS release 4.01 in 3.5" diskette media (720 KB format). Each MSDOSM contains five single user copies of MS-DOS release 4.01. Each single user copy is shrink wrapped and contains MS-DOS user manuals and diskettes. MSDOSM requires a signed license for purchase.

MSOS2M — MS OS/2 release 1.1 on 3.5" diskette media (1.44 MB format). Each MSOS2M contains five single user copies of MS OS/2 release 1.1. Each single user copy is shrink wrapped and contains MS OS/2 user manuals and diskettes. MSOS2M requires a signed license for purchase.

5.25" and 3.5" diskette media

DIAGSOFT — Diagsoft diagnostic software on 3.5" diskette media (720 KB format) and 5.25" diskette media (360 KB format). Each DIAGSOFT contains five single user copies of Diagsoft diagnostics. Each single user copy is shrink wrapped and contains DIAGSOFT user manuals and diskettes. DIAGSOFT requires a signed license for purchase.



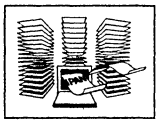
COMPLETE PRODUCT SUPPORT FOR FAST TIME TO MARKET

Intel's OEM support program provides Original Equipment Manufactures with all the assistance they need to get their Intel386™ MicroComputer products to market quickly. Program components include an electronic bulletin board open 22 hours a day, six days a week. Telephone assistance during normal business hours. Complete documentation in a choice of formats, from electronic text to printed manuals. And, monthly product updates. All designed to help you be successful, for fast time to market.

STANDARD FEATURES:

- Electronic access to Intel's OEM Platforms engineering database
- Electronic bulletin board problem resolution (open 22 hours a day, six days a week)
- Electronic access to Intel's product update bulletins (iPUB)
- Monthly subscription service to Intel's product update bulletin (iPUB)
- Phone action line support (iPALS) during normal business hours
- Product documentation in OEM-requested format

September, 1988
Order Number: 280954-001

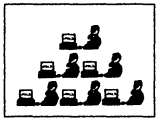


INTEL PRODUCT ASSISTANCE NETWORK (iPAN)

- On-line problem resolution, with guaranteed response.
- Electronic access to engineering database
- Electronic access to Intel's Product Update Bulletins
- Worldwide access through direct-dial or COMPUSERVE
- Access speeds up to 2400 baud

The Intel Product Assistance Network (iPAN) is an electronic information retrieval service that allows OEMs to review our on-line product database, enter questions, and tap into Intel's Product Update Bulletin service. Questions are responded to within 24 hours. The database also includes tips on workarounds and product technical information.

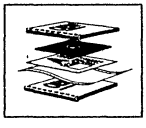
The iPAN database is available to any OEM that purchases a license for Intel386™ MicroComputer products. iPAN can be accessed using a 300, 1200 or 2400-baud modem and any standard terminal communications software. All license holders receive a complimentary COMPUSERVE introduction kit, iPAN user's manual, and an iPAN user access code. (COMPUSERVE connection time is the responsibility of the OEM.)



INTEL PHONE ACTION LINE SUPPORT (iPALS)

- Telephone assistance to Intel's technical support
- 24-hour turnaround on questions Monday-Friday
- Expert assistance on Intel386 MicroComputer products with the needs of OEMs in mind

The Intel Phone Action Line Support (iPALS) is an 8-hour-a-day, five-day-a-week telephone action line for OEMs of Intel386™ MicroComputer products. iPALS personnel are trained to answer both technical and business questions about Intel's product offerings, and are committed to having a response to you within 24 hours. iPALS is available to any OEM that has signed a license.

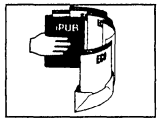


INTEL OEM DOCUMENTATION

- Complete Intel386 MicroComputer product documentation
- Choice of formats: camera-ready artwork, electronic (ASCII) text, and printed user's guides

Intel provides OEMs with complete product documentation, so their customers can become productive on Intel products. Intel OEM documentation is available in three convenient formats: camera-ready artwork, electronic (ASCII) text, and printed manuals.

Camera-ready artwork lets you customize the documentation by adding your own company logo. Electronic format provides the freedom to alter or enhance the documentation to better reflect the OEM's unique product line. Finally, printed manuals in bulk quanities provides finished documentation that the OEM can pass onto their customers immediately. Intel OEM documentation is available to all Intel386 MicroComputer OEMs.



INTEL PRODUCT UPDATE BULLETIN (iPUB)

- Monthly product updates
- Intel386 MicroComputer product histories
- Intel386 MicroComputer BIOS histories
- Software available from Intel
- Documentation updates
- Hardware and software compatibility information
- Spare parts order information

The Intel Product Update Bulletin (iPUB) is a monthly subscription service that serves as official notification to OEMs of engineering changes and technical information on the Intel386 MicroComputer product family. Information contained in iPUB includes board and system product histories, software and hardware compatibility information and spare parts order information.

iPUB is distributed electronically through iPAN and via monthly mailings. Every OEM subscriber receives an OEM binder and a one-year subscription to the service. The iPUB notebook constitutes a comprehensive resource for OEMs integrating Intel386 MicroComputer products.

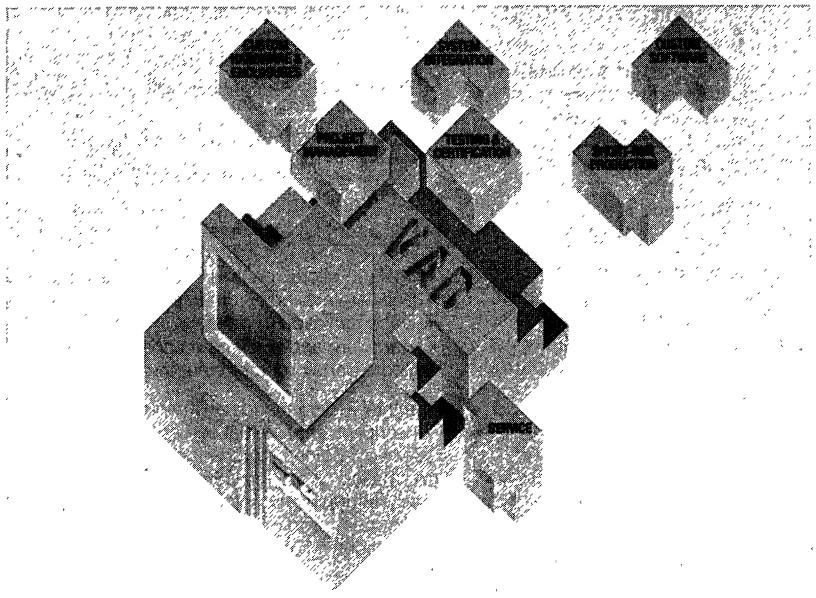
SUPPORTED HARDWARE AND SOFTWARE

The Intel OEM support program is offered to any OEM purchasing any of Intel's MicroComputer products:

- Intel386 MicroComputer Model 301
- Intel386 MicroComputer386 Model 301Z
- Intel386 MicroComputer386 Model 302
- Intel386 MicroComputer baseboard products
- Software products (MS-DOS 4.01, MS OS/2 1.1, DIAGSOFT diagnostics)

ORDERING INFORMATION

Service	Description	Order Code
iPAN	Product Assistance Network	OEMLICENSE
iPALS	Phone Action Line Support	OEMLICENSE
iPUB	Intel Product Update Bulletin	IPUB
OEMDOC	Intel386 MicroComputer	386AT25DOC 386ATZDOC 301ZDOC 302DOC



CUSTOMIZED PRODUCTS AND SERVICES FROM INTEL AND ITS VALUE-ADDED DISTRIBUTORS

Intel's Value-Added Distribution program is designed to provide complete solutions for companies requiring custom system configurations. From Intel you get leading-edge technology, training and comprehensive product service. From Intel Value-Added Distributors (VADs) you get engineering expertise, flexible manufacturing capabilities, and customized solutions. Whether you need special systems software, personalized packaging, integration of third-party components, precise delivery, or any number of unique requirements, Intel and its VAD partners can help you be successful.

COMPETITIVE ADVANTAGES

- Custom solutions based on Intel technology
- Certified Value-Added Distribution Centers staffed with design and manufacturing experts
- Application Technical Specialists to help tune designs
- Third-party hardware and software selection assistance
- Project management expertise
- Integration design, testing and evaluation
- Custom inventory control, shipping, service arrangements

INTEL TECHNOLOGY: THE BEST PLACE TO START

Intel advanced technology is the perfect foundation for building flexible, powerful, cost-effective OEM systems products. But many companies need to create a customized solution for their customers. Working together with its Value-Added Distributors, Intel can provide total turnkey solutions.

intel

April, 1988
Order Number 280979-001

PRODUCT DESIGN SUPPORT

Design and integration support can begin as early as you want. Our nationwide network of sales offices and technical support centers are staffed with Intel-certified Application Technical Specialists who are experts on Intel architectures and on integrating Intel products into custom configurations. Your Application Technical Specialist can work with you to evaluate your needs, make suggestions that will save you money, improve the quality of your product and accelerate your production time to market.

If your design solution requires third-party add-in boards, peripherals or software, our VADs can help with supplier qualification. They have established relationships with hundreds of suppliers and can quickly match your needs with pre-qualified companies that conform to your quality standards. They can also test for your unique peripheral requirements.

Project management assistance is also available. Our VADs can take on the coordination of your entire product and provide drop-shipment to your customer's site. You make all the final decisions; they take care of the day-to-day details.

In short, your VAD Center representative or branch office sales person will sit down with you, listen to and understand your needs, then work with you at whatever level you desire to come up with a creative, cost-effective solution.

PRODUCTION AND MANUFACTURING EXCELLENCE

Together, Intel VADs have invested millions of dollars in state-of-the-art integration facilities and trained personnel. These facilities are there to quickly help you turn your product ideas into a reality.

Intel's certification program ensures that all our VADs conform to Intel's rigorous quality and reliability standards. They employ the latest manufacturing and process control techniques to ensure top quality and a smooth production flow. Their manufacturing facilities can adapt to your changing needs and can ramp up or cut back production as your markets dictate.

They can also perform testing and certification to make sure your products operate safely and within prescribed legal parameters. Custom diagnostics can also be generated by your VAD to assure proper performance levels at your customers installation.

CUSTOM HARDWARE

Intel, through its VADs, can provide both board and system-level customization, including Intel386™ MicroComputer and MULTIBUS® board integration, custom motherboards and backplanes, iSBC® single-board computer test and burn-in, integration of third-party peripherals, multiple processor design, even custom logos, keycaps and chassis. You'll have the assurance that all the components are compatible, fully tested, and that each system will perform as expected.

CUSTOM SOFTWARE

Our VADs' wide-ranging expertise in custom software includes iRMX® real-time software reconfiguration, iRMX and UNIX* drivers, human interface software, communications software, PAL generation, custom diagnostics, networking, and many other areas. They will make sure software and hardware are tightly integrated and serve as a single, accountable source for the entire system.

And, of course, they can also supply you with off-the-shelf PC application software.

SPECIAL SERVICE OPTIONS

Intel's VADs will not only help you produce the highest quality product possible at the lowest possible cost; we can also help with shipping, inventory management and other post-production matters. Many of our VADs will warehouse finished product for you and deliver on demand—or, ship directly to your end user and install the product in any corner of the world. That kind of flexibility saves you expensive, redundant shipping costs and gets your product to its final destination much sooner.

For our OEM customers, we can also accommodate Just-In-Time delivery programs, service your end users directly, set up leasing programs, and provide configuration consulting. And of course, Intel can provide you with technical, sales and service support.

LET US BE YOUR SYSTEMS INTEGRATION PARTNER

If your success depends on suppliers who can provide complete solutions, call your local Intel sales office or Intel distributor today, and let us tell you more about our Value-Added Distributor program. Building on leading-edge Intel technology, our VADs can craft a custom solution for you that's just right for your unique needs.

ALMAC ELECTRONICS

14360 S.E. Eastgate Way
Bellevue, Washington 98007
Carl Gulledge/Mark Thorsteinson
(206) 643-9992

The VAD facility is composed of a 12,000 square foot production floor and engineering lab. Almac is well equipped for medium to high run rate production and engineering custom solutions. An excellent pre-sales support program has been implemented. Specific expertise includes: systems design, integration, networks, third party peripherals, and custom enclosures.

ARROW/KIERULFF ELECTRONICS

7524 Standish Place
Rockville, Maryland 20855
Keith Talbert/Andy Thomson
(301) 424-0244

5230 W 73rd St.
Edina, Minnesota 55435
Chuck Klein
(612) 830-1800

1502 Crocker Avenue
Hayward, California 94544
Scott Robertson
(415) 487-8416

Three facilities, each over 15,000 square feet, staffed by engineers, engineering technicians, production technicians, and buyers.

Highly qualified, dedicated Regional Sales people, ATS's, and Computer Products Specialists trained (and incentivized) to close VAD business. Prefer requirements for complete project management. Specific expertise includes system design and integration, board configuration and testing, cosmetic modification, MULTIBUS custom enclosures, 386AT development, custom configuration and integration.

HAMILTON-AVNET

10950 W. Washington Blvd.
Culver City, California 90230
Lynn Johnson
(213) 558-7040

3688 Nashua Drive
Mississauga, Ontario
Canada L4V 1M5
Darryl Armour
(416) 677-0690

10 F Centennial Drive
Peabody, Mass. 01960
Neal Malatzky
(508) 532-9609

Three VAD centers—LA, Boston, Toronto—65 specialists with the ability to perform a full range of software and hardware enhancement services. Strict ESD control procedures are adhered to. Specific expertise includes board modification and test, system design and integration. Full documentation is available on all systems. Specialties include real time embedded systems, multivendor networks, development systems, peripheral integration and industrial floor enclosures. Complete implementation of quality procedures.

PIONEER STANDARD

4800 E. 131st St.
Cleveland, Ohio 44105
Joe Betro/Mike Thompson
(216) 587-3600

60 Crossroads Park West
Woodbury, New York 11797
Dave Nash
(516) 921-8700

A 60 personnel, 50,000 square foot new VAD center in Cleveland. Dedicated VAD staff of adminstrators, senior buyers, engineers, technicians, Q.C. and sales. Experts in systems integration, just-in-time project management and turnkey solutions. Load operating systems and application packages, option boards, peripherals. Analyzes and tests solutions.

*XENIX is a trademark of Microsoft
UNIX is a trademark of AT&T

MESA TECHNOLOGY

9720 Patuxent Woods Dr.
Columbia, Maryland 21046
Johnny Johnson
(301) 290-8150

Specialists in all types of OEM Industrial Integration. Experience includes multiprocessing systems with heavy emphasis on real-time embedded rackmount systems. In-house MULTIBUS board design capability and system level software, especially I/O drivers, with 9 years experience in iRMX® development.

Recognized experts in TEMPEST system integration and government contracts. Design, manufacture and market their own TEMPEST system products based on AT-bus (301/2 T), MULTIBUS I (321 T, 325 T) and MULTIBUS II (521 T).

MTI SYSTEMS

38 Harbor Park Dr.
Port Washington, New York 11050
Tom Donofrio/J.P. Altier
(516) 621-6200

Highly integrated, complex applications VAD. Expert iRMX, UNIX, DOS application experience. In-house board design capability. Rackmount and custom hardware, system modification experience. Custom software expertise, UNIX packages, X-windows, iRMX drivers. Complete networking abilities, 301/302 models with multiple EGAs.

PIONEER TECHNOLOGIES

9100 Gaither Rd.
Gaithersburg, Maryland 20877
Mike Edison/Tim Olson
(301) 921-0660

A full service 20,000 square foot technical application center offering project management from inception through proposal, engineering, testing, and production/manufacturing. Pioneer's technical staff brings about 20 years of experience with microprocessor hardware, software, and system design and integration. Pioneer offers complete system solutions, including networking (supporting multiserver and remote boot), board and system level (MBI or MBII), operating systems (iRMX, XENIX*, DOS), custom software (diagnostics to drivers), peripheral subsystems, mechanical redesign, 386 platforms (model 301, 302) with iRMX, XENIX/UNIX*, or DOS, and inventory control, kitting, and manufacturing.

WYLE LABORATORIES

7382 Lampson Avenue
Garden Grove, California 92641
Dave Hamilton
(714) 891-1717

A showcase VAD Center. Expanded to 12,000 square feet, increased headcount to 14 people. Capable of processing over 700 systems per month. Excellent at total configuration of systems. Prefer system integration, peripheral enhancement, test and system run-in. Experience in Intel iSBC board run rate modifications.

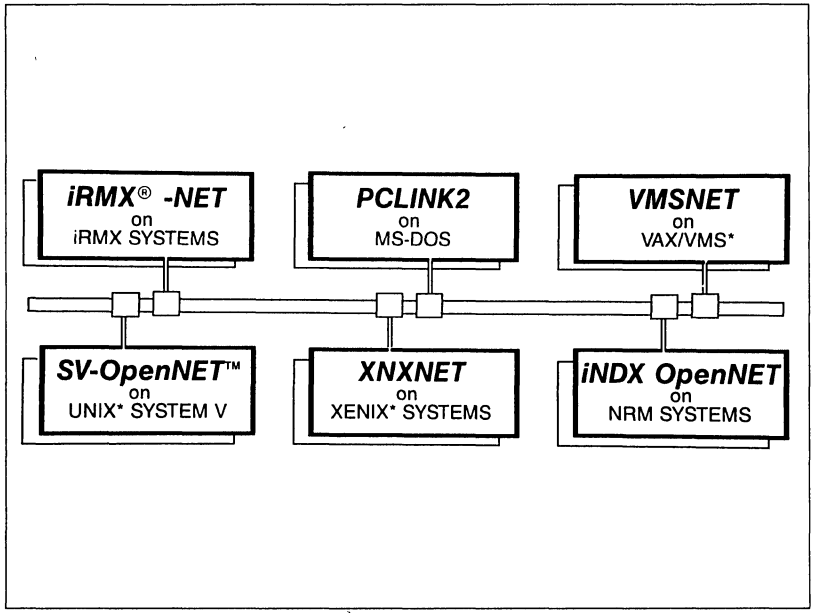
ZENTRONICS

8 Tilbury Court
Brampton, Ontario
Canada L6T 3T4
Paul Malhi
(416) 451-9600

Newest Canadian VAD. A national distributor with a VAD Center providing all types of systems interpretation. Prefer iRMX and AT-bus. Inventory drives, controllers, monitors, etc. to provide complete customer solution. In-house technical (800) hot-line for customer assistance.

Local Area Network Boards and Software

16



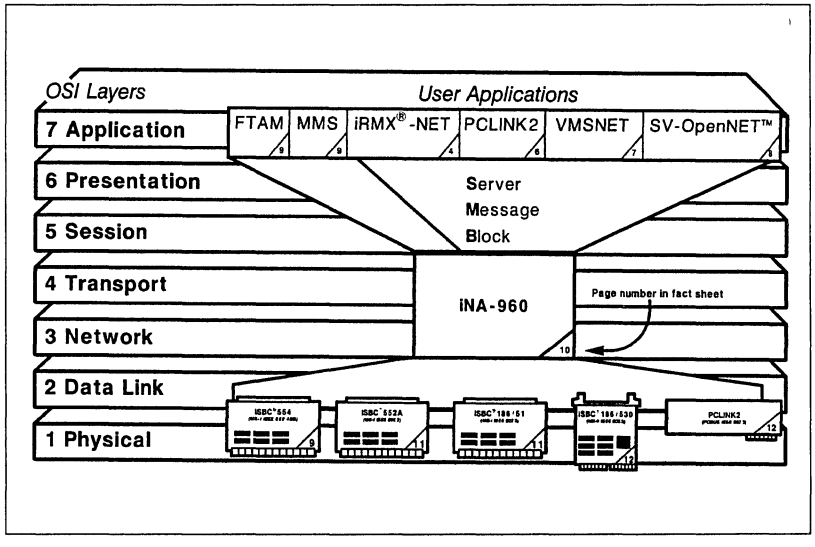
OPENNET™: THE COMPLETE OPEN NETWORK SOLUTION

The OpenNET family provides the OEM with complete *Open Network* solutions for an enterprise-wide, multi-vendor network based on international standards.

FEATURES:

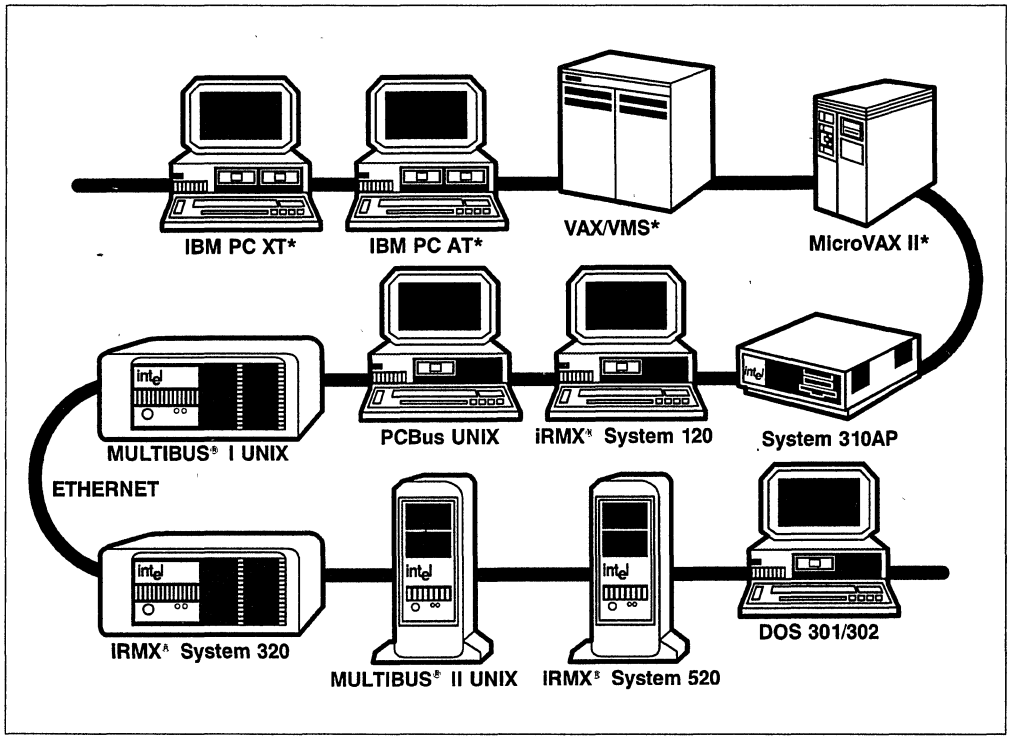
- Interoperability between the factory, office, and engineering environments
- Complete hardware and software network solutions
- On-going customer support through extensive training and application development

GUIDE TO THE OPENNET™ PRODUCTS



September, 1989
Order Number 280669-002

OpenNET™ MEANS OPEN NETWORKS



Users are placing increasing demands for data communications capabilities on their computing applications. The OpenNET family of networking products supplies those capabilities to let OEMs offer solutions to communications-intensive requirements, based on Intel's real-time computing products.

- Open to multiple media
  - IEEE 802.3/Ethernet
  - Thin-wire Ethernet
  - IEEE 802.4
  - X.25
- Open to different Operating Systems
  - iRMX
  - MS-DOS*
  - PC-DOS
  - UNIX SYSTEM V*
  - VAX/VMS*
  - XENIX*
  - iNDX
- Open to expansion
- Open to different hardware
  - MULTIBUS®I
  - MULTIBUS®II
  - PC XT/AT Bus
- Open to different environments
  - Factory
  - Office
  - Lab
  - Engineering Workstation
- Open to multi-vendor solutions
- Open to future upgrades

OpenNET™ ARCHITECTURE

Intel's OpenNET communications architecture applies networking standards to offer an open network. Open connectivity lets MULTIBUS-based systems talk to systems such as IBM PCs and VAX minicomputers.

OpenNET™ DELIVERS INTERNATIONAL STANDARDS

The OpenNET products use ISO and CCITT Data Communications Standards for the Physical, Data Link, Network, and Transport layers of the Open System Interconnection (OSI) model. The Session, Presentation, and Application layers use the Server Message Block (SMB) protocol promulgated by Intel, Microsoft, and IBM. The SMB protocol is used by Microsoft and IBM in their PC networks.

SMB: TRUE TRANSPARENT REMOTE FILE ACCESS

The OpenNET Server Message Block protocol allows applications to access remote files as if they were local. This consistent view of the file system throughout the network allows distributed processing of existing applications without change. The SMB protocol protects the user's data with file security established by either the consumer or server system.

OpenNET™ WILL FOLLOW THE STANDARDS, TRANSPARENTLY

As the OSI standards evolve, Intel will conform to them. The user's application will be protected from the changing protocols because the OpenNET Interface ("\\" on DOS or "//" on XENIX and UNIX) will NOT change. This will allow applications to move to the OSI standards with little or no modification. The OpenNET products will allow the user to run all current applications that use the OpenNET Interface on either the SMB or the OSI protocols.

OpenNET™ SUPPORT FOR DISKLESS WORKSTATION

For certain real-time applications, it is desirable that a networked system not have a local mass storage device such as a hard disk. One example is harsh environments such as factory floor process control, where rotating media can cause system reliability problems. Another example is financial workstations, where stock and market information must be updated in real-time on a centralized file server and accessed by diskless trader workstations.

The Boot Server system responds to network boot requests from diskless Boot Consumer systems and downloads both the appropriate operating system and communications software over the network to boot the diskless system. With the diskless system booted, the File Server system provides the diskless system with network access to mass storage devices. The Boot Server and File Server can reside in separate systems or in the same system on the network.

OpenNET™ VIRTUAL TERMINAL CAPABILITY

The OpenNET product family supports a Virtual Terminal (VT) capability. This feature allows a terminal or PC user to "login", over the network, as an interactive user under the remote node's operating system. VT is purchased separately as an optional application program.

ALL OpenNET™ PRODUCTS BACKED BY INTEL CUSTOMER SUPPORT

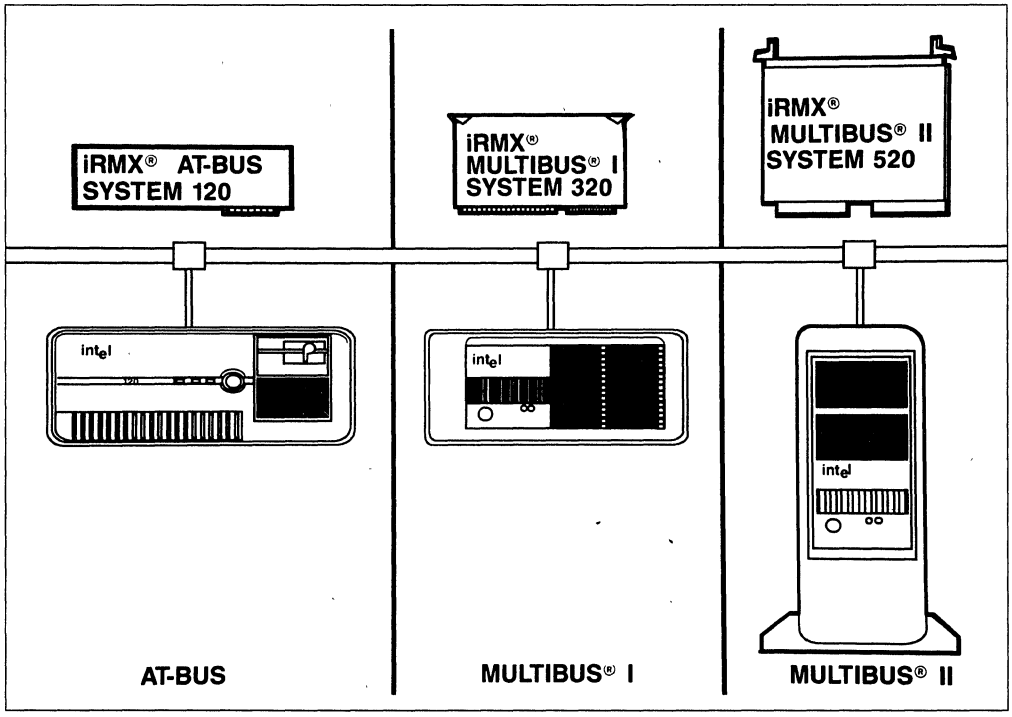
All Intel software products include Intel's Software Support for a 90 day period immediately following the licensing and receipt of the product.

STANDARD SOFTWARE SUPPORT INCLUDES:

- Product updates
- Subscription Service and technical product information distributed via:
  - —Monthly issue of *;Comments* newsletter
  - —Quarterly Troubleshooting Guides
  - —Software Problem Report (SPR) Service
- Technical Information Phone Service (iTIPS™) toll-free hot line
- Membership in Insite™ User Program Library

ADDITIONAL SERVICES AVAILABLE INCLUDE:

- Consulting services on a long or short-term basis (Systems Engineering Support)
- Worldwide training workshops on a wide variety of Intel products
- A full range of hardware maintenance services for end users or OEM/VAR customers



COMPLETE OpenNET™ SOLUTION FOR REAL-TIME SYSTEMS

Real-Time computer systems require a real-time operating system. The iRMX operating system from Intel is the world's most popular operating system for real-time systems.

Many real-time applications require network communication. Intel's iRMX®-NET Release 3.0 delivers a rich set of networking capabilities and a full range of iRMX platform support:

- Transparent Network File Access
- Transport and Distributed Name Server Software with Programmatic Access
- iRMX System 120 (AT-bus), 320 (MULTIBUS I) and 520 (MULTIBUS II) Connections
- Remote Boot for Diskless Systems

Networked iRMX systems serve in a wide range of real-time application areas including data acquisition, factory automation, financial workstations, military, medical instrumentation, simulation and process control.

TRANSPARENT NETWORK FILE ACCESS

iRMX-NET implements the NFA protocol to provide transparent file access capabilities among iRMX, DOS, VAX/VMS, UNIX, XENIX and iNDX systems on the OpenNET network. Remote files are accessed as if they resided on the local iRMX system. iRMX-NET can be configured as a network file consumer, file server, or both, depending on the application's requirements.

The iRMX operating system provides a rich set of human interface commands and system calls for accessing local files. With the addition of iRMX-NET, these commands and system calls are transparently extended to remote access as well. Transparency means that applications using the iRMX Human Interface commands or BIOS system calls do not need to know whether the files they access reside locally or on some remote system.

OSI TRANSPORT AND DISTRIBUTED NAME SERVER WITH PROGRAMMATIC INTERFACE

The iRMX-NET R3.0 product includes iNA 960 R3 OSI Transport and Network software preconfigured for a variety of Intel Network Interface Adapters.

iRMX-NET R3.0 also includes the iRMX-NET Distributed Name Server software. The Distributed Name Server software maintains and provides access to a network directory database. The database is distributed across the network with each system maintaining its own logical piece of the directory. The Distributed Name Server software provides a full set of network directory services and is used to perform such tasks as logical name to network address mapping for establishing network connections between systems.

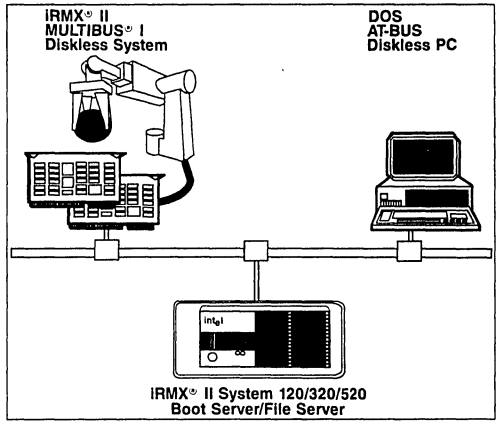
The combination of transparent network file access with iRMX commands and system calls, plus direct programmatic access to the iNA 960 Transport and iRMX-NET Distributed Name Server software gives the programmer a powerful set of capabilities for developing real-time network applications.

iRMX®SYSTEM 120, 320 AND 520 CONNECTIONS

iRMX-NET R3.0 provides networking support for the full range of Intel real-time Systems, from the low-cost AT-Bus System 120, through the MULTIBUS I System 320, to the high-end multiprocessing MULTIBUS II System 520. iRMX-NET R3.0 also supports iRMX board-level designs built around Intel's family of host CPU boards and Network Interface Adapters. Consistent operating system and networking software interfaces provide for easy development of network applications that span the various iRMX platforms.

REMOTE BOOT FOR DISKLESS SYSTEMS

iRMX-NET R3.0 supports networked diskless systems by providing network Boot Consumer, Boot Server and File Server capabilities.



PRODUCT CODES

RMXINETSW	iRMX-NET Networking Software for the iRMX 86 operating system.
RMXIINETSW	iRMX-NET Networking Software for the iRMX II operating system.
sSXM120NETKIT	Preconfigured iRMX-NET and sPCLINK2 for networking iRMX and DOS on the System 120.

REAL-TIME BOARD AND SYSTEM LEVEL SUPPORT

	iRMX® 86	iRMX® II		
	MULTIBUS® I	AT-BUS	MULTIBUS® I	MULTIBUS® II
SYSTEM	SYSTEM 310AP	SYSTEM 120	SYSTEM 310AP, 320	SYSTEM 520
HOST BOARD	iSBC 86/30 iSBC 86/35 iSBC 86/C38 iSBC 286/10(A) iSBC 286/12, 14, 16	SYSTEM 120	iSBC 286/10A iSBC 286/12, 14, 16 iSBC 386/12 iSBC 386/2X iSBC 386 3X	iSBC 386/116 iSBC 386/120 iSBC 386/258
NETWORK INTERFACE ADAPTER	iSBC 552(A) iSBC 186/51*	PCLINK2	iSBC 552(A)	iSBC 186/530

* iSBC 186/51 support requires separate purchase of iNA 960 R30

COMPLETE OpenNET™ SOLUTION FOR THE PC

Users of IBM PC AT, PC XT and other compatible computers can access Intel's OpenNET networking family through the OpenNET PC Link2 hardware and software products. The hardware connection is provided by an 80186/82586-based intelligent expansion board, the PCLINK2 Network Interface Adapter (PCLINK2NIA). The software package incorporates: MS-NET for transparent file access under DOS, iNA 961, NetBIOS interface, dynamic name resolution and user-friendly installation software.

The NetBIOS interface provides the flexibility to use the PCLINK2NIA with commercially available NetBIOS compatible applications, such as IBM's PC-LAN program. Optionally, MS-NET networking software is available for the upper layers.

TRANSPARENT NETWORKING FILE ACCESS

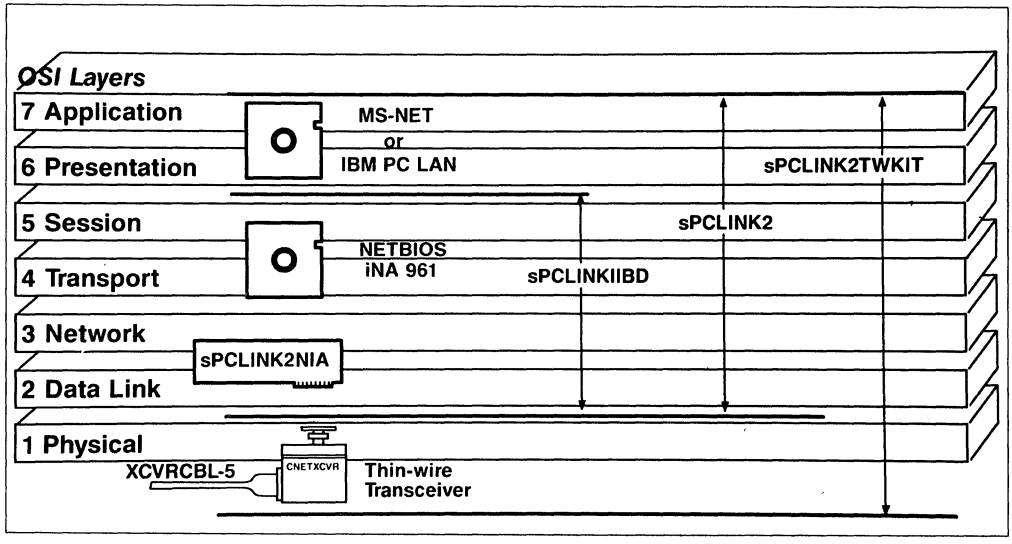
OpenNET/PCLINK2 gives users the freedom to network PCs as consumer workstations or as file servers. PCLINK2 with MS-NET implements the SMB Protocol for easy access to files on other operating systems, such as iNDX, XENIX, UNIX, iRMX, or VAX/VMS.

REMOTE BOOT FOR DISKLESS SYSTEM

Diskless workstation support for the PC is provided by on-board firmware, an iRMX-NET Boot Server and any OpenNET File Server on the network.

PRODUCT CODES

sPCLINK2NIA	PC Link2 Network Interface Adapter, hardware only
sPCLINK2	Seven-layer solution with: sPCLINK2NIA, iNA961, NetBIOS interface, MS-NET
sPCLINKIIBD	Five-layer solution with: sPCLINK2NIA, iNA961, NetBIOS interface
sPCLINK2TWKIT	Seven-layer thin-wire solution with: sPCLINK2, CNETXCVR, XCVRCBL
sPCLINK2DEVKIT	NetBIOS Developer's Kit with: 2-sPCLINK2TWKITs, NetBIOS and Request Block Developer's Software with iNA 961 for PCLINK2



COMPLETE OpenNET™ SOLUTION FOR THE VAX

VAX/VMS Networking software (VMSNET) provides the OpenNET connection for a VAX* or MicroVAX II* system to iRMX, XENIX, DOS, UNIX System V and iNRM systems.

VMSNET enables a MicroVAX or VAX system to act as an OpenNET file server system allowing any OpenNET consumer node transparent file access to files on the MicroVAX or VAX.

The VMSNET product includes one of two types of Ethernet controller boards: a UNIBUS* board for the VAX or a Q-bus* board for the MicroVAX.

VMSNET software performs the OpenNET functions via an implementation of the Network File Access (NFA) file server protocols, VMS consumer bi-directional file transfer utilities, and Intel's iNA 960 transport layer software running on the supplied intelligent LAN controllers.

A set of network management utilities provide (Micro)VAX users with information and statistics about VMSNET activities.

FILE ACCESS

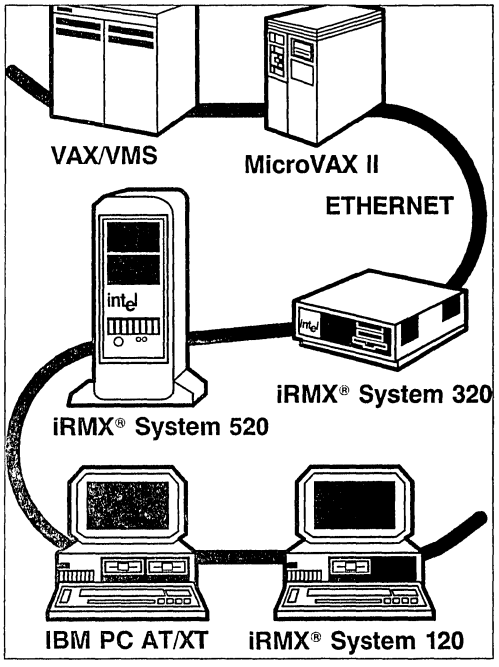
- Transparent file access between a VAX/VMS server and DOS, iRMX, XENIX, UNIX and iNDX systems
- DECnet compatibility: consumer nodes may access remote files using VMS logical names over DECnet (no file locking or compatibility mode opens)

VIRTUAL TERMINAL (VT)

- OpenNET/MS-NET VT protocols supported
- Both Server and Consumer functionality supported

HOST REQUIREMENTS

- VAX 750, 780, 782, 785
- VAX 8200, 8250, 8500, 8530, 8600, 8650
- MicroVAX II
- (Micro)VMS versions 4.5-5.0



HARDWARE FEATURES

- 80186/82586-based LAN Boards
- Unibus power requirements: +5 VDC at 4.5 amps, 6 amps maximum, −15 VDC at .5 amps, 3 amp surge
- Qbus power requirements: +5 VDC at 6 amps, 6 amps maximum.
- Internal cables, mounting hardware and user manuals are included.

Hardware installation and service contracts should be arranged with Digital Equipment Service Personnel.

PRODUCT CODES

VMSNET	VAX/VMS Networking Software for VAX family
MVMSNET	VAX/VMS Networking Software for MicroVAX II

COMPLETE OpenNET™ SOLUTION FOR UNIX SYSTEM V

SV-OpenNET connects Intel SYSTEM V/386 systems with all the OpenNET nodes. SV-OpenNET is available for MULTIBUS I and MULTIBUS II. The product includes a complete solution: communications board, Mail, VT, print spooling, nameserver interface library (NSI), and network management.

SV-OpenNET allows application interfacing through the UNIX TLI library. Applications may also access SV-OpenNET via the higher-level NSI library. SV-OpenNET can also coexist with the UNIX network, RFS.

FEATURES

Server Message Block (SMB) Transparent File Access

- Transparent file access between DOS, iRMX, XENIX, UNIX and iNDX systems.
- Both Server and Consumer functionality supported
- Remote Batch Execution (RBE) through "rexec"

NETWORK ADMINISTRATION AND MANAGEMENT

- Compatible with XENIX-NET
- File-based Nameserver compatible with XENIX-NET
  /net/data files

MAIL

- Supports MMDF
- Interoperates with XENIX-NET mail

VIRTUAL TERMINAL (VT)

- OpenNET/MS-NET VT protocols supported
- Both Server and Consumer functionality supported

PRINT SPOOLING

- Interface through "rprint"
- Supports Core printer spooling protocol

UNIX STANDARD INTERFACE

- Interface via AT&T supplied TLI (Streams) library, allowing all TLI applications to interoperate with SV-OpenNET
- SV-OpenNET provides a library, NSI, for high-level Virtual Circuit (VC) creation and name to address translation. The NSI then communicates directly with the UNIX TLI

HOST REQUIREMENTS

Intel SYSTEM V.3.1 UNIX Operating System on MULTIBUS I or MULTIBUS II

PRODUCT CODES

pSVNET552AKIT	SV-OpenNET with iSBC 552A on MULTIBUS I
pSVNET530KIT	SV-OpenNET with iSBC 186/530 on MULTIBUS II

ISO/OSI CONFORMANT NETWORK SOFTWARE

Intel's MAPNET™ provides all seven layers of the industry-standard ISO/OSI specification for both Broadband (IEEE 802.4) and Ethernet (IEEE 802.3) environments.

The MAPNET software comes preconfigured or configurable to allow the OEM to change parameters as necessary. In addition, MAPNET provides multiple implementation methods (MAP on Broadband, MAP on Ethernet, and the coexistence of Broadband and Ethernet) to get started with MAP. The open software architecture allows an easy port to other operating systems and hardware.

PRECONFIGURED MAP21SXM

The preconfigured form (MAP21SXM) provides ISO/OSI Layers 3 through 7 of the MAP2.1 specification. It is preconfigured with iNA 960 to run on Intel's iSBC-554 IEEE 802.4 Token Bus MAP board to provide a seven layer solution. The preconfigured MAP21SXM software product is supplied with iRMX device drivers, user interface utilities, and the conformance tested MAPNET software.

CONFIGURABLE MAPNET21

The configurable MAPNET21 implements layers 5 through 7 of the MAP2.1 specification. MAPNET21 is designed to interface with iNA 960 and the iSBC-554 to provide a complete seven layer configurable MAP solution for OEMs.

FEATURES

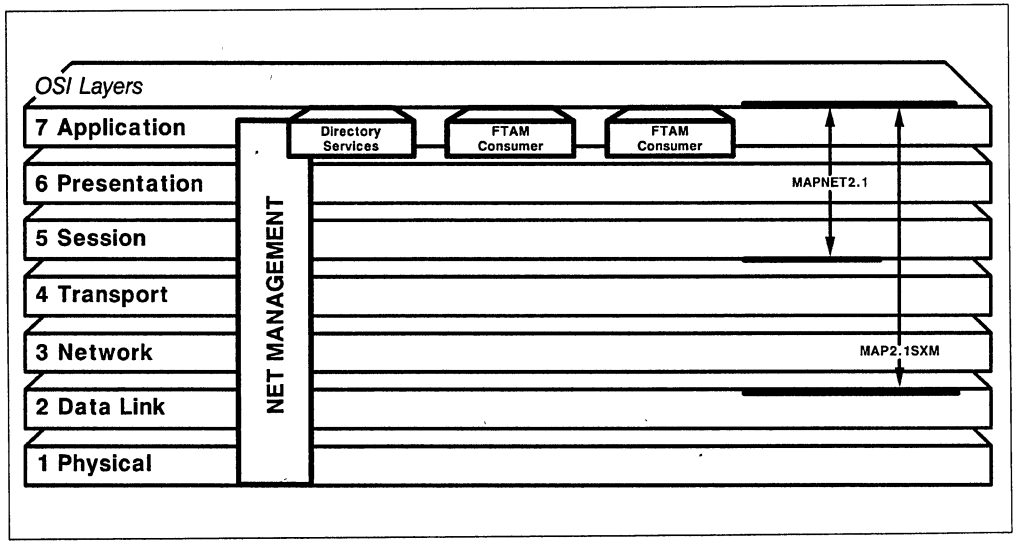
The MAPNET products provide session services, directory services, network management, FTAM, and CASE as specified in the MAP2.1 specification.

Using the services of MAPNET, users can initiate communications with other users on a MAP network, access information regarding resources available on the network, transfer files across the network, and address others by logical names rather than numbered addresses.

The Manufacturing Messaging Service (RS-511 or MMS) for MAPNET on iRMX-86 is also available from independent software vendors.

PRODUCT CODES

MAPNET21	Configurable ISO/OSI Layers 5 through 7 of the MAP2.1
MAP21SXMRO	Preconfigured ISO/OSI Layers 3 through 7 of the MAP2.1



FULLY COMPLIANT ISO/OSI TRANSPORT AND NETWORK

iNA 960 is a complete Network and Transport (ISO/OSI Layers 3 and 4) software system plus a comprehensive set of network management functions, Data Link (OSI Layer 2) drivers for IEEE 802.3 Ethernet and IEEE 802.4 Token Bus (MAP), and system environment features.

FLEXIBLE AND HIGHLY CONFIGURABLE

iNA 960 is a mature, flexible, and ready-to-use software building block for OEM suppliers of networked systems for both manufacturing and office applications (e.g., MAP and TOP).

This software is highly configurable for designs based on the 82586 and 82588 LAN controllers, 82501 and 82502 Ethernet serial interface and transceiver, and the 8086 family of microprocessors.

CONFIGURABLE AT THE OBJECT CODE LEVEL

Consisting of linkable object modules, the iNA 960 software can be configured to implement a range of capabilities and interface protocols. iNA 960 has a large installed base and has been used reliably in a variety of systems from IBM PC XT/ATs to VAX/VMS to IBM mainframes.

BASED ON INTERNATIONAL STANDARDS

Based on the ISO/OSI seven layer model for network communications, iNA 960 implements ISO 8073 Transport Class 4 providing reliable full-duplex message delivery service on top of the internet capabilities offered by the network layer. The iNA 960 network layer is an implementation of the ISO 8473 Network Class 3 Connectionless Network Protocol and supports ISO 9542 End System to Intermediate System Network Dynamic Routing. iNA 960 also supports ISO 8602 Connectionless Transport Protocol (Datagram).

PRECONFIGURED iNA 961

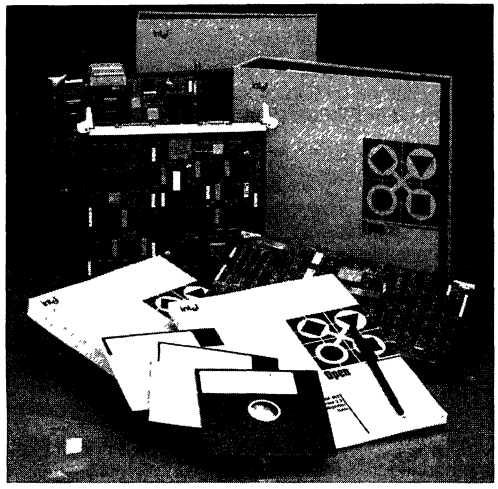
iNA 960 contains the preconfigured iNA 961 which includes support for the iSBC 552A, iSBC 186/530, and the iSBC 554.

REMOTE BOOT SERVER SUPPORT

iNA 960 provides basic boot server capabilities that will transmit predefined images to diskless network nodes that request them.

MULTI-SERVER/CONSUMER SUPPORT

iNA 960 supports the powerful MULTIBUS II feature of multiple host and communications boards. Ideal for LAN load balancing and redundant networks for fault-tolerant systems.



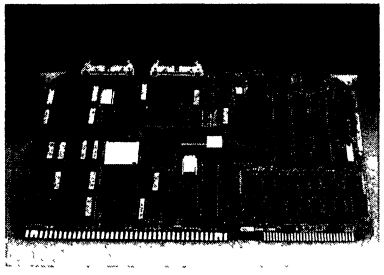
FEATURES

- Certified ISO/OSI Transport and Network Layer Software
- ISO 8072/8073 Transport Class 4
- ISO 8602 Connectionless Transport
- ISO 8348/8473 Connectionless Network
- ISO 9542 End System to Intermediate System Dynamic Routing
- Comprehensive Network Management Functions
- Remote Boot Server for diskless workstations
- Data Link Drivers for iSBC 552A, iSBX 586, iSBC 186/530, iSBC 554, and iSBC 186/51

PRODUCT CODES

INA960J	Includes iNA 961 on iRMX diskette format

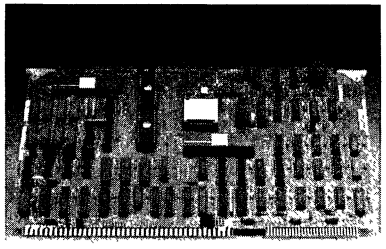
iSBC®186/51 MULTIBUS®I IEEE 802.3/ETHERNET COMMUNICATION COMPUTER



- 82586 LAN coprocessor for Ethernet/ IEEE 802.3 communication
- Two serial interfaces, RS232C and RS422A/RS449 compatible
- 6 MHz 80186 microprocessor
- 128K bytes of dual-port RAM expandable on-board to 256K bytes
- Sockets for up to 192K bytes of JEDEC 28-pin standard memory devices
- Product Code: sSBC18651

CPU . . . . . .80186 (6 MHz)
RAM (Bytes) . . . . . .128K/256K dual port
EPROM (Bytes) 192K (27256), 96K (27128), 48K (2764), 4K (2732)
Serial I/O. . . . . .2 ports, 26-pin 3M connector (RS232C/422A/449)
Ethernet I/O . . . . . .1 port, 10-pin AMP connector
Baud rates (RS232/422/449) . . . . . .75–9600 baud (async.) 1.76–38.4K baud (sync.)
Timers. 3
Interrupts. . . . . .9 levels, 28 sources
iSBX™ Connectors. 2
Software Support . . . . . .iRMX® 86 O.S., iNA 960, iRMX® -NET
Power Requirements +5V . . . . . .7.45 A
+12V. . . . . .40 mA
–12V. . . . . .40 mA

iSBC®/iSXM™ 552A MULTIBUS®I IEEE 802.3/ETHERNET NETWORK INTERFACE ADAPTER

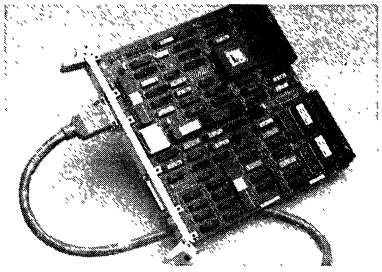


- High Performance IEEE 802.3/ Ethernet compatible network front-end processor
- Resident network software can be down-loaded over the bus or the LAN
- On-board diagnostic and boot firmware
- iSXM™ 552A version is a preconfigured controller for executing iNA 961 (ISO 8073 Transport and ISO 8473 Network software) in System 310 and 320 family products
- Product Code: pSBC552A, pSXM552A

CPU . . . . . .80186 (8 MHz)
LAN Coprocessor 82586
RAM (Bytes) 256K
EPROM (Bytes) 128K (27512), 16K (2764)
MULTIBUS® Address . . . . . .Any 64 KB boundary with a 16 MB address space
Software Support . . . . . .iNA 960/961
Power Requirements +5V . . . . . .6.2 A
+12V. . . . . .0.5 A

MULTIBUS® II OpenNET™ NETWORKING HARDWARE

iSBC®186/530 MULTIBUS®II IEEE 802.3/ETHERNET NETWORK INTERFACE ADAPTER

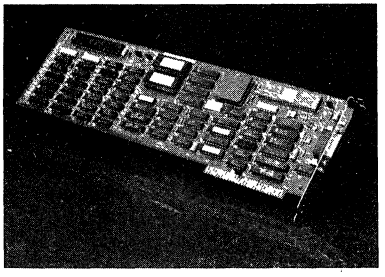


- Provides Ethernet® (IEEE 802.3) compatible networking capability for all MULTIBUS®II systems
- MULTIBUS®II iPSB (Parallel System Bus) interface with full Message Passing capability
- Resident firmware to support Built-in-Self-Test (BIST) power-up diagnostics, and host-to-controller software download
- Four 28-pin JEDEC sites, expandable to 8 sites with iSBC® 341 MULTIMODULE® for a maximum of 512K bytes of EPROM
- Provides one RS232C serial port for use in debug and testing
- Product Code: pSBC186530

CPU		.80186 (8 MHz)
DRAM		.512 KB on-board
EPROM		.4 28-pin JEDEC sites, up to 256 KB (max.) using 27512 devices, up to 512 KB (total) using iSBC® 341 module
Ethernet I/O		.1 channel 15-pin connector
	Controller	.82586 LAN Coprocessor
Serial I/O		.1 channel RS232C, 15-pin connector
	Controller	.8031
	Leads Supported	.TD, RD
Timers		.3
Interrupts		.5 levels with 5 on-board sources and 255 sources from iPSB Bus
Power Requirements	+5V	.8.8 A (excludes power for user-installed memory devices)
	+12V	.50 mA
	−12V	.50 mA

PC BUS OpenNET™ NETWORKING HARDWARE

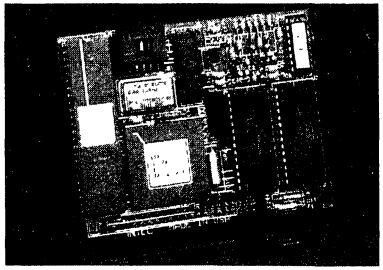
PC LINK2 NETWORK INTERFACE ADAPTER (PC LINK2 NIA)



CPU		.80186 (8 MHz)
LAN Communications Controller		.82586
Ethernet Interface		.15-pin connector, 82501 serial interface
DRAM		256 KB (dual-port), 0 wait-state memory access by the CPU
EPROM		.16 KB
Size		.4.15 in H×13.3 in W
Power Requirements	+5V	.2.0 A
	+12V	.0.5 A

- Intelligent high performance hardware with on-board microprocessor, 16K bytes EPROM and 256K bytes RAM.
- Full slot PC AT, PC XT (or compatible computer system) board
- 80186 microprocessor, 82586 LAN coprocessor, 8 MHz zero-wait-state memory access.
- RAM shared by the PC host and PC Link2 board via an 8K memory window.
- Jumper selection for Ethernet or IEEE 802.3.
- Effective self diagnostics.
- Product Code: sPCLINK2NIA

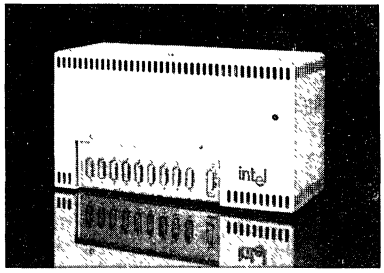
iSBX™ 586 DATA LINK ENGINE MULTIMODULE®BOARD



LAN Coprocessor . . . . . .82586 (8 MHz)
RAM (Bytes) . . . . . 16K (dual-port)
Software Support . . . . . .iNA 960/961
Power Requirements +5V . . . . . .2.0 A
+12V . . . . . .1.0 A

- Provides an IEEE 802.3/Ethernet compatible connection for 8086 and 80186-based host boards over a 16-bit iSBX™ interface
- Single-wide iSBX™ MULTIMODULE™
- Compatible with iNA 960/961 ISO 8073 Transport and ISO 8473 Network software
- Provides an IEEE 802.3 to IEEE 802.4 Router capability when used with the iSBC® 554 IEEE 802.4 LAN controller
- Product Code: sSBX586

iDCM 911-1 INTELLINK™ FAN-OUT UNIT



Size . . . . . .14 in. W × 7.8 in. H × 5.5 in. D
Power Requirements . . . . . .100/120/220/240 VAC, 47–64 Hz

- Connects up to nine Ethernet compatible workstations without the need for transceivers or coaxial cable
- Connects directly to the Ethernet coaxial cable through a standard transceiver cable
- Cascadable to support 17–81 workstations
- Product Code: pDCM9111

ETHERNET/IEEE 802.3 THIN-WIRE TRANSCEIVER

Size . . . . . .2.8 in. × 3.6 in. × 3.8 in.
Power Requirements +12V. . . . . .375 mA
(from transceiver cable)

- Die-cast metal case for protection, reduced EMI, and efficient heat dissipation
- Low inrush current at power-up, auto shutdown when low-input voltage occurs, and surge protection
- IEEE 802.3-compliant, Ethernet V1.0/ V2.0 compatible
- Three LEDs monitor power status, packet collisions and signal quality
- Removable BNC type cable tap
- User-configurable for use with or without heartbeat
- Product Code: CNETXCVR

OpenNET™ NETWORKING ORDERING INFORMATION

Code	Description

iRMX-NET OpenNET PRODUCTS

RMXINETSW	iRMX-NET for iRMX 86 operating system
RMXIINETSW	iRMX-NET for iRMX II operating system on MB-I, MB-II, and System 120
sSXM120NETKIT	Preconfigured iRMX-NET and sPCLINK2 for networking iRMX and DOS on System 120

PCLINK2 OpenNET PRODUCTS

sPCLINK2NIA*	PC Link2 Network Interface Adapter Hardware only
sPCLINK2*	Seven-layer Solution with sPCLINK2NIA, iNA961, NetBIOS interface, MS-NET
sPCLINKIIBD*	Five-layer Solution with sPCLINK2NIA, iNA961, NetBIOS interface
sPCLINK2TWKIT*	Seven-layer Thin-wire Solution with sPCLINK2, CNETXCVR, XCVRCBL
sPCLINK2DEVKIT*	NetBIOS Developer's Kit with 2-sPCLINK2TWKITs, NetBIOS programmer kit
PCLDOSRBIRO	Request Block Developer's Software with iNA 961 for PCLINK2

VAX/VMS OpenNET PRODUCTS

VMSNET	Networking Software for VAX family
MVMSNET	VAX/VMS Networking Software for MicroVAX II

UNIX SYSTEM V OpenNET PRODUCTS

pSVNET552AKIT*	UNIX SV-OpenNET with iSBC 552A on MULTIBUS I
pSVNET530KIT*	UNIX SV-OpenNET with iSBC 186/530 on MULTIBUS II

MAP/TOP OpenNET PRODUCTS

MAPNET21	Configurable ISO/OSI Layers 5 through 7 of the MAP2.1
MAPNET21RF	Royalty fee for MAPNET21
MAP21SXMRO	Preconfigured ISO/OSI Layers 3 through 7 of the MAP2.1, includes license
MAP21SXMRF	Royalty fee for MAP21SXM
SBC5541	iSBC554-1 MULTIBUS I MAP Communications Engine, Xmit: CH 3′, 4′ Rcv: CH P, Q
SBC5543	iSBC554-3 MULTIBUS I MAP Communications Engine, Xmit: CH6′, FM1′ Rcv: CH T, U

iNA 960 OpenNET ISO/OSI PRODUCTS

INA960J	ISO/OSI Transport and Network layers, includes iNA961
INA960RF	Royalty fee for INA960

MULTIBUS I AND MULTIBUS II IEEE 802.3/ETHERNET PRODUCTS

sSBC18651*	iSBC 186/51 MULTIBUS I IEEE 802.3/Ethernet Communication Computer
pSBC552A*	iSBC 552A MULTIBUS I IEEE 802.3/Ethernet Network Interface Adapter
pSXM552A*	iSBC 552A preconfigured for Intel System 310 and 320, includes iNA 961 royalty
pSBC186530*	iSBC 186/530 MULTIBUS II IEEE 802.3/Ethernet Network Interface Adapter

OpenNET NETWORKING ACCESSORIES

sSBX586*	iSBX 586 MULTIMODULE IEEE 802.3/Ethernet Data Link board
pDCM9111	iDCM 911-1 Intellink Fan-out Unit
CNETXCVR	Thin-wire transceiver. Requires transceiver cable (XCVRCBL)
XCVRCBL	Ten-foot transceiver cable

Code	Description
iRMX-NET OpenNET PRODUCTS	
462040-001	iRMX-NET Software Release 3.0 Installation and Configuration Guide
462041-001	iRMX-NET Software Release 3.0 User's Guide
PCLINK2 OpenNET PRODUCTS	
460665-001	MS-NET User's Guide
450772-001	PCLINK2 Hardware Reference Manual
462305-001	PCLINK2 NIA Hardware Installation Guide
462311-001	PCLINK2 Software Developer's Manual
462308-001	PCLINK R3.0 Software For DOS—Installation Guide
VAX/VMS OpenNET PRODUCTS	
480071-001	VAX/VMS OpenNET User's Manual
UNIX SYSTEM V OpenNET PRODUCTS	
462740-001	SV-OpenNET User's Manual
462741-001	SV-OpenNET Installation and Administration Guide
MAP/TOP OpenNET PRODUCTS	
461298-001	MAPNET User's Guide
454209-001	iSBC 554 Network Interface Adapter Hardware Reference Manual
460432-001	MAP Broadband Starter Kit Guide
iNA 960 OpenNET ISO/OSI PRODUCTS	
462250-001	iNA 960 R3 Programmer's Reference Manual
462252-001	iNA 960 R3 Installation and Configuration Guide
MULTIBUS I AND MULTIBUS II IEEE 802.3/ETHERNET PRODUCTS	
122330-001	iSBC 186/51 COMMputer Board Hardware Reference Manual
149228-001	iSBC 552A IEEE 802.3 Communications Controller User's Guide
149226-002	iSBC 186/530 Network Interface Adapter User's Guide
OpenNET NETWORKING ACCESSORIES	
122290-001	iSBX 586 MULTIMODULE Ethernet Communication Controller Hardware Reference Manual
122074-002	iDCM 911-1 Intellink Cluster Module Reference Manual
280665-001	Ethernet/IEEE 802.3 Thin-wire Transceiver Factsheet

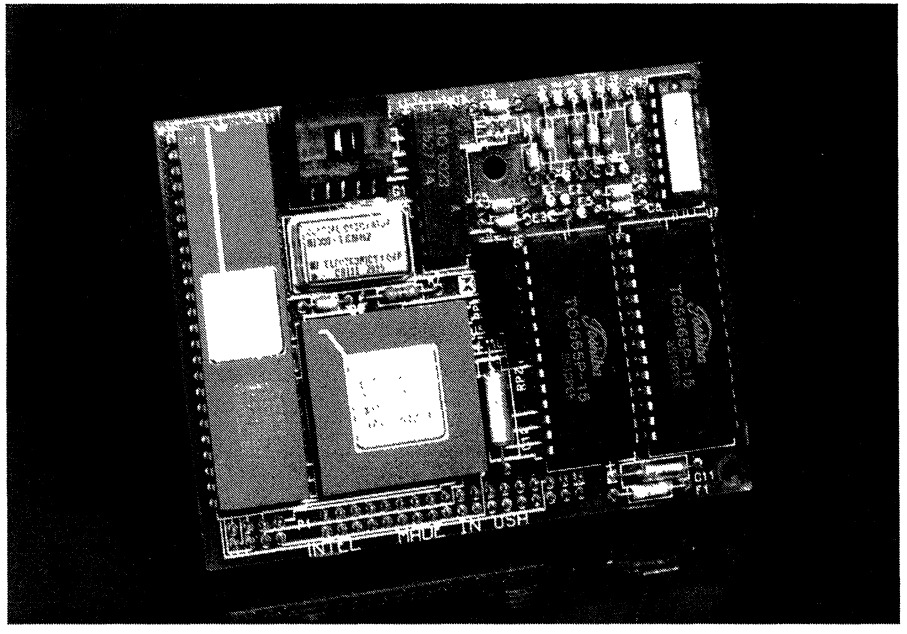
*UNIX is a registered trademark of AT&T
MS-DOS, XENIX are trademarks of Microsoft Corporation
DECnet, VAX/VMS, MicroVAX, UNIBUS, Q-bus are trademarks of Digital Equipment Corporation
sSXM120NETKIT, sPCLINK2NIA, sPCLINK2, sPCLINKIIBD, sPCLINK2TWKIT, sPCLINK2DEVKIT, sSBC18651, and sSBX586 are manufactured by Intel Singapore, Ltd.
pSVNET552AKIT, pSVNET530KIT, pSBC552A, pSXM552A, pSBC186530, and pDCM9111 are manufactured by Intel Puerto Rico, Inc.

iSBX™ 586
Ethernet Data Link Engine

- **Provides an IEEE 802.3 (Ethernet) Data Link Connection for Host Boards with 16-Bit SBX Bus Capability**
- **Based on Intel's 8 MHz 82586 LAN Coprocessor Chip which Includes the following Features:**
  - **— Automatic Retransmission**
  - **— On-Board Multicast Address Filtering**
  - **— Host Interface via Buffer Chaining**
- **16 Kbytes of Local Dual-Ported Buffer RAM**
- **Single-Wide iSBX™ Board That Conforms to IEEE 959 Specifications**
- **Compatible with iNA 960 ISO Transport Layer Software**
  - **— Direct Support for iRMX® Operating Systems**
  - **— Source Code Support for Other Operating Systems**

The iSBX™ 586 Ethernet Data Link Engine is a single wide iSBX sized card that provides a low cost Ethernet controller for single board computers with 16-bit SBX bus capability. Based on the 82586 Local Area Network Coprocessor, the iSBX 586 implements the data link (Layer 2) and physical (Layer 1) layers of the International Standards Organization (ISO) Open Systems Interconnect (OSI) Reference Model. This allows the iSBX 86 to supply an IEEE 802.3 10 Mbps (Ethernet) connection for a single board computer having a 16-bit SBX bus connector.

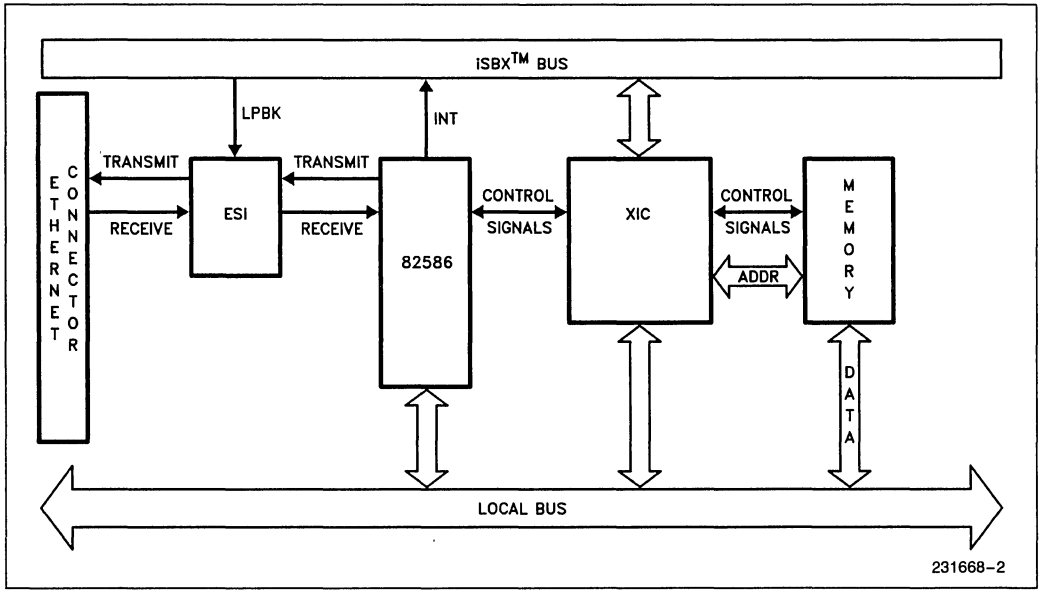
The iSBX 586 MULTIMODULE is a low cost building block that can implement an Ethernet connection at various levels of integration. One application for the iSBX 586 is as a "best effort" datagram message delivery engine. In conjunction with the host single board computer running iNA 960 R2.0 ISO Transport Software, the iSBX 586 can allow for a four-layer, OpenNET compatible solution for Ethernet connections.



231668-1

November 1989
Order Number: 231668-001

BLOCK DIAGRAM



iSBX™ 586 Block Diagram

SPECIFICATIONS

SBX Interface	IEEE 959
Data Transfer	16 bits
Signal Levels	See the iSBX 586 Hardware Reference Manual
Signals Supported	All iSBX bus signals are supported except: MA2 MINTR1 MCLK OPT1 MDACK TDMA MDRQT −12V
Serial Interface	IEEE 802.3 compatible
DC Power Requirements	All voltages supplied by the iSBX interface +5V DC ±5%, 2A max. +12V DC ±5%, 1A max.

ENVIRONMENTAL

Temperature: 0°C to 55°C Operating (Free moving air across the base board and iSBX 586)
−40°C to +65°C Non-Operating

Humidity: 5% to 90% Operating
5% to 95% Non-Operating

Refer to the iSBX™ 586 Hardware Reference Manual (not supplied) for details.

ORDERING INFORMATION

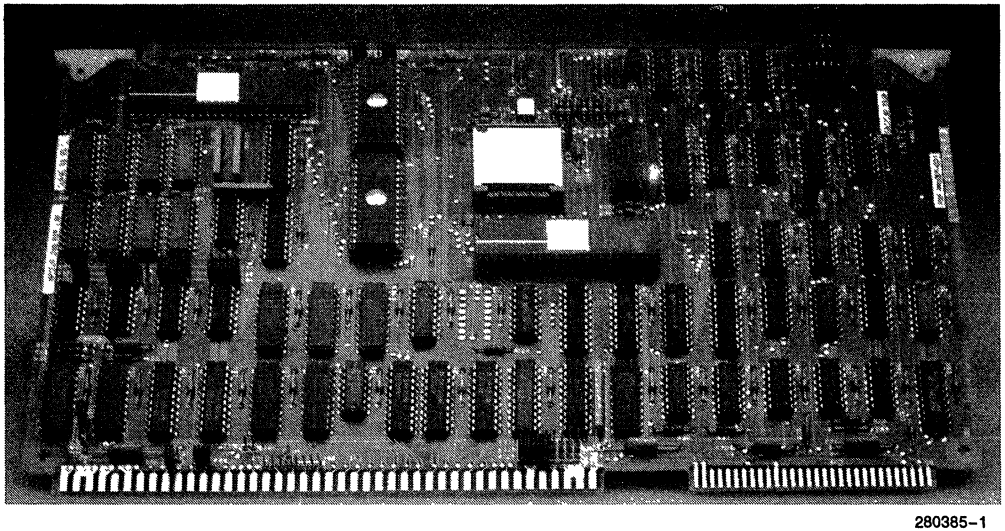
Order Code	Description
SBX 586	Ethernet Data Link Engine

iSBC® 552A AND iSXM™ 552A IEEE 802.3 COMPATIBLE NETWORK INTERFACE ADAPTORS

- **Provides High-Performance Network Front-End Processing for All MULTIBUS® I Systems Regardless of the Operating System of the Host**
  - **— Intelligent Controller with an 8 MHz 80186 Processor and 256K of DRAM Memory**
  - **— IEEE 802.3 Network Port Driven by the 82586 LAN Coprocessor**
- **Can Execute On-Board the Intel iNA 960/961 Software, an Implementation of Industry Standard ISO 8073 Transport and ISO 8473 Network Protocols**
- **Resident Network Software Can be Down-Loaded over the Bus or LAN**
- **On-Board Diagnositc and Boot Firmware**
- **Supported by XNX-NET and RMX-NET Network File Service Software Products**
- **Available in Two Versions**
  - **— iSBC 552A is a Flexible, Intelligent Communications Controller for IEEE 802.3 LANs**
  - **— iSXM™ 552A is a Preconfigured Controller for Executing iNA 961 Transport and Network Software as a Fully Qualified System Extension Module for the System 310 Family Products**

The iSBC 552A and iSXM 552A products are designed for communications front end processor applications connecting MULTIBUS I systems onto IEEE 802.3 compatible LANs. These Network Interface Adaptors are dedicated to the network communications tasks within a system allowing the host to spend more time processing user applications. A major advantage of these products is that they can be used to network existing systems and established designs without forcing the redesign of the entire system architecture.

The iSBC and iSXM 552A boards can be used with any operating system because they require only a high level interface to communicate with the host (e.g. transport commands in the case of the iSXM 552A board). The result is a powerful system building block which enables the OEM to network MULTIBUS I based systems with different operating systems. Applications for the 552A products include networked multiuser XENIX based systems for the office and laboratory, iRMX-based systems for real-time applications, or many other system applications.



280385-1

October 1989
Order Number: 280385-001

THE iSBC® BOARD vs THE iSXM™ BOARD

The iSBC 552A version is a board that offers the hardware necessary for the user to construct an IEEE 802.3 front-end processor for custom requirements. The Intel iNA 960 ISO standard transport and network software can be configured and optimized to run on the iSBC 552A board.

The iSXM 552A version is a product that is preconfigured for Intel's family of System 310 products, includes the necessary internal system cabling, and is fully qualified to run in System 310 products. The iSXM 552A board supports the iNA 961 ISO standard transport and network software with no configuration activities required of the customer. iSXM 552A board customers receive the iNA 961 software through a separate purchase of a software license.

Operating Systems Environment

The iSBC/iSXM 552A board and iNA 960/961 software can function in any MULTIBUS I environment. The communication between the iSBC/iSXM 552A and the host processor is entirely independent of any host operating systems. iNA 960/961 uses the MIP protocol to interface with the host processor. The MIP is a reliable, host operating system independent, process to process communication scheme between any processors on the MULTIBUS I System Bus. iNA 960/961 can service multiple processes utilizing its services at the same time.

A host processor passes iNA 960/961 commands and buffers in the MULTIBUS I system memory to the iNA 960/961 software. This software is responsible for updating the response fields of these commands. It is responsible for copying the user send buffer in MULTIBUS I system memory into its onboard buffers for transmission and for copying received messages to user buffers in MULTIBUS I system memory.

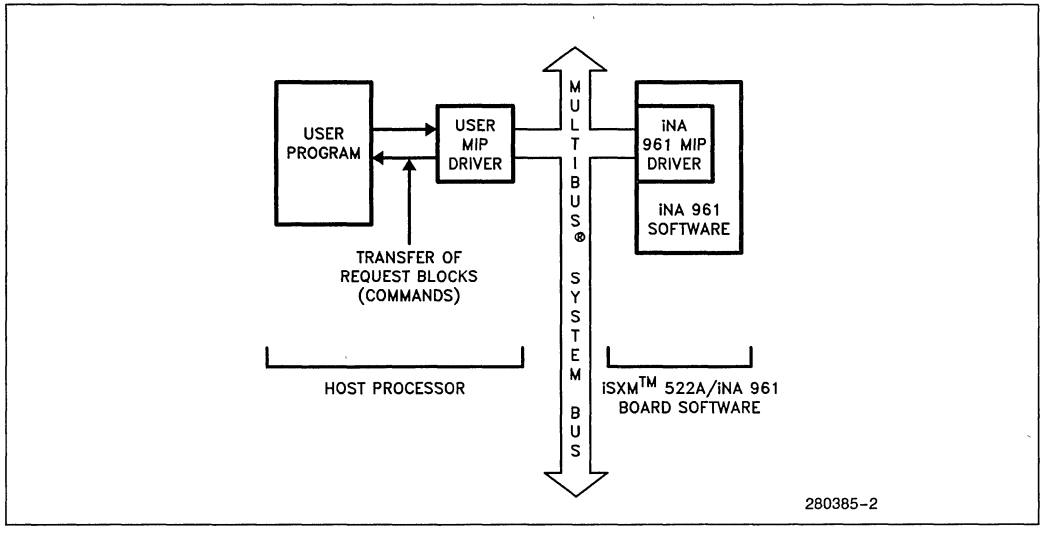


Figure 1. iNA 961 MIP Interface

DEVELOPMENT ENVIRONMENT

The iSXM 552A board is a complete system product that allows a user to emphasize the development of high level software, such as a network file server. The iSXM 552A board and the iNA 961 software together form a transport that integrates into any MULTIBUS I system. iNA 961 is supplied in a boot loadable file format. This file can be loaded into the iSXM 552A by a host processor or through a remote boot server network node. The boot firmware on the iSXM 552A supports both functions. In order to remote boot the host system, appropriate host processor firmware and software is required.

The iSBC 552A allows a user to fine tune iNA 960 and to put the software on the board. Both iNA 960 and the iSBC 552A can be flexibly configured to best meet the users' requirements. An Intel development system, together with the Intel I2ICE™ system or equivalent product can be used if the user desires to do extensive development work on the iSBC 552A. Intel also supplies a wide range of host processor boards and systems (such as the iSBC 286/12 and system 310) that will function well both with the iSBC 552A or the iSXM 552A board.

SPECIFICATIONS

Data Transfer: 8 or 16 bits

Average Raw MULTIBUS I Transfer Rate:

8.7 Mbits/second (450 ns, 16-bit system memory and no MULTIBUS I contention)

Transceiver Interface

Transmit Data Rate:	10 Mbits/second
Signal Levels:	Series 10,000 ECL-compatible
Host Interrupts:	One MULTIBUS I non-vector interrupt for use in system/host handshaking
MULTIBUS Interface:	The iSBC/iSXM 552A board conforms to all AC and DC requirements outlined in Intel MULTIBUS I Specification. Order Number 142686-022m except for the following signals: Signal DAT0–DAT7 Signal Specification: I_{IL} = 180 μA I_{IH} = 125 μA
DC Power Required:	All voltages supplied by the MULTIBUS I Interface +5.0V ±5%, 6.2A maximum +12.0V ±5%, 0.5 maximum

Environmental

Temperature:	0°C to +55°CC Operating
	−40°C to −65°C Non-Operating
Humidity:	5% to 90% Operating
	5% to 95% Non-Operating

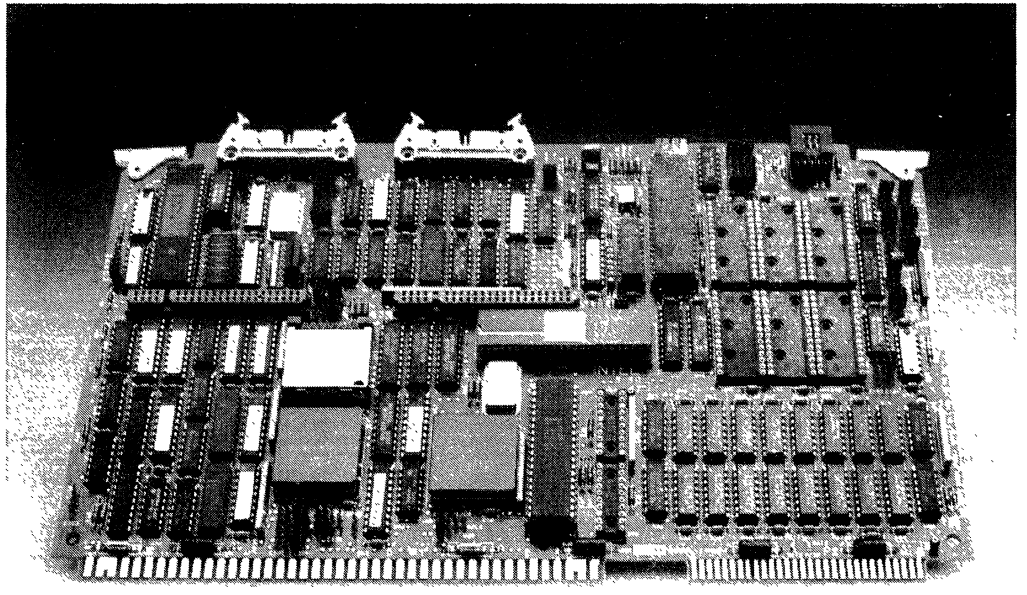
ORDERING INFORMATION

Order Code	Description
SBC552A	IEEE 802.3 COMMengine
SXM552A	IEEE 802.3 Transport Engine for iNA961 and SYP310 systems
NA960	Configurable transport software usable with the SBC552A
NA961	Preconfigured transport software for the SXM552A

iSBC® 186/51 COMMUNICATING COMPUTER

- 6 MHz 80186 Microprocessor
- 128K Bytes of Dual-Ported RAM Expandable On-Board to 256K Bytes
- 82586 Local Area Network Coprocessor for Ethernet/IEEE 802.3 Specifications
- Two Serial Interfaces, RS-232C and RS-422A/RS-449 Compatible
- Supports Transport Layer Software (iNA 960) and Higher Layer Communications Software (such as iRMX®-NET)
- Sockets for up to 192K Bytes of JEDEC 28 Pin Standard Memory Devices
- Two iSBX™ Bus Connectors
- 16M Bytes Address Range of MULTIBUS® Memory
- MULTIBUS Interface for Multimaster Configurations and System Expansion
- Supported by a Complete Family of Single Board Computers, Peripheral Controllers, Digital and Analog I/O, Memory, Packaging and Software

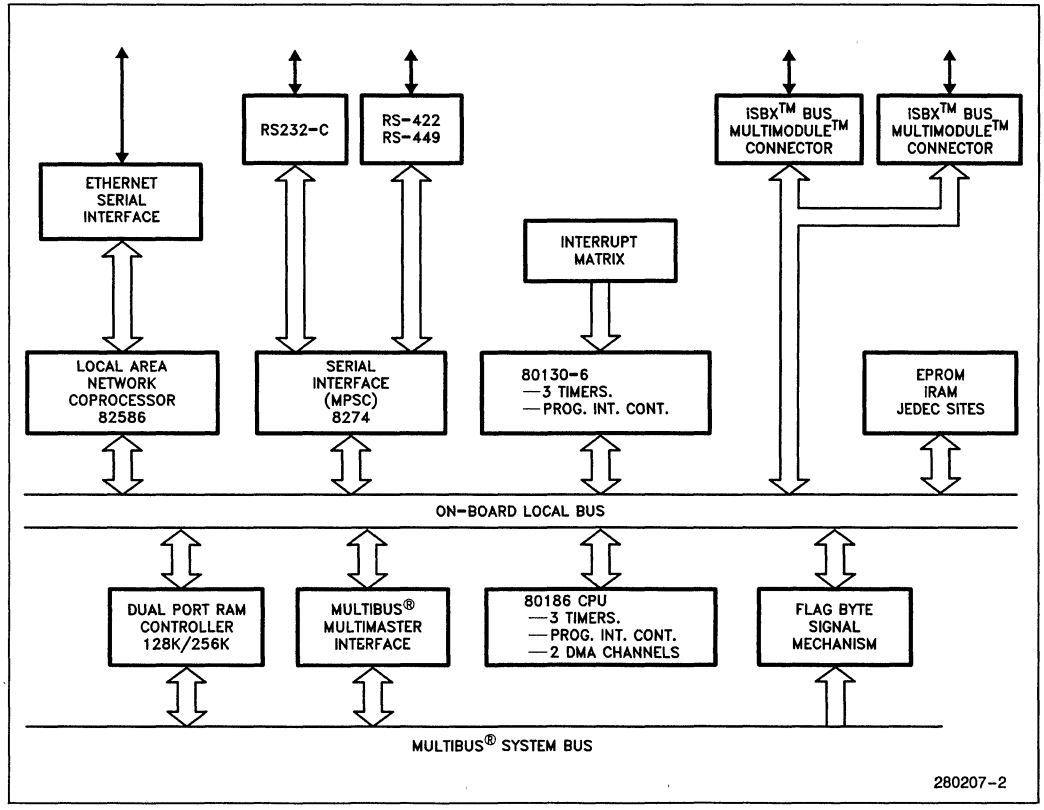
The iSBC® 186/51 COMMUNICATING COMPUTER is a member of Intel's OpenNET™ family of products, and supports Intel's network software. The iSBC 186/51 utilizes Intel's VLSI technology to provide an economical self-contained computer for applications in processing and local area network control. The combination of the 80186 Central Processing Unit and the 82586 Local Area Network Coprocessor makes it ideal for applications which require both communication and processing capabilities such as networked workstations, factory automation, office automation, communications servers, and many others. The CPU, Ethernet interface, serial communications interface, 128K Bytes of RAM, up to 192K Bytes of ROM, I/O ports and drivers and the MULTIBUS interface all reside on a single 6.75″ x 12.00″ printed circuit board.



280207-1

October 1989
Order Number: 280207-001

BLOCK DIAGRAM



iSBC® 186/51 Block Diagram

SPECIFICATIONS

Word Size

Instruction: 8, 16, 24, or 32 bits
Data: 8, 16 bits

System Clock

6.00 MHz ± 0.1%

Cycle Time

Basic Instruction Cycle

6 MHz— 1000 ns
333 ns (assumes instruction in the queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles).

Memory Capacity/Addressing

Six Universal Memory Sites support JEDEC 24/28 pin EPROM, PROM, iRAM and static RAM.

Example for EPROM:

Device	Total Capacity	Address Range
2732	24K Bytes	F8000–FFFFF$_H$
2764	48K Bytes	F0000–FFFFF$_H$
27128	96K Bytes	E0000–FFFFF$_H$
27256	192K Bytes	C0000–FFFFF$_H$

On-Board RAM

Board	Total Capacity	Address Range
iSBC 186/51	128K Bytes	0–1FFFF$_H$

With MULTIMODULE™ RAM

Board	Total Capacity	Address Range
iSBC 304	256K Bytes	0–3FFFF$_H$

I/O Capacity

Serial two programmable channels using one 8274.

SBX Bus two 8/16-bit SBX bus connectors allow use of up to 2 single-wide modules or 1 single-wide module and 1 double-wide SBX module.

Serial Communications Characteristics

Synchronous — 5–8 bit characters; internal or external character synchronization; automatic sync insertion

Asynchronous — 5–8 bit characters; break character after generation; 1, ½, or 2 stop bits; false start bit detection

Baud Rates

Frequency (KHz) (S/W Selectable)	Baud Rate (Hz)		
	Synchronous	Asynchronous	
	÷1	÷16	÷64
153.6		9600	2400
76.8		4800	1200
38.4	38,400	2400	600
19.2	19,200	1200	300
9.6	9,600	600	150
4.8	4,800	300	75
2.4	2,400	150	
1.76	1,760	110	2400

NOTE:
Frequency selected by I/O write of appropriate 16-bit frequency factor to baud rate register (80186 timer 0 and 80130 baud timer).

Timers

Input Frequencies

Reference 1.5 MHz ±0.1% (0.5 μs period nominal)
Event Rate: 1.5 MHz max.

Interfaces

Ethernet — IEEE 802.3 compatible

MULTIBUS® — IEEE 796 compatible

MULTIBUS® — Master D16 M24 I16 V0 EL

Compliance

iSBX™ Bus — IEEE P959 compatible

Serial I/O — RS-232C compatible, configurable as a data set or data terminal,
RS-422A/RS-449

Physical Characteristics

Width: 12.00 in. (30.48 cm)

Height: 6.75 in. (17.15 cm)

Depth: 0.70 in. (1.78 cm)

Weight: 18.7 ounces (531 g.)

Environmental Characteristics

Operating Temperature: 0°C to 55°C

Relative Humidity: 10% to 90% (without condensation)

80186 Output Frequencies/Timing Intervals

Function	Single Timer/Counter		Dual (Cascaded) Timer/Counter	
	Min	Max	Min	Max
Real-Time Interrupt	667 ns	43.69 ms	667 ns	47.72 Minutes
Programmable One-Shot	1000 ns	43.69 ms	1000 ns	47.72 Minutes
Rate Generator	22.889 Hz	1.5 MHz	0.0003492 Hz	1.5 MHz
Square-Wave Rate Generator	22.889 Hz	1.5 MHz	0.0003492 Hz	1.5 MHz
Software Triggered Strobe	1000 ns	43.69 ms	1000 ns	47.72 Minutes
Event Counter		1.5 MHz		

Connectors

Interface	Double-Sided Pins	Centers (In.)	Mating Connectors
Ethernet	10	0.1	AMP87531-5
Serial I/O	26	0.1	3M 3452-0001 Flat or AMP88106-1 Flat

Electrical Characteristics

D.C. Power Supply Requirements

Configuration	Maximum Current (All Voltages ± 5%)		
	+5	+12	−12
SBC 186/51 as shipped:			
Board Total	7.45A	40 mA	40 mA
With separate battery back-up	6.30A	40 mA	40 mA
Battery back-up	1.15A	—	—
With SBC-304 Memory Module Installed:			
Board Total	7.55A	40 mA	40 mA
With separate battery back-up	6.30A	40 mA	40 mA
Battery back-up	1.25A	—	—

NOTES:
1. Add 150 mA to 5V current for each device installed in the 6 available Universal Memory Sites.
2. Add 500 mA to 12V current if Ethernet transceiver is connected.
3. Add additional currents for any SBX modules installed.

Reference Manual

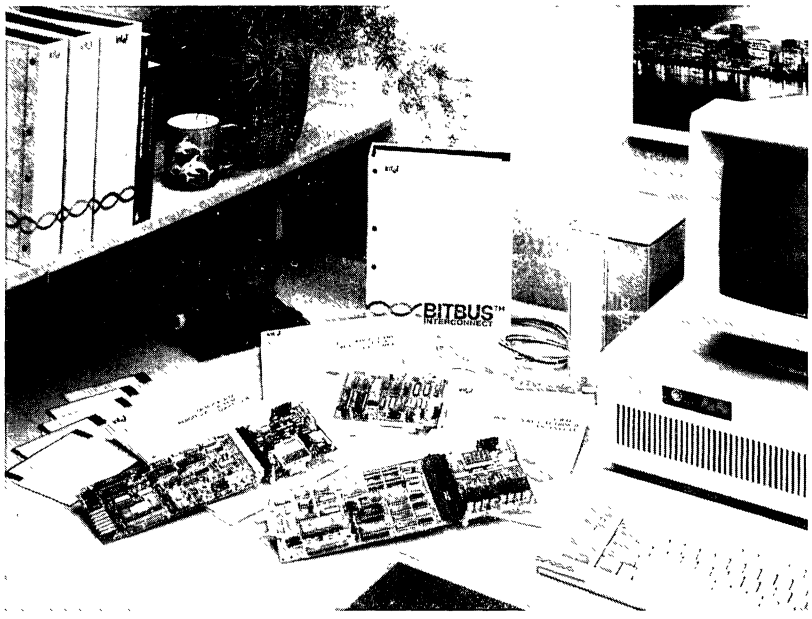
122330— iSBC 186/51 Hardware Reference Manual (NOT SUPPLIED)

Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

ORDERING INFORMATION

Order Code	Description
SBC18651	Communicating Computer

Distributed Control Modules 17



A COMPLETE BITBUS™ NETWORK YOU CAN HAVE UP AND RUNNING IN TWO HOURS

The BITBUS™ Starter Kit is a complete hardware/software kit containing BITBUS analog and digital boards, tailored application software, and all the accessories (e.g., power supply and cables) required to set up a simple but functional BITBUS network. A first-time user can construct a BITBUS network and execute sample application programs within two hours of opening the box. He can then incorporate this basic network into his own distributed control application.

FEATURES:

- Self-contained BITBUS kit requiring only an IBM PC or compatible host.
- BITBUS analog and digital boards, plus PC Gateway to the BITBUS network
- Sample application software with built-in installation, configuration, and diagnostic software.
- No BITBUS experience necessary.

September, 1989
Order Number 280638-001

REAL-TIME INTERCONNECT FOR DISTRIBUTED CONTROL

The Intel BITBUS network provides the optimal solution for building real-time distributed control systems. The BITBUS serial bus architecture overcomes many of the limitations inherent in traditional industrial connection methods to give you increased performance, reliability, and flexibility and lower implementation costs.

DESIGNED FOR FIRST-TIME BITBUS USERS

The BITBUS Starter Kit is the ideal way for first-time BITBUS users to learn about the BITBUS architecture. Shortly after unpacking this kit, you can be confidently executing your first BITBUS distributed control application.

EVERYTHING YOU NEED

Based on standard Intel products, the BITBUS Starter Kit includes the BITBUS analog board, the BITBUS digital board, the PC Gateway into the BITBUS network, power supply and cables. Supporting the standard product are demonstration boards that the user can manipulate to display analog or digital functionality.

BUILT-IN INSTALLATION AND DIAGNOSTIC SOFTWARE

Application software included with the BITBUS Starter Kit provides network setup information, as well as comprehensive error-checking software to verify that the network is configured correctly. If there is an error, the software directs you to the problem and suggests a correction. Once the network is working properly, the software steps you through optional configurations, from a host-based centralized control system to a node-based distributed control scheme. Each configuration allows you to interact with the network.

SERVICE, SUPPORT AND TRAINING

Intel provides worldwide support for repair, on-site service, network design, and installation. Development support options include phone support, subscription service, on-site consulting, and customer training.

INTEL QUALITY AND RELIABILITY

The BITBUS Starter Kit is designed and manufactured in accordance with Intel's high quality standards. We then verify quality through rigorous testing in our state-of-the-art Environmental Test Laboratory.

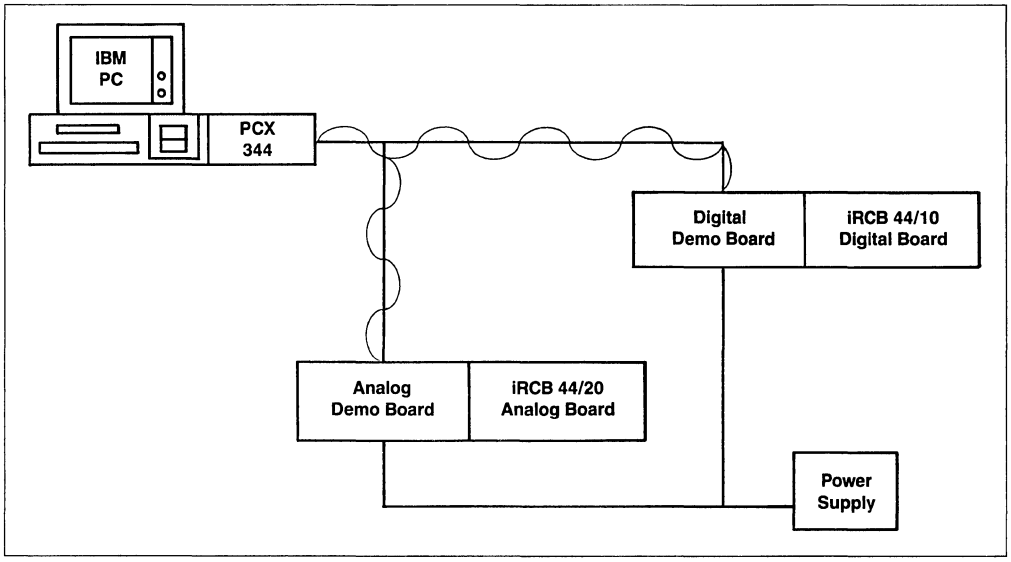


Figure 1: BITBUS Starter Kit

ORDERING INFORMATION

Product Code	Qty	Product Content
BITBUSKIT		Basic Starter Kit which includes the following items:
	1	BITBUS Starter Kit User's Guide
		HARDWARE
	1	iPCX 344A – BITBUS IBM PC Interface Board – PC Gateway to BITBUS Network
	1	iRCB 4410A – BITBUS Digital I/O Remote Controller Board
	1	iRCB 4420A – BITBUS Analog I/O Remote Controller Board
	1	Digital Demonstration Board
	1	Analog Demonstration Board
	1	Power Supply, 25 Watt, UL, VDE, CSA approved
	1	Required Cables, SRAMS, Jumpers, etc.
		SOFTWARE
	1	iDCS100 – BITBUS Toolbox – The set of six software utilities that simplify development of host application software
	1	iDCS110 – Bitware – iDCX 51 interface library and declaration files
	1	Starter Kit Application Software
BITBUSKITPLUS		Expanded version of the BITBUSKIT providing programming languages used to develop host (8086 environment) and node code (8051 environment) in addition to the basic BITBUS network.

Table 1. Standard BITBUS™ Interfaces

Interface	Specification
Electrical	RS485
Cable	10-conductor flat ribbon or 1 to 2 wire shielded twisted pair
Back-plane connector	64-pin Standard DIN
Control-board form-factor	Single-height, Double-depth Eurocard
Data Link control	Synchronous Data-link Control (SDLC)
Data transfer rate	62.5K baud, 375K baud and 500K to 2.4M baud
Message formats	Compatible with iDCX format command/response/status
Common command sequences	Integral Remote Access and Control (RAC) function
Operating systems	Interface libraries for iRMX 86, 88, 286R, MS-DOS, and ISIS (for iPDS only)

Table 2. BITBUS™ Microcontroller Interconnect Modes Of Operation

	Speed Kb/S	Maximum Distance Between Repeaters M/ft	Maximum # Nodes Per Segment*	Maximum # Repeaters Between A Master And Any Slave
Synchronous	500-2400	30/100	28	0
Self Clocked	375 62.5	300/1000 1200/4000	28 28	2 10

RELATED LITERATURE

iPCX 344A fact sheet
(Order #. 280414-002)

iRCB 44/10A fact sheet
(Order #: 280213-003)

iRCB 44/20A fact sheet
(Order #: 280721-2)

BITBUS Software Development Environment fact sheet
(Order #: 280622-001)

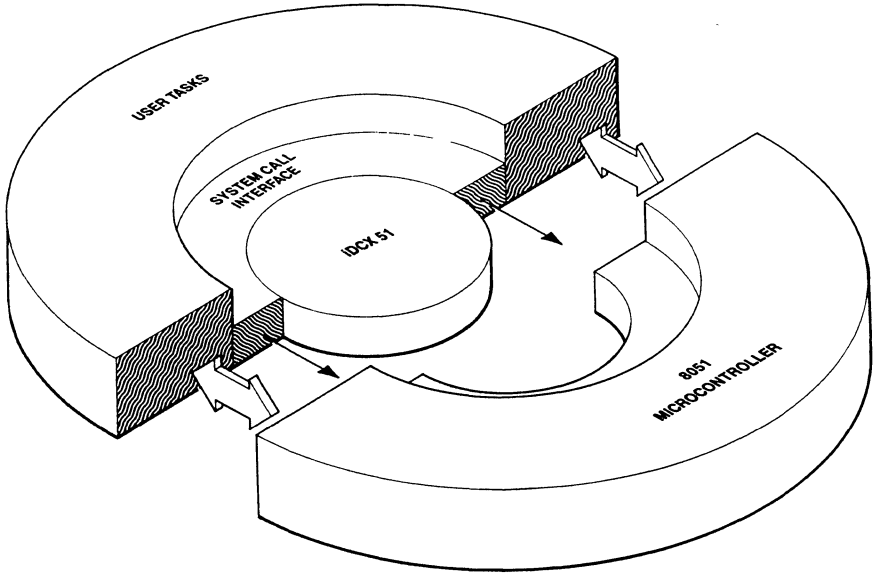
iDCX 51
DISTRIBUTED CONTROL EXECUTIVE

- **Supports MCS®-51 and RUPI™-44 Familes of 8-Bit Microcontrollers**
- **Real-Time, Multitasking Executive — Supports up to 8 Tasks at Four Priority Levels**
- **Local and Remote Task Communication**
- **Small—2.2K Bytes**
- **Reliable**
- **Simple User Interface**
- **Dynamic Reconfiguration Capability**
- **Compatible with BITBUS™/Distributed Control Modules (iDCM) Product Line**

The iDCX 51 Executive is compact, easy to use software for development and implementation of applications using the high performance 8-bit family of 8051 microcontrollers, including the 8051, 8044, and 8052. Like the 8051 family, the iDCX 51 Executive is tuned for real-time control applications requiring manipulation and scheduling of more than one task, and fast response to external stimuli.

The MCS-51 microcontroller family coupled with iDCX 51 is a natural combination for applications such as data acquisition and monitoring, process control, robotics, and machine control. The iDCX 51 Executive can significantly reduce applications development time, particularly BITBUS distributed control environments.

The iDCX 51 Executive is available in two forms, either as configurable software on diskette or as preconfigured firmware within the 8044 BEM BITBUS microcontroller.



280176-1

Figure 1. iDCX 51 Distributed Control Executive

*XENIX™ is a trademark of Microsoft Corporation.

October 1987
Order Number: 280176-003

MICROCONTROLLER SUPPORT

The iDCX 51 Executive is designed to support the MCS-51 and RUPI-44 families of 8-bit microcontrollers. MCS-51 microcontrollers that are supported include the 8051, 80C51, 8052, 8031, 8032, and 8751 devices. The RUPI-44 microcontrollers include the 8044, 8344, and 8744 devices. All of these microcontrollers share a common 8051 core.

ARCHITECTURE

Real-time and Multitasking

Real-time control applications must be responsive to the external environment and typically involve the execution of more than one activity (task or set of tasks) in response to different external stimuli. Control of an industrial drying process is an example. This process could require monitoring of multiple temperatures and humidity; control of fans, heaters, and motors that must respond accordingly to a variety of inputs. The iDCX 51 Executive fully supports applications requiring response to stimuli as they occur, i.e., in real-time. This real-time response is supported for multiple tasks often needed to implement a control application.

Some of the facilities precisely tailored for development and implementation of real-time control application systems provided by the iDCX 51 Executive are: task management, interrupt handling, message passing, and when integrated with communications support, message passing with different microcontrollers. Also, the iDCX 51 Executive is driven by events: interrupts, timers, and messages ensuring the application system always responds to the environment appropriately.

Task Management

A task is a program defined by the user to execute a particular control function or functions. Multiple programs or tasks may be required to implement a particular function such as "controlling Heater 1". The iDCX 51 Executive recognizes three different task states as one of the mechanisms to accomplish scheduling of up to eight tasks. Figure 2 illustrates the different task states and their relationship to one another.

The scheduling of tasks is priority based. The user can prioritize tasks to reflect their relative importance within the overall control scheme. For instance, if Heater 1 must go off line prior to Heater 2 then the task associated with Heater 1 shutdown could be assigned a higher priority ensuring the correct shutdown sequence. The RQ WAIT system call is also a scheduling tool. In this example the task implementing Heater 2 shutdown could include an instruction to wait for completion of the task that implements Heater 1 shutdown.

The iDCX 51 Executive allows for PREEMPTION of a task that is currently being executed. This means that if some external event occurs such as a catastrophic failure of Heater 1, a higher priority task associated with the interrupt, message, or timeout resulting from the failure will preempt the running task. Preemption ensures the emergency will be responded to immediately. This is crucial for real-time control application systems.

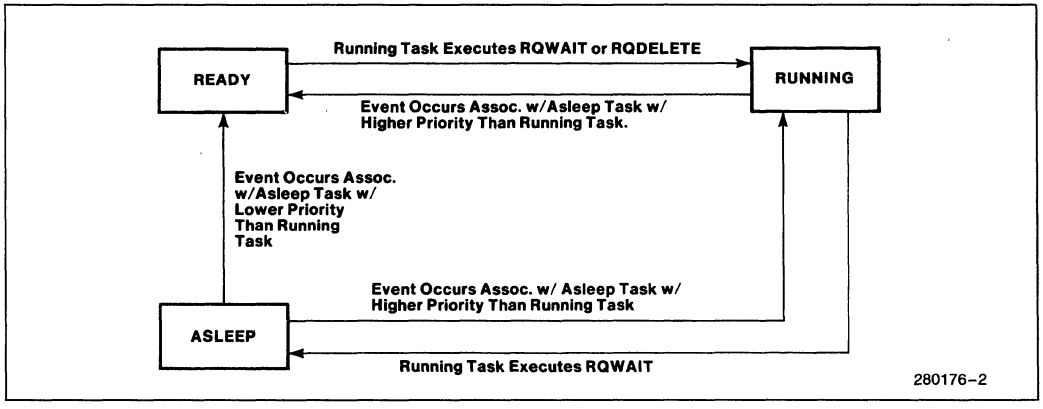


Figure 2. Task State Transition Diagram

Interrupt Handling

The iDCX 51 Executive supports five interrupt sources as shown in Table 1. Four of these interrupt sources, excluding timer 0, can be assigned to a task. When one of the interrupts occurs the task associated with it becomes a running task (if it were the highest priority task in a ready state). In this way, the iDCX 51 Executive responds to a number of internal and external stimuli including time intervals designed by the user.

Table 1. iDCX 51 Interrupt Sources

Interrupt Source	Interrupt Number
External Request 0	00H
Timer 0	01H
External Request 1	02H
Timer 1	03H
Internal Serial Port 1	04H

Message Passing

The iDCX 51 Executive allows tasks to interface with one another via a simple message passing facility. This message passing facility can be extended to different processors when communications support is integrated within a BITBUS/iDCM system, for example. This facility provides the user with the ability to link different functions or tasks. Linkage between tasks/functions is typically required to support development of complex control applications with multiple sensors (input variables) and drivers (output variables). For instance, the industrial drying process might require a dozen temperature inputs, six moisture readings, and control of: three fans, two conveyor motors, a dryer motor, and a pneumatic conveyor. The data gathered from both the temperature and humidity sensors could be processed. Two tasks might be required to gather the data and process it. One task could perform a part of the analysis, then include a pointer to the next task to complete the next part of the analysis. The tasks could continue to move between one another.

REMOTE TASK COMMUNICATION

The iDCX 51 Executive system calls can support communication to tasks on remote controllers. This feature makes the iDCX 51 Executive ideal for applications using distributed architectures. Providing communication support saves significant application development time and allows for more effective use of this time. Intel's iDCM product line combines hardware and software to provide this function.

In an iDCM system, communication between nodes occurs via the BITBUS microcontroller interconnect. The BITBUS microcontroller interconnect is a high performance serial control bus specifically intended for use in applications built on distributed architectures. The iDCX 51 Executive provides BITBUS support.

BITBUS™/iDCM COMPATIBLE

A pre-configured version of the iDCX 51 Executive implements the BITBUS message format and provides all iDCX 51 facilities mentioned previously: task management, interrupt handling, and message passing. This version of the Executive is supplied in firmware on the 8044 BEM with the iDCM hardware products: the iSBX™ 344A BITBUS Controller MULTIMODULE™; the iDCX 344A BITBUS controller board for the PC; and the iRCB boards.

Designers who want to use the iDCX executive on an Intel BITBUS board should purchase either DCS110 or DSC120 BITBUS software. Both of these products include an interface library to iDCX 51 procedures and other development files. It is not necessary to purchase the iDCX 51 Executive.

SIMPLE USER INTERFACE

The iDCX 51 Executive's capabilities are utlilized through system calls. These interfaces have been defined for ease of use and simplicity. Table 2 includes a listing of these calls and their functions. Note that tasks may be created at system initialization or run-time using the CREATE TASK call.

Other Functions such as GET FUNCTION IDS, ALLOCATE/DEALLOCATE BUFFER, and SEND MESSAGE, support communication for distributed architectures.

Table 2. iDCX 51 System Calls

Call Name	Description
TASK MANAGEMENT CALLS	
RQ$CREATE$TASK	Create and schedule a new task.
RQ$DELETE$TASK	Delete specified task from system.
RQGETFUNCTION$IDS	Obtain the function IDs of tasks currently in the system.
RQ$ALLOCATE	Obtain a message buffer from the system buffer pool.
RQ$DEALLOCATE	Return a message buffer to the system buffer pool.
RQ$SEND$MESSAGE	Send a message to specified task.
RQ$WAIT	Wait for a message event.
MEMORY MANAGEMENT CALLS	
RQGETMEM	Get available system memory pool memory.
RQ$RELEASE$MEM	Release system memory pool memory.
INTERRUPT MANAGEMENT CALLS	
RQ$DISABLE$INTERRUPT	Temporarily disable an interrupt.
RQ$ENABLE$INTERRUPT	Re-enable an interrupt.
RQ$WAIT	Wait for an interrupt event.
TIMER MANAGEMENT CALLS	
RQSETINTERVAL	Establish a time interval.
RQ$WAIT	Wait for an interval event.

Another feature that eases application development is automatic register bank allocation. The Executive will assign tasks to register banks automatically unless a specific request is made. The iDCX 51 Executive keeps track of the register assignments allowing the user to concentrate on other activities.

SYSTEM CONFIGURATION

The user configures an iDCX 51 system simply by specifying the initial set of task descriptors and configuration values, and linking the system via the RL 51 Linker and Locator Program with user programs.

Each task that will be running under control of the executive has an Initial Task Description (ITD) that describes it. The ITD specifies to the executive the amount of stack space to reserve, the priority level of the task (1–4), the internal memory register bank to be associated with the task, the internal or external interrupt associated with the task, and a function ID (assigned by the user) that uniquely labels the task. The ITD can also include a pointer to the ITD for the next task. In this way an ITD "chain" can be formed. For example, if four ITD's are chained together, then when the system is initialized, all four tasks will be put into a READY state. Then, the highest priority task will run.

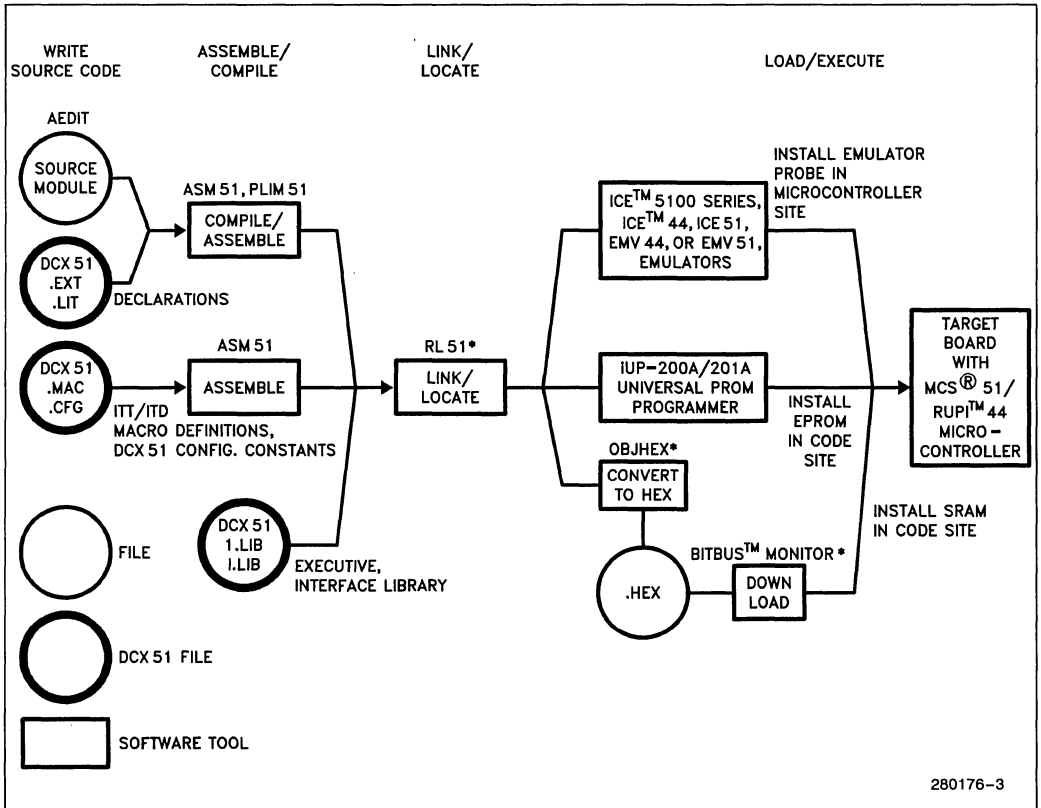
The DCX 51 user can control several system constants during the configuration process (Table 3). Most of these constants are fixed, but by including an Initial Data Descriptor (IDD) in an ITD chain, the system clock priority, clock time unit, and buffer size can be modified at run-time.

This feature is useful for products that use the same software core, but need minor modification of the executive to better match the end application. The initial data descriptor also allows the designer, who is using an 8044 BEM BITBUS Microcontroller, to modify the preconfigured (on-chip) iDCX 51 Executive.

Programs may be written in ASM 51 or PL/M 51. Intel's 8051 Software Development Package contains both ASM 51 and RL 51. Figure 3 shows the software generation process.

Table 3. DCX 51 Configuration Constants

Constant Name	Description
RQ CLOCK PRIORITY	The priority level of the system clock.
RQ CLOCK TICK	The number of time cycles in the system clock basic time unit (a "tick").
RQ FIRST ITD	The absolute address of the first ITD in the ITD chain.
RQ MEM POOL ADR	The start address of the System Memory Pool (SMP) in Internal Data RAM.
RQ MEM POOL LEN	The length of the SMP.
RQ RAM IDD	The absolute RAM address of where iDCX 51 checks for an Initial Data Descriptor (IDD) during initialization.
RQ SYS BUF SIZE	The size, in bytes, of each buffer in the system buffer pool.



NOTE:
*RL 51 is included with ASM51 and PL/M 51; OBJHEX and the BITBUS Monitor are part of the DCS100 BITBUS Toolbox.

Figure 3. Software Generation Process

SOPHISTICATED INTERNAL MEMORY MANAGEMENT

The amount of internal memory available ranges from 128 to 256 bytes depending on the type of microcontroller used.

Internal memory is used for the executive, stack spare for "running" tasks, space for message buffers, and reserved memory for variables storage. Other memory is used for register space. Except for register space, the allocation of internal memory is controlled by the executive, user-specified task/data descriptors and system configuration constants.

To optimize use of this limited resource, iDCX 51 provides dynamic (run-time) memory management.

INITIALIZATION AND DYNAMIC MEMORY MANAGEMENT

At initialization (see Figure 4), the iDCX 51 Executive creates the System Memory Pool (SMP) out of the remaining initial free space (i.e. memory not used by the iDCX 51 Executive or for register space). Next, stack space is created for each of the initial tasks that will be running on the system. If reserved memory is requested (using an IDD), that memory is also set aside. Finally, multiple buffers (size specified during iDCX 51 configuration or using an IDD) are allocated from any remaining memory. These buffers form the System Buffer Pool (SBP) that can be used to create additional stack space or to locate messages sent between tasks.

During run-time, the iDCX 51 Executive dynamically manages this space. If a task is deleted, its stack space is returned to the System Buffer Pool for use by other tasks or as a message buffer.

As new tasks are dynamicallly created, the executive reserves the needed stack space. If no space is available, the executive deallocates a buffer from the System Buffer Pool and then allocates the needed stack space.

To send or receive a message, the executive allocates one or more buffers from the SBP for space to locate the message. With iDCX 51, messages can be optionally located in external (off-chip) memory. The pre-configured executive in the 8044 BEM BITBUS microcontroller, however, always locates messages in internal memory.

RELIABLE

Real-time control applications require reliability. The nucleus requires about 2.2K bytes of code space, 40 bytes on-chip RAM, and 218 bytes external RAM.

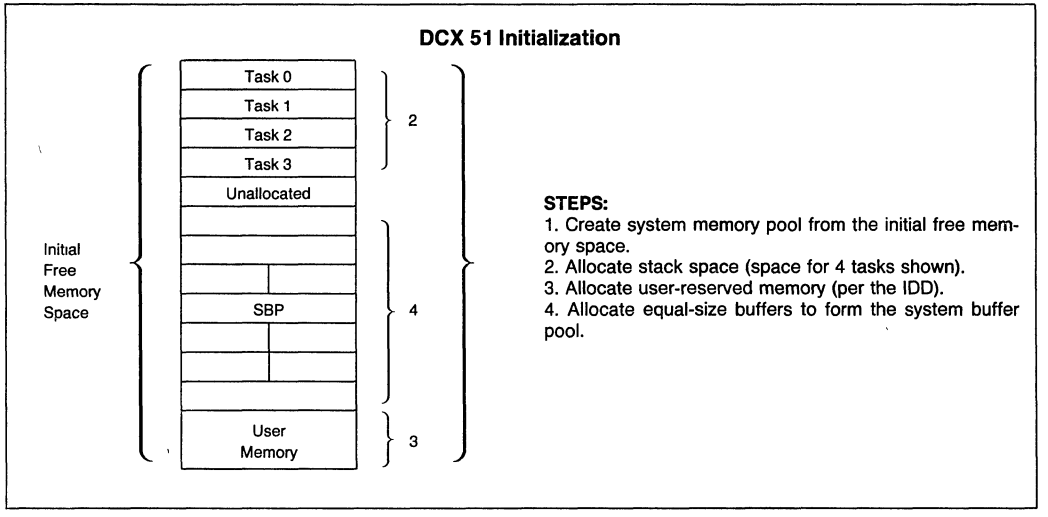


Figure 4. iDCX 51 Initialization of Internal Memory

Streamlined code increases performance and reliability, and flexibility is not sacrificed as code may be added to either on-chip or external memory.

The iDCX 51 architecture and simple user interface further enhance reliability and lower cost. For example, the straightforward structure of the user interfaces, and the transparent nature of the scheduling process contribute to reliability of the overall system by minimizing programming effort. Also, modularity increases reliability of the system and lowers cost by allowing user tasks to be refined independent of the system. In this way, errors are identified earlier and can be easily corrected in each isolated module.

In addition, users can assign tasks a Function ID that allows tracking of the tasks associated with a particular control/monitorig function. This feature reduces maintenance and trouble shooting time thus increasing system run time and decreasing cost.

OPERATING ENVIRONMENT

The iDCX 51 Executive supports applications development based on any member of the high performance 8051 family of microcontrollers. The Executive is available on diskette with user linkable libraries or in the 8044 BITBUS Enhanced Microcontroller preconfigured in on-chip ROM. (The 8044 BEM is an 8044 component that consists of an 8051 microcontroller and SDLC controller on one chip with integral firmware.)

When in the iDCM environment (Figure 5), the preconfigured iDCX 51 Executive can communicate with other BITBUS series controller boards. The BITBUS board at the master node can be associated with either an iRMX™, PC-DOS or XENIX* host system.

DEVELOPMENT ENVIRONMENT

Intel provides a complete development environment for the MCS-51 and RUPI-44 families of microcontrollers. The iDCX 51 Executive is only one of many of the software development products available. The executive is compatible with the following software development utilities available from Intel:

- 8051 Macro Assembler (ASM 51)
- PL/M 51 Compiler
- RL 51 Linker and Relocator Program
- LIB 51

Intel hardware development tools currently available for MCS-51 and RUPI-44 microcontroller development are:

- ICE-5100/252 Emulator for the MCS-51 family of microcontrollers
- ICE-5100/044 Emulator for the RUPI-44 family of microcontrollers (8044, 8344, 8744)
- iUP-200A/201A PROM Programmer, 21X software, and iUP programming modules

The DCX 51 Executive is also compatible with older hardware development tools (no longer available), which include:

- EMV-51/44 Emulation Vehicles
- ICE-51/44 In-Circuit Emulators

Table 4 shows the possible MCS-51 and RUPI-44 families development environments: host systems, operating systems, available software utilities, and hardware debug tools.

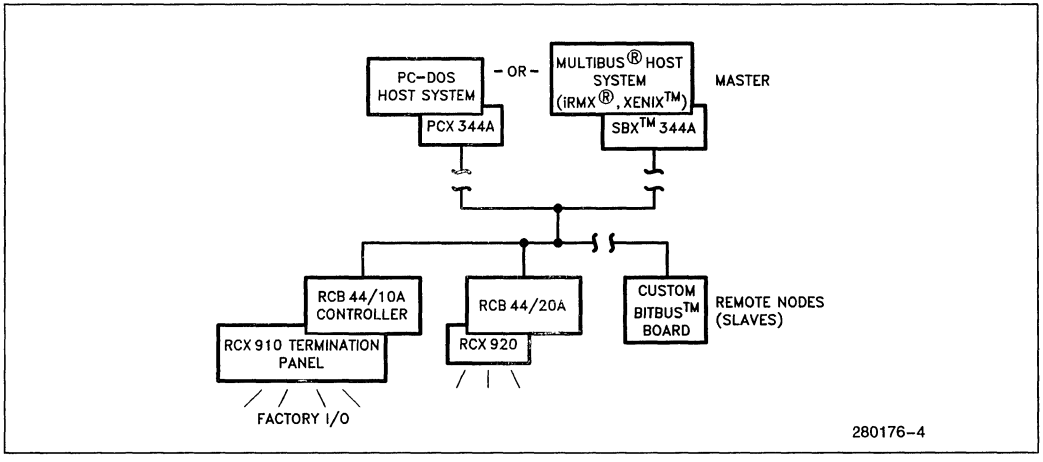


Figure 5. iDCM Operating Environment

SPECIFICATIONS

Supported Microcontrollers

8031 80C31
8051 80C51
8032 8751
8744 8044
8344 8052

Compatible DCM BITBUS™ Software

DCS 100 BITBUS Toolbox Host Software Utilities

DCS 110 BITWARE DCM44 Code for BITBUS emulation

Reference Manual (Supplied)

460367-001— iDCX 51 Distributes Control Executive User's Guide for Release 2.0.

ORDERING INFORMATION

Part Number	Description
DCX51SU	Executive for 8051 Family of Microcontrollers. Single User License, Development Only. Media Supplied for All Host Systems (Table 3).
DCX51RF	Royalty (Incorporation) Fee for iDCX Executive. Set of 50 incorporations. IDCX 51 RF does not ship with software (Order DCX 51SU).

Table 4. MCS®-51/RUPI™-44 Families Development Environments

Development Utilities	Host Systems				
	PC/MS-DOS	iRMX® 86	iPDS™	Intellec®	
				Series II	Series III/IV
SOFTWARE					
ASM 51 + Utilities(1)	✓	✓	✓	✓	✓
PL/M 51 + Utilities(1)	✓	✓	✓	✓	✓
iDCX 51 Executive	✓	✓	✓		✓
HARDWARE					
ICE-5100/044/252	✓				✓
iUP-200A/201A	✓				✓
EMV-51(2), EMV-44(2)				✓	✓
ICE-51(2), ICE-44(2)				✓	✓
iPDS + iUP-F87/44A PROM Programmer			✓		

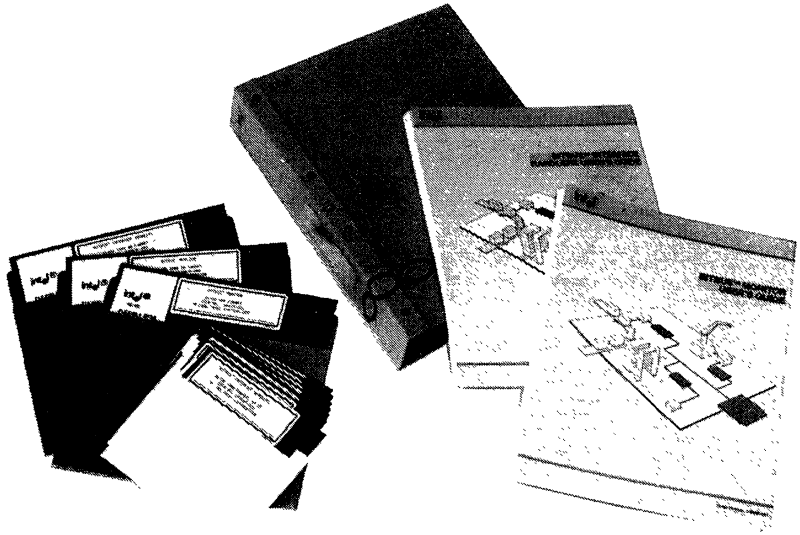
NOTES:
1. Utilities include RL 51, LIB 51, and AEDIT. Software for Series II systems is down-revision version.
2. These products are no longer available.

DCS100 BITBUS™ TOOLBOX HOST SOFTWARE UTILITIES

- **Six Utilities Simplify Development of Host Software for Controlling BITBUS™-Based Systems**
- **Includes the BITBUS™ Monitor Which Provides On-Line Monitoring and Control of a BITBUS™ System**
- **Reliable and Easy to Use**
- **Universal BITBUS™ Interface and BITBUS™ Interface Handler Libraries Provide 32 System Management/ Control Procedure**
- **Compatible with Intel's C, PL/M and ASM Languages**
- **For DOS, iRMX® 86/286, XENIX*, and iPDS™ Host Systems**

The BITBUS Toolbox provides a set of utilities designed to simplify development of host system software for controlling a BITBUS network. The Toolbox includes: two libraries of procedures that can be called from the host code; an on-line program called the BITBUS Monitor which is invaluable for troubleshooting, monitoring, and manually controlling a system; and code conversion/communication software to support applications software development on a PC.

The procedure libraries contain common procedures used by the host to read or write data to remote node I/O ports, download or upload programs and data, start and stop tasks (program modules) running on the nodes, send and receive messages, and perform a variety of system status and control functions. By using these libraries, the programmer's task of generating BITBUS host code is substantially reduced.



280732-1

*IBM, XT and AT are trademarks of International Business Machines Corporation.
*XENIX is a trademark of Microsoft Corporation.

October 1987
Order Number: 280732-001

THE BITBUS™ TOOLBOX—PRODUCT DESCRIPTION

The BITBUS Toolbox is used to develop host code for controlling a BITBUS network, and is an essential tool for both centralized and distributed control applications.

With centralized control, the host code sends commands to a node to read and update the I/O. All the decisions are made at the host. Normally, this kind of system would require extensive host code. However, the Toolbox includes the UBI and BIH procedure libraries that can be called to perform simple or complex control procedures.

In addition to the Toolbox, all BITBUS boards include, in firmware, a set of procedures known as Remote Access and Control (RAC). By sending simple messages to these procedures, basic I/O functions can be performed. The RAC procedures are listed in Table 1.

With distributed control systems, programs run on the remote BITBUS boards (nodes) and offload the host system of most decision making responsibilities. Using UBI calls or the BITBUS Monitor, commands can be sent to the nodes to control tasks or to periodically upload data for further analysis or storage. The software tools in the BITBUS Toolbox reduce the time and effort necessary to develop host code for these applications.

In addition to the DCS100 BITBUS Toolbox, other host code tools include a full set of host software compilers, libraries, debuggers, and in-circuit emulators. The BITBUS Toolbox is described in detail in the sections that follow.

Table 1. Remote Access and Control Procedures

Name	Function
RESET_STATION	Perform a software reset.
CREATE_TASK	Perform an RQ$CREATE$TASK system call.
DELETE_TASK	Perform an RQ$DELETE$TASK system call.
GET_FUNCTION_ID	Perform an RQGETFUNCTION$IDS call.
RAC_PROTECT	Suspend or resume RAC services.
READ_I/O	Return values from specified I/O ports.
WRITE_I/O	Write to the specified I/O ports.
UPDATE_I/O	Update the specified I/O ports.
UPLOAD_MEMORY	Return the values in specified memory area.
DOWNLOAD_MEMORY	Write values to specified memory area.
OR_IO	OR values into specified I/O ports.
AND_IO	AND values into specified I/O ports.
XOR_IO	XOR values into specified I/O ports.
READ_INTERNAL	Read values at specified internal RAM areas.
WRITE_INTERNAL	Write values to specified internal RAM areas.
NODE_INFO	Return device related information.
OFFLINE	Set node offline.
UPLOAD_CODE	Read values from code memory space.
DOWNLOAD_CODE	Write values to specified EEPROM memory.

BITBUS™ TOOLBOX UTILITIES

The DCS100 BITBUS Toolbox includes six host software utilities. They include:

- Universal BITBUS Interface (UBI)—a set of 28 procedures for implementing remote I/O and controlling a BITBUS network.
- BITBUS Interface Handlers (BIH)—four basic procedures for sending/receiving messages over a BITBUS network.
- BITBUS Monitor (BBM)—An on-line program with 36 commands that enable a user to configure, troubleshoot, monitor, and manually control a BITBUS network.
- PC Bridge—Communications program for the PC to support software development on a PC and download into an iRMX or XENIX-hosted BITBUS network.
- OBJHEX Conversion Utility—Converts an object file to hex format for downloading code.
- UDI2DOS—Converts Intel object code programs to .exe format for execution on the PC.

Universal BITBUS™ Interface

UBI is a library of 28 procedures called by the host program to manage the I/O, download or upload code and data, manage tasks on a node, send and receive messages, and perform an assortment of miscellaneous functions. These procedures are listed in Table 2, below.

Table 2. UBI Procedure Calls

I/O	
BQANDI/O	AND I/O
BQORI/O	OR I/O
BQXORI/O	Excl. OR I/O
BQ$WRITE$I/O	Write I/O
BQ$READ$I/O	Read I/O
BQ$UPDATE$I/O	Write I/O and read back
MEMORY MANAGEMENT	
BQABSLOAD	Download program to code memory
BQ$WRITE$CODE$MEM	Write to code memory
BQ$READ$CODE$MEM	Read code memory
BQ$WRITE$INT$MEM	Write to internal data memory
BQ$READ$INT$MEM	Read internal data memory
BQ$WRITE$EXT$MEM	Write to external data memory
BQ$READ$EXT$MEM	Read external data memory
TASK MANAGEMENT	
BQ$CREATE$TASK	Create task
BQ$DELETE$TASK	Delete task
BQGETFUNCTION$IDS	Read task function IDs
MESSAGE MANAGEMENT	
BQ$FLUSH	Clear an iSBX/iPCX interface
BQ$RECEIVE$MESSAGE	Receive a message
BQ$SEND$MESSAGE	Send a message

Table 2. UBI Procedure Calls (Continued)

MISCELLANEOUS CALLS	
BQ$DELAY	Perform a time delay
BQ$NODE$INFO	Return node information
BQ$PROBE$SBX	Check for BITBUS iSBX board
BQ$PROTECT$RAC	Lockout (protect) a node
BQ$RESET$DEVICE	Initiate a software reset
BQ$RESYNC$NODE	Set a node offline, prep. to resync
BQSETPORT	Set port I/O address
BQSETSBX	Set port I/O address
BQ$SHELL	Shell escape and then return

The UBI utility includes libraries interfacing with PL/M and C host code running within DOS, iRMX, and XENIX environments. Also included are declaration files for the procedures.

To use these procedures, the UBI calls are incorporated into the source code modules together with parameters needed by the procedures (e.g. node address, port address, memory location, task number, and data). The source module and UBI declaration files are then compiled and linked with the UBI library.

When the call executes, the called procedure will be performed, data will be returned (in the case of READ or UPLOAD procedures) together with an error code. These error codes can help the host system take corrective action.

BITBUS™ Interface Handlers (BIH)

BIH is a library of four basic procedures for sending and receiving messages between the host and network nodes. These procedures, listed in Table 3, are most useful when generating custom UBI-like procedures. The BIH utility includes procedure libraries and declaration files for DOS, iRMX 86/88/286, and ISIS-PDS (iPDS)-based systems.

Table 3. BIH Procedure Calls

Call Name	Description
CQDCMINIT	Performs any initialization required by the BITBUS Interface Handlers.
CQDCMRECEIVE	Receives one message from any BITBUS node.
CQDCMSTATUS$CHECK	Determines whether a BITBUS message is available to receive.
CQDCMTRANSMIT	Transmits one message to any BITBUS node.

To use these libraries, the appropriate declaration file is included with the host source code. The modules are then compiled and the resultant object module is linked to the BIH library.

BITBUS™ Monitor

The BITBUS Monitor (BBM) is an on-line program that is invaluable for troubleshooting and testing a BITBUS system and can also be used to manually control a system. BBM commands are listed in Table 4.

Table 4. BITBUS™ Monitor Commands

I/O	
AIO	And I/O
OIO	Or I/O
RIO	Read I/O
UIO	Update I/O
WIO	Write to I/O
XIO	Exclusive OR I/O
MEMORY MANAGEMENT	
LOAD	Download to code memory
RCMEM	Read code memory
RIMEM	Read internal memory
RXMEM	Read data memory
WCMEM	Write to code memory
WIMEM	Write to internal memory
WXMEM	Write to data memory
TASK MANAGEMENT	
CTASK	Create a task
DTASK	Delete a task
SYS	Display node task status

MESSAGE MANAGEMENT	
DMSG	Display a message
RMSG	Receive a message
SMSG	Send a message
TMSG	Sends, receives, displays a message
MISCELLANEOUS COMMANDS	
DELAY	Suspend Activity
EXIT	Exit BBM
FLUSH	Clears an iSBX/iPCX interface
HELP	Provide on-line help
INCLUDE	Open/execute a BBM file
LIST	Creates a copy of the BBM session
LOCK	Lockout (protect) a node
NODEINFO	Node information
PAUSE	Wait until <CR>
RESET	SW reset at a node
RESYNC	Set a node offline
SETPORT/ SETSBX	Set port I/O address
SHELL	XENIX/DOS shell escape from BBM
SYMBOLS	Display/create/change the value of a user symbol
UNLOCK	Unprotect
VERBOSE	Controls echo and prompts

I/O ACCESS

Six commands are provided for writing to and reading from I/O ports on remote nodes. With these commands, an operator can test the I/O connected to a BITBUS node or monitor the status of an input port. The I/O commands allow an operator to quickly isolate a problem at a remote node.

MEMORY ACCESS

Seven memory access commands are provided. These commands allow the operator to download and upload both code (programs) and data (variables) between the host system and remote BITBUS nodes. Internal RAM memory within the 8044BEM microcontroller can also be accessed. In addition, the BBM supports code download to both static RAM and E^2PROM devices. The memory access commands are especially useful for on-target application development.

The BITBUS Monitor enables the user to reference a memory location by using a symbolic reference or label. For example, if a task running on a node includes a program variable called "rate", the operator can modify this variable simply by typing:

```
WIMEM <node address> .rate 6CH
```

In this case, the program will execute with a value of 6C hex for "rate".

Symbolic references can also be used for other BBM parameters, such as node address, port address, and data. Symbolic access allows the user to more easily test and modify programs at run time.

TASK MANAGEMENT

Four commands are available to monitor and control the running of tasks on the nodes.

The DCX 51 real time multitasking executive found on all BITBUS boards can support up to 7 user tasks (in addition to the RAC task). Each of these tasks have an initial Task Descriptor (ITD) which assigns a function ID to the task plus other important run-time parameters used by the executive. By chaining ITDs together, multiple tasks can become active upon power up.

The BBM commands allow tasks to selectively be made active (CTASK) or inactive (DTASK). In addition, the SYS command can be used to display which nodes are present and operational in a system and display the function IDs for active tasks on each node. The task management commands are especially useful when developing/troubleshooting multitasking control programs.

MESSAGE OPERATIONS

These four commands are used to send and receive messages to and from tasks on remote nodes.

MISCELLANEOUS COMMANDS

The BBM includes 15 commands that are used to control the operating status of nodes, and to support various troubleshooting functions. These commands include:

The HELP command—an on-line facility that displays the complete BBM command directory or detailed information on using the commands.

The SHELL command—allows an operator to do a shell escape to DOS or XENIX, perform the needed operating system function, and return to the monitor.

The RESET, FLUSH, and RESYNC commands—used to clear a node that is hung.

OPERATING ENVIRONMENT

The BITBUS Monitor will run on DOS, iRMX 86/286, XENIX and iPDS-based systems. Both 5¼" and 8" media is provided for iRMX and XENIX systems. The iPDS version of the monitor does not include the following BBM commands (or equivalent UBI calls): DELAY, LIST, PAUSE, RCMEM, RESYSC, SETPORT, SYMBOLS, TMSG, VERBOSE, WCMEM.

PC Bridge

The PC Bridge is a communications program that runs on a PC-DOS or MS-DOS system, and is used to establish a communication link between the PC and an Intel iRMX 86/286 or XENIX-based microcomputer system. The software engineer can use the Bridge in two ways. First, he can develop host or node programs on the PC and download the code to the host system or remote nodes. He can also use the PC as a virtual terminal to the host system. The PC Bridge effectively expands the development environment for the software engineer.

The link between the PC and the host microcomputer can either be over an RS232 cable (supplied) or via a modem link. The PC Bridge transfers data at up to 19.2K baud (asynchronous) and supports XON/XOFF flow control.

OBJHEX

OBJHEX is an object code to hex code conversion utility similar to the OH51 hex converter supplied with Intel "8051" languages. OBJHEX has the additional ability to retain the object module's symbol table during the conversion process. The table is stored at the host system and enables the BITBUS Monitor to symbolically access program memory. OBJHEX runs on both DOS and iRMX86 (5¼", 8" media)-based systems.

UDI2DOS

UDI2DOS converts Intel object code (8086 OMF) to the .exe format so that it will run within a DOS environment.

SPECIFICATIONS

Media Provided

	BBM	UBI	BIH	PC Bridge	OBJHEX	UDI2DOS
Series II						
III			X			
IV						
iPDS	A	A	A			
IRMX 5¼"	X	X	X	X	X	
8"	X	X	X	X	X	
XENIX 5¼"	X	X		B	X	
8"	X	X		B	X	
DOS	X	X	X	X	X	X

NOTES:
A. iPDS uses Release 1 Toolbox.
B. Supports operation with XENIX. XENIX disks not required.

Documentation (supplied)

BITBUS Toolbox Overview and Installation Guide	460235-001
BITBUS Monitor User's Guide	148686-002
Universal BITBUS Interface User's Guide	460236-001
BITBUS Interface Handlers User's Guide	148685-002
PC Bridge Communications Utility User's Guide	149236-001
BITBUS OBJHEX Conversion Utility User's Guide	460237-001

Compatible Software

Intel ASM, PL/M, and C languages (8086/80286/80386 versions)

Order Codes

Order Number	Description
DCS100SU	BITBUS Toolbox Host Software Utilities, single-use license for development only. Includes RS232 cables to connect an Intel microcomputer system with an IBM* PC-XT* or PC-AT*, and full documentation. See above for media provided.
DCS100BY	BITBUS Toolbox Host Software Utilities. Same as above, except sold with a buyout license. Allows incorporation of UBI and BIH procedure libraries—no additional incorporation fee is required.

DCS110 BITWARE
DCS120 PROGRAMMERS SUPPORT PACKAGE

- **Supports Calls to the 8044BEM Microcontroller On-Chip, Multitasking DCX 51 Executive**
- **Fully Compatible with Intel's ASM51 and PL/M51 Languages**
- **DCS110 also Includes DCM44 Code to Support Emulation/Debug of BITBUS™ Node Code using Intel In-Circuit Emulators**
- **For DOS, iRMX®, iPDS™, and Series III/IV Development Environments**

The DCS110 and DCS120 packages are designed to support software development of distributed control BITBUS applications. Both products include a DCX51 interface library so that BITBUS application programs can make calls to the DCX51 Executive. DCS110 also includes a DCM44 downloadable file that enables an Intel in-circuit emulator such as the ICE™ 5100/044 to emulate a BITBUS environment. By using an in-circuit emulator together with DCS110, the developer can easily and quickly debug BITBUS application code.

280731-1

October 1987
Order Number: 280731-001

DCX 51 ENVIRONMENT

The 8044BEM microcontroller, used on every BITBUS board, includes in firmware a preconfigured version of the DCX 51 Executive. DCX 51 provides a variety of services to the application code, including: task management; interrupt management; inter-task communications; memory management; and timing services. Up to 7 user tasks can run concurrently under DCX 51. Each task has a unique Initial Task Descriptor (ITD) that describes to the executive several run-time parameters (e.g. stack space, priority level, etc.). By also specifying an Initial Data Descriptor (IDD), the executive can be partially reconfigured. Modifiable run-time constants include the system clock rate, clock priority, internal memory buffer size, and user (internal) memory size. DCX 51 calls are listed in Table 1.

By running applications under DCX 51, the designer can make optimal use of the 8044BEM microcontroller. If a task needs to wait for a message, an interrupt, or a time period, DCX 51 will temporarily assign access to the 8044 to another task. In this way, multiple tasks can access the microcontroller.

Table 1. DCX 51 Procedure Calls

Call Name	Description
Task Management Calls	
RQ$CREATE$TASK	Create and schedule a new task.
RQ$DELETE$TASK	Delete specified task from system.
RQGETFUNCTION$IDS	Obtain the function IDs of tasks currently in the system.
Intertask Communication Calls	
RQ$ALLOCATE	Obtain a message buffer from the system buffer pool.
RQ$DEALLOCATE	Return a message buffer to the system buffer pool.
RQ$SEND$MESSAGE	Send a message to specified task.
RQ$WAIT	Wait for a message event.
Memory Management Calls	
RQGETMEM	Get available memory from the system memory pool.
RQ$RELEASE$MEM	Release memory to the system memory pool.
Interrupt Management Calls	
RQ$DISABLE$INTERRUPT	Temporarily disable an interrupt.
RQ$ENABLE$INTERRUPT	Re-enable an interrupt.
RQ$WAIT	Wait for an interrupt event.
Timer Management Calls	
RQSETINTERVAL	Establish a time interval.
RQ$WAIT	Wait for an interval event.

Interfacing to the DCX 51 Executive

To interface with the executive, DCS110 and DCS120 both include a DCX 51 interface library plus a set of "include" files. The interface library, which is linked to the application modules, allow the code to access DCX 51 procedures. The "include" files consist of DCX 51 declaration and macro definition files that help simplify source code development. These files are listed in Table 2.

DCS110 Bitware Software Package

In addition to the DCX 51 interface files, DCS110 also includes a DCM44 object file to support debug of node code using an Intel in-circuit emulator. DCM44 is the firmware found in all 8044BEM BITBUS microcontrollers and together with an Intel in-circuit emulator, successfully duplicates the 8044BEM environment. Emulators that are supported include the ICE™ 5100/044, the ICE 44, and the EMV 44.

Developing Applications Software

Using DCS110 or DCS120 software to develop BITBUS applications software is a straightforward, multi-step process as diagrammed in Figure 1. The designer uses a text editor to write the application code either in ASM 51 or PL/M 51. The source code modules are then assembled/compiled along with the DCX 51 "include" files. The final step is to link together all of the modules, the DCX 51 interface library, and the DCM44I.LIB file. The linked/located absolute object module can then be downloaded to the target board or burned into EPROM.

Table 2. DCS110/120 Files

Filename	Description
DCX 51 Support Files:	
DCX51I.LIB	Interface library to the DCX 51 executive. Provides the linker with the address of data variables and entry points for DCX51 procedures called from other object modules.
DCX51A.EXT DCX51A.LIT DCX51P.LIT	External and literal declaration files. These files support DCX 51 calls from ASM 51 and PL/M 51 code.
DCXB0P.EXT DCXB1P.EXT DCXB2P.EXT DCXB3P.EXT	DCX 51 External procedure declarations for PL/M 51 modules using 8044 register banks 0, 1, 2 or 3.
DCX51A.MAC	Initial Task Descriptor (ITD) and Initial Data Descriptor (IDD) macro definitions.
APPL1.A51 APPL2.A51	Sample application, parts 1 and 2; template for generating ITDs and IDD.
DCM44I.LIB	This file maps out reserved memory needed by the 8044BEM firmware and is linked to other user object modules using the RL51 Linker.
DCM44 Firmware Files (DCS110 Only):	
DCM44	DCM44 (BITBUS) code for Intel ICE™/EMV emulators.

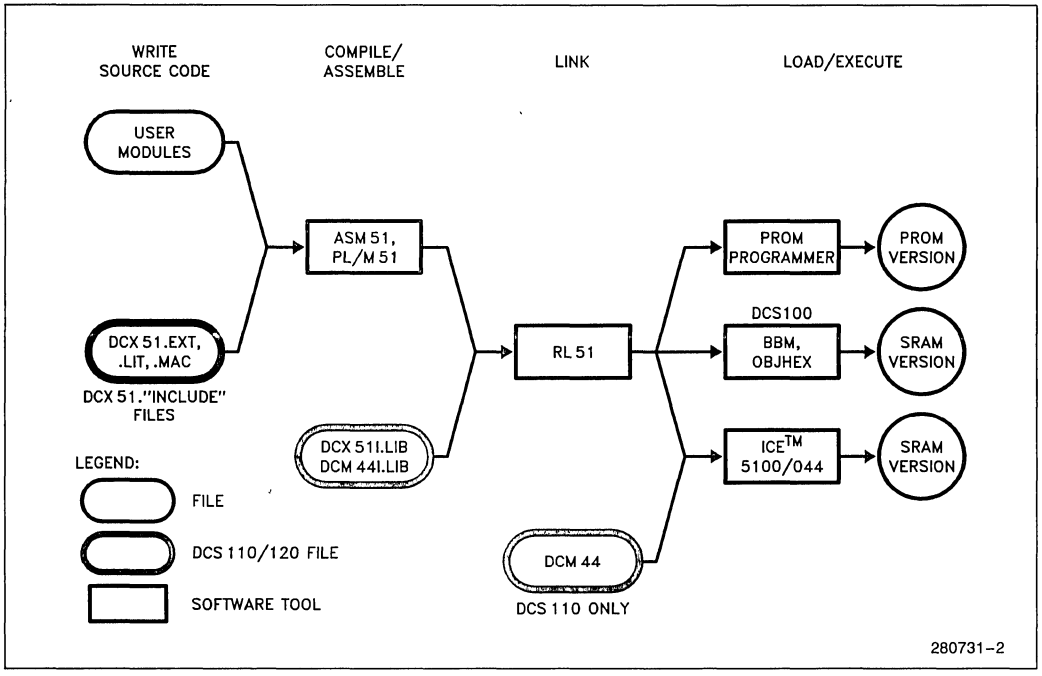


Figure 1. DCS110/120 Software Development Environment

Development Environments

Both DCS110 and DCS120 are shipped with media to support software development on PC/MS-DOS, iRMX 86, iPDS, and Intellec® Series III/IV systems. DCS110 is available with a single-use license for application development and debug. Designers planning to incorporte DCX 51 files in their application should purchase the DCS120 "buyout" product.

Order Codes	Description
DCS110SU	Bitware Software Package. Includes DCM44 code to emulate a BITBUS environment when using an Intel in-circuit emulator and interface files to support procedure calls to DCX 51. Provided with documentation and PC-DOS, iRMX 86 (5¼", 8"), iPDS, and Series III/IV media. Single-use license.
DCS120BY	Programmers Support Package. Includes interface files to support procedure calls to DCX 51. Provided with documentation and PC-DOS, iRMX 86 (5¼", 8"), iPDS, and Series III/IV media. Buyout license allows incorporation of software into product—no additional incorporation fee is required.

COMPATIBLE SOFTWARE TOOLS

DCS100	BITBUS Toolbox Host Software Utilities for PC/MS-DOS, iRMX 86/286, XENIX*, iPDS, and Series III/IV host systems.
AEDIT	Source Code and Text Editor for all Intel host environments (consult data sheet for order codes).

*XENIX is a trademark of Microsoft Corp.

8051 LANGUAGES

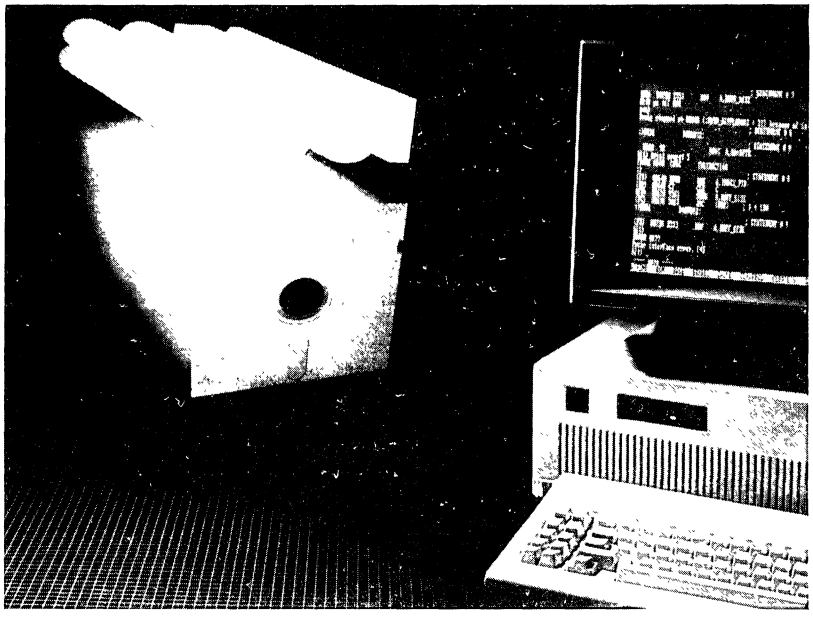
(Note: All products also include RL51 Linker/Relocator, LIB51 Librarian, and OH51 object to hex code converter)

D86ASM51	ASM 51 Assembler for PC-DOS host system
R86ASM51	ASM 51 Assembler for iRMX 86 host system
I86ASM51	ASM 51 Assembler for Series III/IV host systems
MC151ASM	ASM 51 Assembler for iPDS and Series II host systems
D86PLM51	PL/M 51 Compiler for PC-DOS host system
R86PLM51	PL/M 51 Compiler for IRMX 86 host system
I86PLM51	PL/M 51 Compiler for Series III/IV host systems
iMDX352	PL/M 51 Compiler for iPDS and Series II host systems

IN-CIRCUIT EMULATORS AND PROM PROGRAMMERS

(Note: + indicates that the product is no longer available)

ICE5100/044	In-Circuit Emulator for the RUPI™-44 Family (hosted on PC-DOS, and Series III/IV—see data sheet for order codes)
ICE-44+	8044 In-Circuit Emulator (hosted on Series II–IV systems)
iPDSEMV44CON+	Kit to add 8044 support to an EMV-51/51A emulator (iPDS host)
iUP-200A, iUP-201A	Universal PROM programmer (hosted on PC-DOS, iPDS, and Series III/IV; see data sheet for order codes)



COMPLETE SOFTWARE DEVELOPMENT SUPPORT FOR THE MCS®-51 FAMILY OF MICROCONTROLLERS

Intel supports application development for its MCS®-51 family of microcontrollers with a complete set of development languages and utilities. These tools include a macroassembler, a PL/M compiler, linker/relocator program, a librarian utility, and an object-to-hex utility. Develop code in the language(s) you desire, then combine object modules from different languages into a single, fast program. These tools were designed to work with each other, with the MCS-51 architecture, and with the Intel ICE5100 in-circuit emulator.

FEATURES

- Support for all members of the Intel MCS-51 family of embedded microcontrollers
- ASM-51 Macroassembler
- PL/M-51 high-level language
- Linker/Relocator program
- Library utility
- Object to hexadecimal converter
- Hosted on IBM PC XT/AT V3 0 or later
- Worldwide service and support

November, 1989
Order Number 280819-001

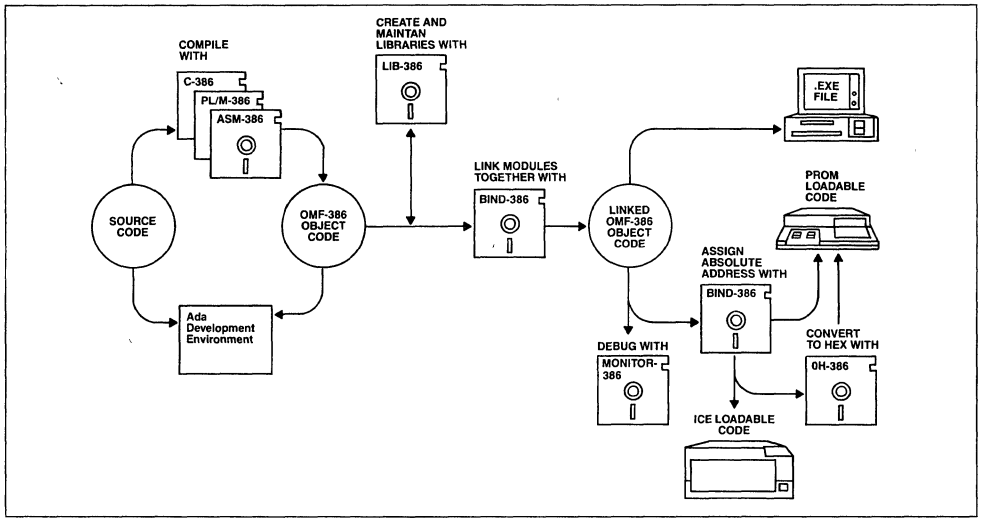


Figure 1: MCS®-51 Application Development Process

ASM-51 MACROASSEMBLER

ASM-51 is the macroassembler for the MCS-51 family of microcontrollers. ASM-51 provides full and accurate support for all of the specific component's instructions. It also provides symbolic access to the many features of the MCS-51 family of microcontrollers. Also provided is an "include" file with all the appropriate component registers and memory spaces defined.

The macro facility in ASM-51 saves development and maintenance time, since common code sequences need only be developed once.

PL/M-51 COMPILER

PL/M-51 is a high-level language designed to support the software requirements of the MCS-51 family of microcontrollers. The PL/M-51 compiler translates PL/M high-level language statements into MCS-51 relocatable object code. Major features of the PL/M-51 compiler include:

- **Structured programming for ease of maintenance and enhancement.** The PL/M-51 language supports modular and structured programming, making programs easier to understand, maintain, and debug.
- **Data types facilitate various common functions.** PL/M-51 supports three data types to facilitate various arithmetic, logic and address functions. The language also uses BASED variables that map more than one variable to the same memory location to save memory space.
- **Interrupt attribute speeds coding effort.** The INTERRUPT attribute allows you to easily define interrupt handling procedures The compiler will generate code to save and restore the program status word for INTERRUPT procedures.
- **Code optimization reduces memory requirements.** The PL/M-51 compiler has four different levels of optimization for significantly reducing the size of the program.
- **Language compatibility saves development time.** PL/M-51 object modules are compatible with object modules generated by all other MCS-51 language translators. This compatability allows for easy linking of all modules and the ability to do symbolic debugging with the Intel ICE5100 in-circuit emulator.

RL-51 LINKER/RELOCATOR

Intel's RL-51 utility is used to link multiple MCS-51 object modules into a single program, resolve all references between modules and assign absolute addresses to all relocatable segments. Modules can be written in either ASM-51 or PL/M-51.

LIB-51

The Intel LIB-51 utility creates and maintains libraries of software object modules. Standard modules can be placed in a library and linked into your applications programs using RL-51. When using libraries, the linker will link only those modules that are required to satisfy external references.

OH OBJECT TO HEXADECIMAL CONVERTER

The OH utility converts Intel OMF-51 object modules into standard hexadecimal format. This allows the code to be loaded directly into PROM via non-Intel PROM programmers.

SERVICE, SUPPORT, AND TRAINING

Intel augments its MCS-51 architecture family of development tools with a full array of seminars, classes, and workshops; on-site consulting services, field application engineering expertise; telephone hot-line support; and software and hardware maintenance contracts. This full line of services will ensure your design success.

ORDERING INFORMATION

D86ASM51*	MCS-51 Assembler for PC XT or AT system (or compatible), running DOS 3.0 or higher
D86PLM51*	PL/M-51 Software Package for PC XT or AT system (or compatible), running DOS 3.0 or higher

*Also includes: Relocator/Linker, Object-to-hex converter, and Librarian.

For direct information on Intel's Development Tools, or for the number of your nearest sales office or distributor, call 800-874-6835 (U.S.). For information or literature on additional Intel products, call 800-548-4725 (U.S. and Canada).

ICE™-5100/044 IN-CIRCUIT EMULATOR

IN-CIRCUIT EMULATOR FOR THE RUPI™-44 FAMILY OF PERIPHERALS

The ICE-5100/044 In-Circuit Emulator is a complete hardware/software debug environment for developing embedded control applications based on the Intel RUPI™-44 family of peripherals, including the 8044-based BITBUS™ board products. With high-performance 12 MHz emulation, symbolic debugging, and flexible memory mapping, the ICE-5100/044 emulator expedites all stages of development: hardware development, software development, system integration, and system test; shortening your project's time to market.

FEATURES

- Full speed to 12 MHz.
- 64KB of emulation mapped memory.
- 254 frames of execution trace.
- Symbolic debug.
- Serial link to an IBM PC XT, AT, 100% compatible.
- Four address breakpoints with in-range, out-of-range, and page breaks.
- On-line disassembler and single line assembler.
- Source code display.
- ASM-51 and PL/M-51 language support.
- Pop-up help.
- DOS shell escape.
- On-line tutorial.
- Built-in CRT based editor.
- System self-test diagnostics.
- Worldwide service and support.

September, 1989
Order Number 280818-001

ONE TOOL FOR ENTIRE DEVELOPMENT CYCLE

The ICE-5100/044 emulator speeds target system development by allowing hardware and software design to proceed simultaneously. You can develop software even before prototype hardware is finished. And because the ICE-5100/044 emulator precisely matches the component's electrical and timing characteristics, it's a valuable tool for hardware development and debug. Thus, the ICE-5100/044 emulator can debug a prototype or production system at any stage in its development, without introducing extraneous hardware or software test tools.

HIGH-SPEED, REAL-TIME EMULATION

The ICE-5100/044 emulator provides full-speed, real-time emulation up to 12 MHz. Because the emulator is fully transparent to the target system, you have complete control over hardware and software debug and system integration.

64KB of zero wait-state emulation memory is available to replace target system code memory, allowing software debug to begin even before prototype hardware is finished.

FLEXIBLE BREAKPOINTING FOR QUICK PROBLEM ISOLATION

The ICE-5100/044 emulator supports three different types of break specifications: specific address breaks on up to 64,000 possible addresses; range breaks, both within and outside a user-defined range; and page breaks, up to 256 pages on 256-byte boundaries. 254 frames of execution trace memory provide ample debug information, with each frame divided into 16 bits of program execution address and 8 bits of external event information. A maximum of four tracepoints allows qualified trace for a variety of debug conditions.

SYMBOLIC DEBUGGING FOR FAST DEVELOPMENT

Design team productivity is enhanced by the use of symbolic debug references to program line, high-level statements, and module and variable names. The terms used to develop programs are the same used for system debugging.

PATCH CODE WITHOUT RECOMPILING

Code-patching is easy with the ICE-5100/044 emulator's single-line assembler. Machine code can be disassembled to mnemonics for significantly easier debugging and project development.

EASY TO LEARN AND USE

The ICE-5100/044 is accompanied by a full tutorial that explains all system functions and provides many examples. Additional features such as on-line help, a built-in CRT-based editor, and DOS shell escape make the emulator fast and easy to use for both novice and experienced users. You can develop your own test suites or save frequently-used debug routines as debug procedures (PROCs) that can be invoked with a single command.

WORLDWIDE SERVICE AND SUPPORT

The ICE-5100/044 emulator is supported by Intel's worldwide service and support organization. In addition to an extended warranty, you can choose from hotline support, on-site system engineering assistance, and a variety of hands-on training workshops.

ELECTRICAL CONSIDERATIONS

The emulation processor's user-pin timings and loadings are identical to the 8044 component except as follows.

- Up to 25 pf of additional pin capacitance is contributed by the processor module and target adaptor assemblies.
- Pin 31, $\overline{EA}$, has approximately 32 pf of additional capacitance loading due to sensing circuitry.
- Pins 18 and 19, XTAL1 and XTAL2, respectively, have approximately 15 to 16 pf of additional capacitance when configured for crystal operation.

PROCESSOR MODULE DIMENSIONS

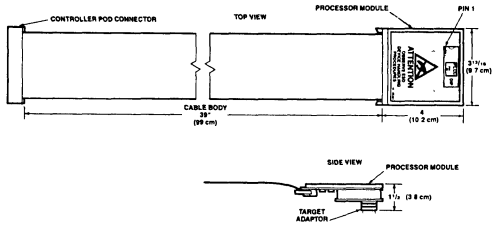


Figure 1. Processor Module Dimensions

DESIGN CONSIDERATIONS

Execution of user programs that contain interrupt routines causes incorrect data to be stored in the trace buffer. When an interrupt occurs, the next instruction to be executed is placed into the trace buffer before it is actually executed. Following completion of the interrupt routine, the instruction is executed and again placed into the trace buffer.

SPECIFICATIONS

Host Requirements:

IBM PC-XT, AT or compatible
PC-DOS 3.0 or later
512K RAM
One floppy drive and hard disk

Physical Characteristics:

The ICE-5100/044 emulator consists of the following components:

Unit	Width		Height		Length	
	Inch	Cm	Inch	Cm	Inch	Cm
Controller Pod	8.25	21.0	1.5	3.8	13.5	34.3
User Cable					39.0	99.0
Processor Module*	3.8	9.7	1.5	3.8	4.0	10.2
Power Supply	7.6	18.1	4.0	10.2	11.0	28.0
Serial Cable					144.0	360.0

*with supplied target adaptor.

Electrical Characteristics:

Power supply
100-120V or 220-240V selectable
50-60 Hz
2 amps (AC max) @ 120V
1 amp (AC max) @ 240V

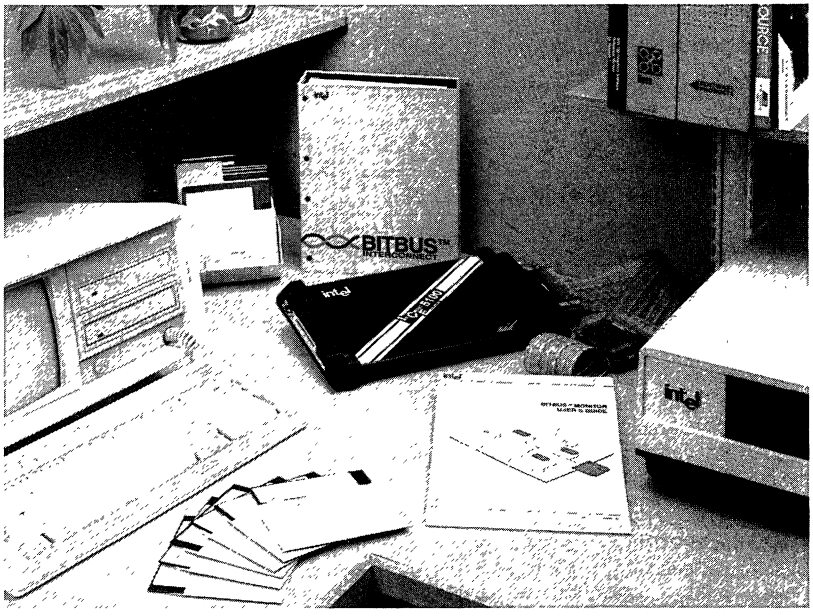
Environmental Characteristics:

Operating temperature: +10°C to +40°C (50°F to 104°F)
Operating humidity: Maximum of 85% relative humidity, non-condensing

ORDERING INFORMATION

Order Code	Description
pI044KITAD	Kit contains ICE-5100/044 user probe assembly, power supply and cables, serial cables, target adapter, crystal power accessory, emulator controller pod, emulator software, DOS host communication, ASM-51 and AEDIT text editor (requires software license).
pI044KITD	Kit contains the same components as pI044KITAD, excluding ASM-51 and the AEDIT text editor (requires software license).
pC044KITD	Conversion kit for ICE-5100/452, ICE-5100/451, or ICE-5100/252 running PC-DOS 3.0 or later, to provide emulation support for MCS-51 components (requires software license).
D86ASM51	ASM/RL 51 package for PC-DOS (requires software license).
D86PLM51	PL/M/RL 51 package for PC-DOS (requires software license).
D86EDINL	AEDIT text editor for PC-DOS.

For direct information on Intel's Development Tools, or for the number of your nearest sales office or distributor, call 800-874-6835 (U.S.). For information or literature on additional Intel products, call 800-548-4725 (U.S. and Canada).



Intel has all the software tools you'll need to implement high-performance applications using Intel BITBUS™ products. Tools include assemblers and compilers for host and BITBUS node code development, debug monitors, in-circuit emulators, and specialized BITBUS software. Intel's software tools are full-featured, easy-to-use, and help generate reliable, easily maintained code in a minimum amount of time. Intel's complete solution helps get your BITBUS-based distributed network quickly to market.

BITBUS NETWORK CONFIGURATIONS

A BITBUS network usually consists of a master (or supervisory) node and multiple remote nodes as shown on figure 1. All BITBUS host interface boards and remote control boards use the 8044 BITBUS Enhanced Microcontroller (8044BEM). The 8044BEM has built-in communications software, memory management and I/O control procedures together with a multitasking operating system. This built-in software, known as DCM44, greatly simplifies the programmer's software design task.

BITBUS networks can be configured in two ways, either as distributed I/O systems with centralized control, or as distributed control systems.

September, 1989
Order Number 280622-001

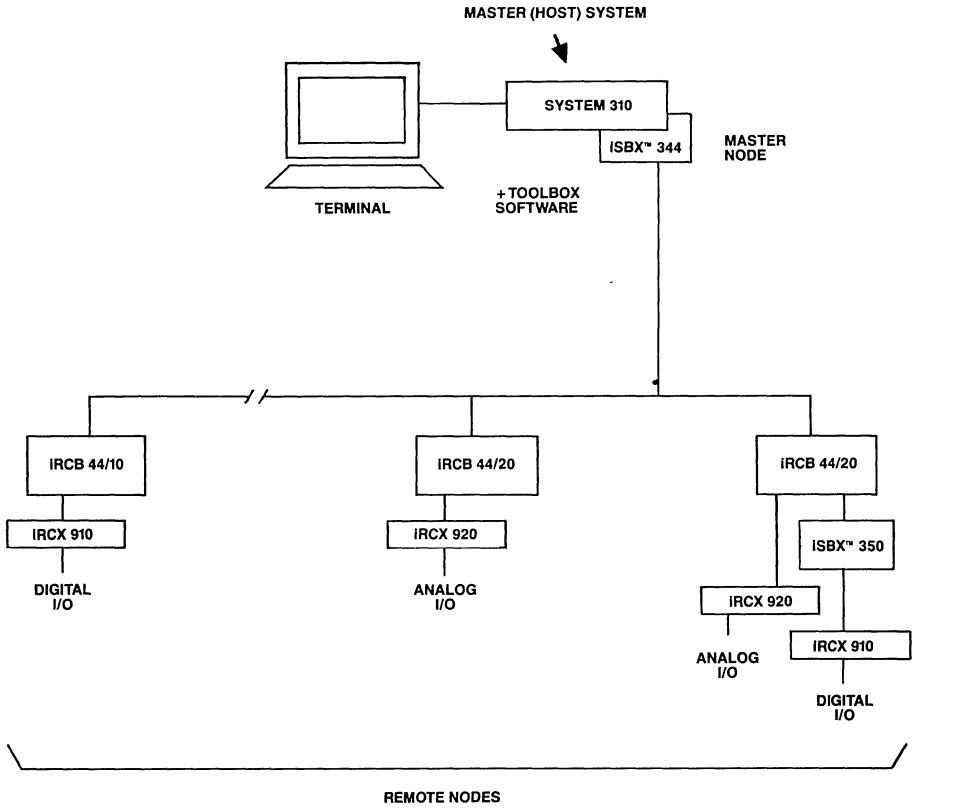


Figure 1: BITBUS™ Network

BUILT-IN RAC PROCEDURES SIMPLIFY DISTRIBUTED I/O APPLICATIONS

Distributed I/O systems are easy to design Node code (code that runs on the remote BITBUS board) is not required because the network is controlled by the master (host) system. To simplify host code, each BITBUS board comes with a built-in set of procedures known as Remote Access and Control (RAC). The master sends out commands to the nodes and uses these RAC procedures to collect data or to turn on and off motors, valves, indicator lights, and other output devices.

DISTRIBUTED CONTROL BOOSTS PERFORMANCE AND RELIABILITY

Besides using BITBUS for distributed I/O, BITBUS can also be used to implement powerful distributed control systems. With distributed control, the system can more easily control rapidly changing, complex processes (e.g. robotics) and gain the added benefit of higher network reliability that is inherent in distributed control systems.

With distributed control, each board functions as a controller performing a set of dedicated tasks. On a periodic basis, the master can send a command to a remote board to collect process control data or request that a new task start running on a remote board. The built-in DCX 51 multitasking executive on the 8044 BITBUS microcontroller allows up to 7 user tasks to run on the node at the same time. The 12 MHz 8044 8-bit microcontroller, together with the multitasking executive, allows each BITBUS remote board to easily control multiple, complex processes.

HOST SOFTWARE TOOLS

Intel's host software development tools include the BITBUS Toolbox, a wide range of compilers and assemblers for all of Intel's microprocessors, software debug monitors, and in-circuit emulators.

BITBUS™ Toolbox—The Software Tool for All Applications

The BITBUS Toolbox is a set of six software utilities that greatly simplify development of host applications software for BITBUS systems. The utilities include: the BITBUS Monitor, two procedure libraries known as the Universal BITBUS Interface and the BITBUS Interface Handlers; PC Bridge communications software; and the OBJHEX and UDI2DOS code converters.

BITBUS™ Monitor.

The BITBUS Monitor provides the designer an on-line "window" into the BITBUS network. Over 35 commands are available allowing an operator to check on the operation of various nodes, turn I/O either on or off, connect or disconnect nodes from the network, start or stop tasks running on a node, and download/upload code to/from remote boards. The Monitor is invaluable when first installing the BITBUS system, and is useful later to troubleshoot a node or the equipment connected to it.

Universal BITBUS™ Interface and BITBUS Interface Handlers.

The Universal BITBUS Interface (UBI) is similar in function to the BITBUS Monitor, except that UBI calls can be made directly from the user's host application program rather than from an operator's terminal. Procedures are included that duplicate most of the BITBUS Monitor commands. The UBI is most useful for downloading code to a node, uploading data to the host, starting and stopping tasks running on the node, and writing/reading data to/from the BITBUS boards' I/O ports.

If a programmer wants to develop custom, UBI-like procedures, the Toolbox includes the BITBUS Interface Handlers, which are a set of 4 basic procedures that support communication with a BITBUS node.

PC Bridge, OBJHEX, and UDI2DOS—The Personal Computer Gateway to BITBUS™.

The BITBUS Toolbox also includes the PC Bridge communications software and the OBJHEX conversion utility. Many BITBUS networks will use an Intel 310 system as the host in order to take advantage of the system's performance or multitasking capabilities. The PC Bridge and OBJHEX utilities enable the designer to use a PC to generate BITBUS node code, and then download the code through the 310 system to any node on the BITBUS network. The software also allows an operator to use a PC as a virtual terminal to the 310 system.

Some designers may choose to use their PC as the host system for the BITBUS network. To support these networks, the Toolbox includes the UDI2DOS utility, which is used to convert object code, developed using Intel tools, to a ".exe" format so that it will run on a PC.

The BITBUS Toolbox can be used on DOS, iRMX® 86/286, XENIX#, and iPDS™ based systems.

Host Code Compilers, Assemblers, and Other Tools

Intel's languages include PL/M, Fortran, PASCAL, C, and assembler for most of Intel's family of 8, 16, and 32-bit microprocessors. For debug support, PSCOPE, iSDM™, and Soft-Scope#, which are available in several versions, provide the programmer powerful software tools to rapidly isolate and correct faulty host code. These tools are supported on a variety of host systems, including DOS, iRMX 86/286, and XENIX.

For programmers who need an even fuller featured debug environment, Intel's I2ICE™ system combines the capabilities of an in-circuit emulator together with the PSCOPE 86 debug monitor and a 16-channel logic analyzer. The I2ICE system supports 8086, 8088, 80186, 80188, and 80286 code development. For programmers who are designing 80386 code, Intel provides the ICE™ 386 in-circuit emulator. The I2ICE and ICE 386 emulators are supported on DOS and Intel Series III/IV development systems.

SOFTWARE TOOLS FOR BITBUS™ CONTROLLER BOARDS

By adding node programs to BITBUS boards, the designer can take full advantage of the BITBUS boards' 8044 microcontroller's processing abilities. Programmed remote boards enable the designer to configure powerful, distributed control systems with a minimum investment in hardware.

Developing node code for remote BITBUS boards is just as easy as developing host code. Instead of using iAPX-based software, BITBUS boards run programs developed using "8051" tools. These tools include PL/M 51 and ASM 51 languages, RL51/LIB51 Linker/Locator/Librarian, and the ICE 5100/044 in-circuit emulator. BITBUS-specific software tools include DCS110 BITWARE and the DCS120 Programmer's Support Package.

PL/M 51 and ASM 51 Languages

The programmer can write node code using either PL/M 51 or the ASM 51 assembler. Many programs are written using PL/M 51 because the language's higher level statements reduce programming time and produce reliable, easy-to-maintain code. If necessary, speed-critical code is written using ASM 51.

* XENIX is a trademark of Microsoft Inc., Soft-Scope is a registered trademark of Concurrent Sciences, Inc

Multitasking Executive and DCS120 Maximize System Performance

Included in the 8044BEM microcontroller on every BITBUS board is the DCX 51 multitasking executive, which allows up to 7 user tasks plus the RAC task to run on the board concurrently. If the programmer is writing code for a remote board that controls several interrelated tasks, he can segment the code into separate tasks and increase overall performance by using the multitasking management provided by the executive. Twelve DCX 51 calls are available providing tasks with timing services, communications to other tasks on the board, memory management services, and the ability to dynamically create and delete running tasks.

To access DCX 51 services, Intel provides the DCS120 Programmer's Support Package, which includes an interface library to DCX 51 plus DCX 51 Procedure declaration files. To use DCS120, the programmer adds the declaration files to the source code. Then, after the source modules are compiled, the interface library is linked with the object modules and any other user libraries.

ICE 5100/044 and DCS110— The Bug Chasers

To provide debug support for node code development, Intel provides the ICE 5100/044 in-circuit emulator and the DCS110 BITWARE product. ICE 5100/044 includes an 8044 probe that plugs into the BITBUS board in place of the BITBUS 8044 microcontroller. BITWARE, which is DCM44 firmware, provides the necessary software so the ICE 5100/044 can emulate a BITBUS environment. DCS110 also includes the DCX 51 interface library and declaration files that are provided in the DCS120 product.

INTEL SOFTWARE DEVELOPMENT TOOLS—COMPLETE IN EVERY WAY

Intel provides a complete set of tools for the software designer ranging from compilers and debug monitors for the host system and BITBUS nodes to specialized BITBUS software, like the BITBUS Toolbox and BITWARE. These tools are available for a wide variety of development environments, including Intel's system 310 and the PC as shown in Table 1.

	BITBUS™ TOOLS — TOOLBOX —								NODE CODE			ICE (NOTE E)			EPROM PROG.	
	BBM	UBI	BIH	PC Bridge	OBJHEX	UDI2DOS	BITWARE	Prog. Spt. Pkg.	ASM 51	PL/M 51	RL 51, LIB 51	ICE 5100/044	ICE 44	EMV 44	iUP200A/201A with iUPF87/44A module and iPPS sw	iPDS with iUPF87/44A module and iPPS sw
Series II									C	C	C		E		X	
III			X				X	X	X	X	X	X	E		X	
IV							X	X	X	X	X	X	E		X	
iPDS	A	A	A				X	X	C	C	C			E		X
iRMX 5¼"	X	X	X	X	X		X	X	D	D	D					
8"	X	X	X	X	X		X	X	D	D	D					
XENIX 5¼"	X	X		B	X											
8"	X	X		B	X											
DOS	X	X	X	X	X	X	X	X	X	X	X	X			X	

Notes:
A iPDS uses Release 1 Toolbox
B Supports operation with XENIX. XENIX disks not required
C Down-revision version
D Available for iRMX® 86
E ICE 44 and EMV 44 have been replaced by the ICE™ 5100/044

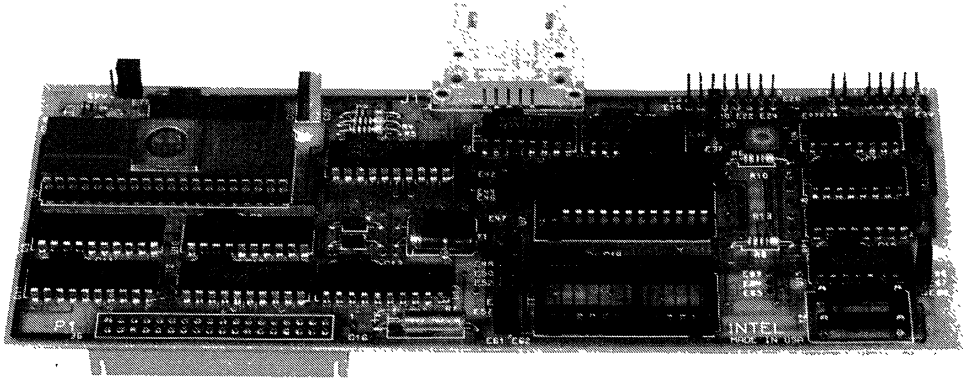
Table 1

Product	Order Code
BITBUS Toolbox	iDCS100
BITWARE	iDCS110
Programmer's Support Package	iDCS120

iSBX™ 344A
BITBUS™ INTELLIGENT MULTIMODULE™ BOARD

- **High Performance 12 MHz 8044 Controller**
- **Integral Firmware Including the iDCX 51 Executive Optimized for Real-Time Control Applications**
- **Full BITBUS™ Support**
- **2 28-Pin JEDEC Memory Sites for User's Control Functions**
- **Low Cost, Double-Wide iSBX™ BITBUS Expansion MULTIMODULE™ Board**
- **Power Up Diagnostics**

The iSBX 344A BITBUS Intelligent MULTIMODULE board is the BITBUS gateway to all Intel products that support the iSBX I/O Expansion Interface. Based on the highly integrated 8044 component (an 8-bit 8051 microcontroller and an SDLC-based controller on one chip) the iSBX 344A MULTIMODULE board extends the capability of other microprocessors via the BITBUS interconnect. With the other members of Intel's Distributed Control Modules (iDCM) family, the iSBX 344A MULTIMODULE board expands Intel's OEM microcomputer system capabilities to include distributed real-time control. Like all members of the iDCM family, the iSBX 344A MULTIMODULE board includes many features that make it well suited for industrial control applications such as: data acquisition and monitoring, process control, robotics, and machine control.



280247-1

March 1988
Order Number: 280247-002

OPERATING ENVIRONMENT

Intel's Distributed Control Modules (iDCM) product family contains the building blocks to implement real-time distributed control applications. The iDCM family incorporates the BITBUS interconnect to provide standard high speed serial communication between microcontrollers. The iDCM hardware products: including the iSBX 344A MULTIMODULE board, iPCX 344A board and all iRCB BITBUS Remote Controller Boards communicate in an iDCM system via the BITBUS interconnect as shown in Figure 1.

As a member of the iDCM product line the iSBX 344A MULTIMODULE board fully supports the BITBUS microcontroller interconnect. Typically, the iSBX 344A MULTIMODULE board would be part of a node (master or slave) on the BITBUS interconnect in an iDCM system. As shown in Figure 2 the iSBX 344A MULTIMODULE board plugs into any iSBC® board with an iSBX connector.

The iSBX 344A MULTIMODULE board is the hardware interface between Intel's MULTIBUS® and the BITBUS environment. With this interface the user can harness the capabilities of other Intel microprocessors e.g. 80386, 80286, 80186, 8086, 80188, 8088 in a iDCM system or extend an existing MULTIBUS system with the iDCM family.

MULTIBUS® Expansion

Typically, MULTIBUS iSBC boards have a maximum of two iSBX I/O expansion connectors. These connectors facilitate addition of one or two iSBX I/O MULTIMODULE boards with varying numbers of I/O lines. The iSBX 344A MULTIMODULE board increases the number of I/O lines that can be accommodated by a MULTIBUS system by at least an order of magnitude.

Extending BITBUS™/iDCM System Processing Capability

The iSBX 344A MULTIMODULE board allows utilization of other processors in a iDCM system to accommodate particular application requirements. The MULTIMODULE board is compatible with any iSBX connector so that any board having a compatible connector can potentially enhance system performance. Intel's DCS100 BITBUS Toolbox Software provides easy to use high performance software interfaces for iSBC boards. The iSBC 86/35, 286/12, and 188/48 boards are a few examples. Custom configurations are also possible with user customized software.

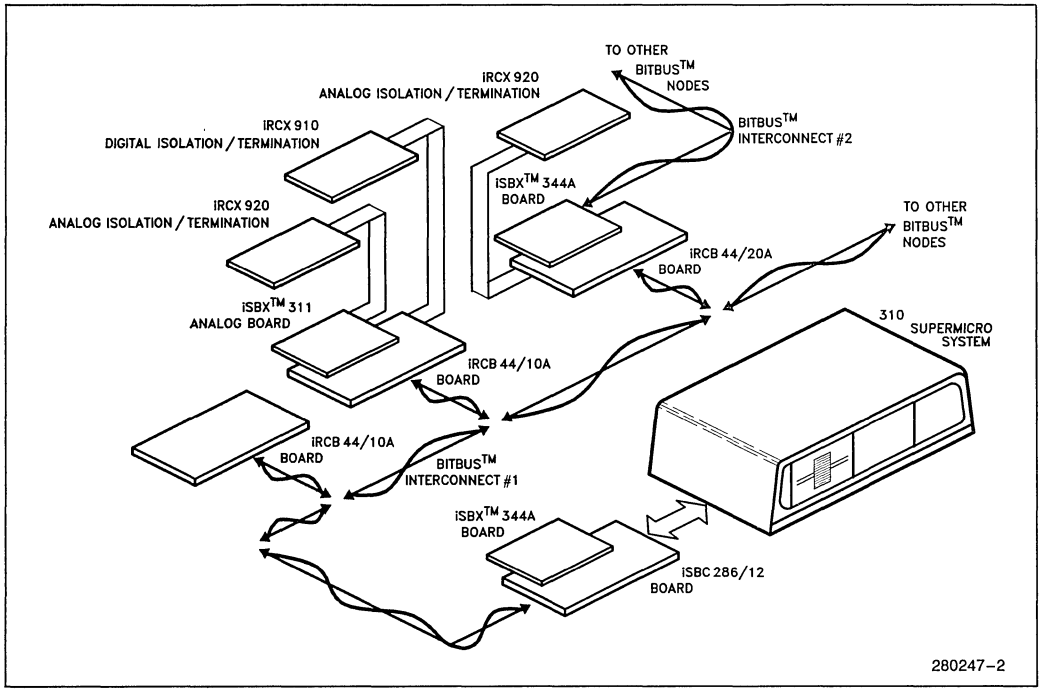


Figure 1. iDCM Operating Environment

ARCHITECTURE

Figure 3 illustrates the major functional blocks of the iSBX 344A board: 8044 BITBUS Enhanced Microcontroller (BEM), memory, BITBUS microcontroller interconnect, Byte FIFO interface, initialization and diagnostic logic.

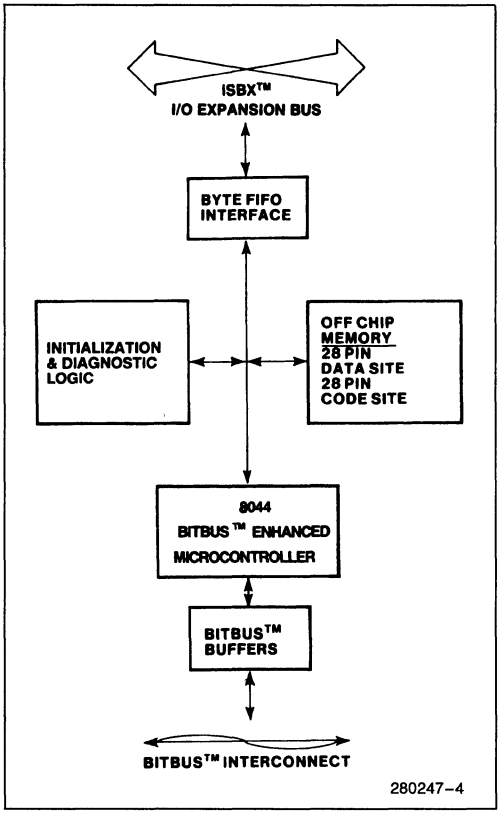


Figure 3. iSBX™ 344A Block Diagram

iDCM Controller

The heart of the iSBX 344A MULTIMODULE board's controlling and communication capability is the highly integrated 12 MHz 8044 microcontroller. The 8044 consists of the advanced 8-bit, 8051 microcontroller and a SDLC-based controller called the Serial Interface Unit (SIU). This dual processor architecture allows complex control and high speed communication to be realized cost effectively.

The 8044 BEM microcontroller also includes built-in firmware known as DCM44. This firmware includes a set of functions called Remote Access and Control (RAC), a preconfigured version of the DCX51 Executive, communications software, and a power-up test procedure.

Memory

The iSBX 344A MULTIMODULE board memory consists of two internal and external memory. Internal memory is located in the on-chip memory of the iDCM controller. The iDCX 51 Executive and the remaining 8044 BEM firmware ration this resource. However, eight bytes of bit addressable internal memory are reserved for the user. Ample space is reserved for user programs and data in the iSBX 344A MULTIMODULE board external memory.

Two 28-pin JEDEC sites comprise the iSBX 344A MULTIMODULE board external memory. One site has been dedicated for data; the other for code. Table 1 lists the supported memory devices for each site. Intel's 2764 and 27128 are examples. The user may choose one of two memory configurations and specify different memory sizes by placing the proper jumpers at system initialization. The most flexible configuration option provides the user with access to the code site for program download or upload. This feature ensures expansion of an existing system is easily accommodated. For example, the addition of another conveyor to a material handling system would require adding another controller or controllers and changes to existing applications code and addition of new code.

Table 1. Supported Memory Devices

Device	Data Site	Code Site
4K x 8–64K x 8 EPROM/ROM	No	Yes
2K x 8–32K x 8 SRAM	Yes	Yes
2K x 8–16K x 8 NVRAM and E2PROM	No	Yes

BITBUS™ Microcontroller Interconnect

The iSBX 344A MULTIMODULE board fully supports the BITBUS microcontroller interconnect. The BITBUS interconnect is a serial bus optimized for control applications. The interconnect supports both synchronous and self-clocked modes of operation. These modes of operation are selectable dependent on application requirements as are the transmission rates. Table 2 shows different combinations of modes of operations, transmission rates, and distances. The SDLC-based protocol, BITBUS message format, and compatibility with Intel's other software and hardware products comprise the remainder of this established architecture. These features contribute to BITBUS reliability and usefulness as a microcontroller interconnect.

The BITBUS connection consists of one or two differential pair(s) of wires. The BITBUS interface of the iSBX 344A MULTIMODULE board consists of a half-duplex RS 485 transceiver and an optional clock source for the synchronous mode of operation.

Byte FIFO Interface

The Byte FIFO Interface on the iSBX 344A MULTIMODULE board implements the required hardware buffering between the 8044 BEM and an extension. An extension is defined as a device attached to the iSBX I/O expansion interface on the iSBX 344A MULTIMODULE board. In an iDCM system, an example of an extension is an iSBC 286/12 board which may be considered the host board in a MULTIBUS system. When used with the software handlers in the BITBUS Toolbox, implementation of this interface is complete.

For particular applications, the user may wish to develop a custom software interface to the extension or host board. On the iSBX 344A MULTIMODULE board side of the interface the iDCM firmware automatically accepts messages for the FIFO. No user code is required, increasing the time available for application system development.

The Byte FIFO supports both byte and message transfer protocol in hardware via three register ports: data, command, and status. The extension side supports polled, interrupt, and limited DMA modes of operation (e.g. 80186 type DMA controllers).

Initialization and Diagnostic Logic

Like the other members of Intel's Distributed Control Modules (iDCM) product line, the iSBX 344A MULTIMODULE board includes many features which make it well suited for industrial control applications. Power up diagnostics is just one of these features. Diagnostics simplify system startup considerably, by immediately indicating an 8044 BEM or external bus failure. The LEDs used for power up diagnostics are available for user diagnostics after power up as well as to further contribute to reliable operation of the system.

Initial iSBX 344A MULTIMODULE board parameters are set by positioning jumpers. The jumpers determine the BITBUS mode of operation: synchronous, self-clocked, transmission rate, and address of the iSBX module in the BITBUS system. This minimizes the number of spare boards to be stocked for multiple nodes, decreasing stocking inventory and cost.

INTEGRAL FIRMWARE

Resident firmware located in the 8044 BEM includes: a pre-configured iDCX 51 Executive for user program development; a Remote Access and Control (RAC) function that enables user communication and control of different microcontrollers and I/O points; a communications gateway to connect the BITBUS interconnect, iSBX bus, and iDCX 51 Executive tasks; and power up diagnostics.

Table 2. BITBUS™ Microcontroller Interconnect Modes of Operation

	Speed Kb/s	Maximum Distance Between Repeaters M/ft	Maximum # Nodes Per Segment	Maximum # Repeaters Between a Master and Any Slave
Synchronous	500–2400	30/100	28	0
Self Clocked	375 62.5	300/1000 1200/4000	28 28	2 10

Segment: Distance between master and repeater or a repeater and a repeater.
Synchronous mode requires user supplied crystal.

The iDCX 51 Executive is an event-driven software manager that can respond to the needs of multiple tasks. This real-time multitasking executive provides: task management, timing, interrupt handling, and message passing services. Table 3 shows the iDCX 51 calls. Both the executive and the communications gateway allow for the addition of up to seven user tasks at each node while making BITBUS operations transparent.

The Remote Access and Control Function is a special purpose task that allows the user to transfer commands and program variables to remote BITBUS controllers, obtain the status of a remote I/O line(s), or reverse the state of a remote I/O line. Table 4 provides a complete listing of the RAC services. No user code need be written to use this function.

The services provided by the iSBX 344A MULTIMODULE board integral firmware simplify the development and implementation of complex real-time control application systems. All iDCM hardware products contain integral firmware thus supplying the user with a total system solution.

DEVELOPMENT ENVIRONMENT

Intel provides a complete development environment for the iSBX 344A MULTIMODULE board. Software development support consists of: the 8051 Software Development Package, the DCS100 BITBUS Toolbox Host Software Utilities, the DSC110 Bitware for ICE™ Support, and the DCS120 Programmer's Support Package. The 8051 Software Development Package provides the RL 51 Linker and Relocator Program, and ASM 51. PL/M 51 is also available. Hardware tools consist of the In-Circuit Emulator (ICE 5100/044).

Table 3. iDCX 51 Calls

Call Name	Description
TASK MANAGEMENT CALLS	
RQ$CREATE$TASK	Create and schedule a new task.
RQ$DELETE$TASK	Delete specified task from system.
RQGETFUNCTION$IDS	Obtain the function IDs of tasks currently in the system.
INTERTASK COMMUNICATION CALLS	
RQ$ALLOCATE	Obtain a message buffer from the system buffer pool.
RQ$DEALLOCATE	Return a message buffer to the system buffer pool.
RQ$SEND$MESSAGE	Send a message to specified task.
RQ$WAIT	Wait for a message event.
MEMORY MANAGEMENT CALLS	
RQGETMEM	Get available SMP memory.
RQ$RELEASE$MEM	Release SMP memory.
INTERRUPT MANAGEMENT CALLS	
RQ$DISABLE$INTERRUPT	Temporarily disable an interrupt.
RQ$ENABLE$INTERRUPT	Re-enable an interrupt.
RQ$WAIT	Wait for an interrupt event.
TIMER MANAGEMENT CALLS	
RQSETINTERVAL	Establish a time interval.
RQ$WAIT	Wait for an interval event.

Table 4. RAC Services

RAC Service	Action Taken by Task 0
RESET_STATION	Perform a software reset.
CREATE_TASK	Perform an RQ$CREATE$TASK system call.
DELETE_TASK	Perform an RQ$DELETE$TASK system call.
GET_FUNCTION_ID	Perform an RQGETFUNCTION$IDS call.
RAC_PROJECT	Suspend or resume RAC services.
READ_I/O	Return values from specified I/O ports.
WRITE_I/O	Write to the specified I/O ports.
UPDATE_I/O	Update the specified I/O ports.
UPLOAD_MEMORY	Return the values in specified memory area.
DOWNLOAD_MEMORY	Write values to specified memory area.
OR_I/O	OR values into specified I/O ports.
AND_I/O	AND values into specified I/O ports.
XOR_I/O	XOR values into specified I/O ports.
READ_INTERNAL	Read values at specified internal RAM areas.
WRITE_INTERNAL	Write values to specified internal RAM areas.
NODE_INFO	Return device related information.
OFFLINE	Set node offline.
UPLOAD_CODE	Read values from code memory space.
DOWNLOAD_CODE	Write values to specified EEPROM memory.

NOTE:
Internal memory locations are included in the 192 bytes of data RAM provided in the microcontroller. External memory refers to memory outside the microcontroller — the 28-pin sockets of the iSBX 344A module and the iRCB 44/10A board. Each RAC Access Function may refer to multiple I/O or memory locations in a single command.

SPECIFICATIONS

CPU

8044 BITBUS Enhanced Microcontroller (BEM)

Word Size

Instruction: 8 bits

Data: 8 bits

Processor Clock 12 MHz

Instruction Execution Times

1 μs 60% instructions

2 μs 40% instructions

4 μs Multiply & Divide

Memory Capacity/Addressing

iDCM Controller: Up to 64 Kbytes code

Address Range

	Option A	Option B
External Data Memory	0000H–7FFFH	0000H–7FFFH
External Code Memory	1000H–0FFFFH	8000H–0FEFFH
Internal Code Memory	0000H–0FFFH	0000H–0FFFH

Option A: Supports maximum amount of external EPROM code memory.
Option B: Supports downloading code into external RAM or EEPROM memory.

Terminations

Sockets provided on board for ¼ Watt 5% Carbon type resistors. Resistor value to match characteristic impedance of cable as closely as possible—120Ω or greater.

Message Size

54 bytes max

8044 BITBUS™ Enhanced Microcontroller (8044 + Firmware) I/O Addressing as Viewed from the 8044

Function	Address	Read	Write	Bit	Comments
Data	FF00H	✓	✓		
Command	FF01H	✓	✓		Write sets command to extension — Read clears command from extension
Status					
-RFNF*	B3H	✓		✓	Also INT1 Input
-TFNE*	B2H	✓		✓	Also INT0 Input
-TCMD*	92H	✓		✓	
LED #1	90H	✓	✓	✓	
LED #2	91H	✓	✓	✓	
RDY/NE*	B4H	✓	✓	✓	
Node Address	FFFFH	✓			
Configuration	FFFEH	✓			

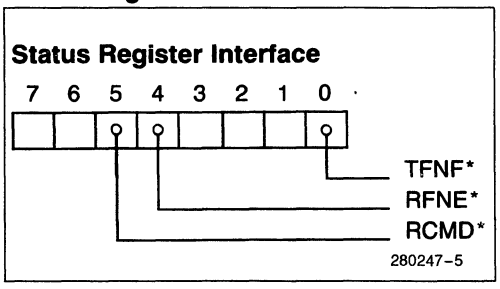
iSBX™ 344A MULTIMODULE™ Board I/O Addressing as Viewed from the iSBX™ 344A MULTIMODULE™ Board

Register Function	Address	Comments
Data	Base'	Read/Write
Command	Base' + 1	Write sets command from extension Read clears command to extension
Status	Base' + 2	Read Only

Interrupt/DMA Lines

Signal	Location	Interface Option
RINT	MDRQ/MINT0	INT
TINT	MINT1	INT
RCMI	OPT0	INT or DMA
RDRQ	MDRQ/MINT0	DMA
TDRQ	MINT1	DMA

Status Register Interface



Connector Options

10 Pin Plug

Flat Cable: 3M 3473-6010, TB Ansley 609-1001M, or equal

Discrete Wire: BERG 65846-007, ITT Cannon 121-7326-105, or equal

Pinout

Pin	Signal
1	+12V
2	+12V
3	GND
4	GND
5	DATA*
6	DATA
7	DCLK*/RTS*
8	DCLK/RTS
9	RGND
10	RGND

Electrical Characteristics

Interfaces

iSBX™ I/O Expansion Bus: supports the standard I/O Expansion Bus Specification with compliance level IEEE 959.

Memory Sites: Both code and data sites support the standard 28-pin JEDEC site.

BITBUS™ Interconnect: Fully supported synchronous mode at 2.4 Mbits/sec and self clocked mode for 375 kbits/sec and 62.5 kbits/sec

The iSBX 344A MULTIMODULE board presents one standard load to the BITBUS bus

Power Requirements

0.9A at +5V ±5% (does not include power to the memory devices)

Physical Characteristics

Double-wide iSBX™ MULTIMODULE™ Form Factor

Dimensions

Height: 10.16 mm (0.4 in) maximum component height

Width: 63.5 mm (2.50 in)

Length: 190.5 mm (7.50 in)

Weight: 113 gm (4 ounces)

Environmental Characteristics

Operating Temperature: 0°C to 55°C at 200 Linear Feet/Minute Air Velocity

Humidity: 90% non-condensing

Reference Manual (NOT Supplied)

148099— iSBX 344A Intelligent BITBUS Interface Board User's Guide

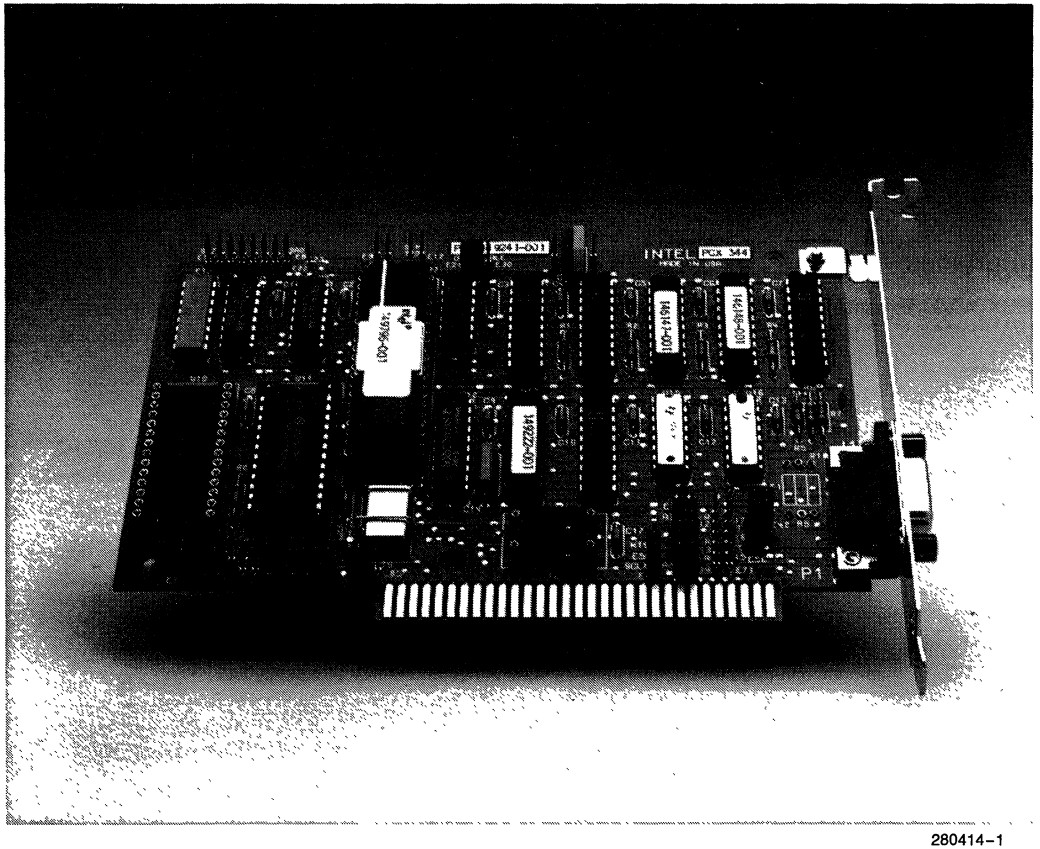
Ordering Information

Part Number	Description
iSBX 344A	BITBUS Intelligent MULTIMODULE board

iPCX 344A
BITBUS™ IBM* PC INTERFACE BOARD

- **High Performance 12 MHz 8044 Single-Chip Microcontroller**
- **Integral Firmware Optimized for Real-Time Control Applications Using the BITBUS™ Interconnect**
- **Fully Supports Intel's Complete Remote Control Board Product Line (iRCB)**
- **Compatible with Intel's DOS-Based Development Tools**
- **External Memory Sites for User's Control Programs**
- **IBM PC System Form Factor Board**
- **Power Up Diagnostics**

The iPCX 344A BITBUS IBM PC INTERFACE board provides the BITBUS gateway to IBM's family of Personal and Industrial Computers. Based on Intel's highly integrated 8044 (an 8051 microcontroller and an SDLC controller on one chip) the iPCX 344A IBM PC INTERFACE board extends the real-time control capability of the IBM PC via the BITBUS interconnect. The PC system performs the human interface functions for the BITBUS interconnect. Like all members of Intel's Distributed Control Modules (iDCM) family, the iPCX 344A IBM PC INTERFACE board includes features that make it well suited for Industrial Control applications such as: data acquisition and monitoring, process control, machine control, and statistical process control (SPC).



280414-1

*IBM is a trademark of International Business Machines.

March 1988
Order Number: 280414-002

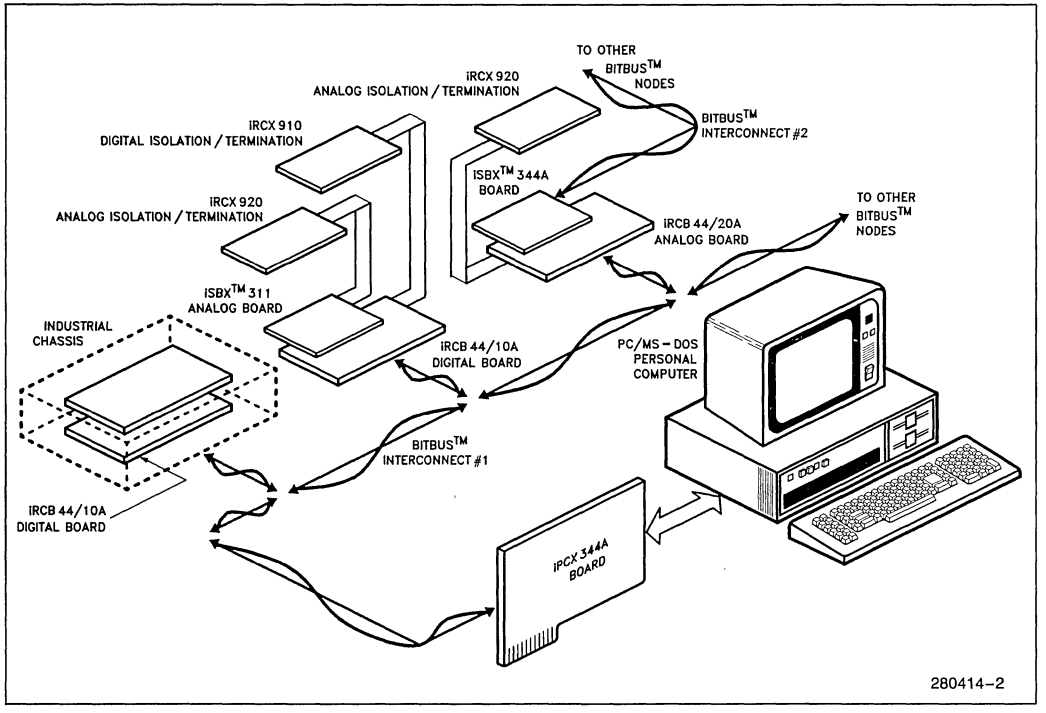


Figure 1. iDCM Operating Environment

OPERATING ENVIRONMENT

Intel's Distributed Control Modules (iDCM) product family provides the building blocks to implement real-time distributed I/O control applications. All of the iDCM family utilizes the BITBUS interconnect to provide standard high speed serial communication between microcontrollers. The iDCM hardware products: including the iPCX 344A board, iSBX™ 344A MULTIMODULE™ board and all iRCB BITBUS Remote Controller Boards communicate in an iDCM system via the BITBUS interconnect as shown in Figure 1.

As a member of the iDCM Product line, the iPCX 344A IBM PC INTERFACE board fully supports the BITBUS microcontroller interconnect. Typically, the iPCX 344A IBM PC System INTERFACE board will be part of a node (master or slave) on the BITBUS interconnect. The iPCX 344A board plugs into the PC add-in slot.

The iPCX 344A IBM PC INTERFACE board is the hardware interface between the PC system and the BITBUS environment. With this interface the user can utilize the human interface and application software of the PC and extend the I/O range of the PC to include real-time distributed control.

ARCHITECTURE

Figure 2 illustrates the major functional blocks of the iPCX 344A IBM PC INTERFACE board: 8044 BITBUS ENHANCED MICROCONTROLLER, memory, BITBUS interconnect, PC System Interface, and initialization/diagnostic logic.

Memory, mode of operation, and bus transmission rate options are easily selected by the user, thereby decreasing inventory levels and associated costs.

8044 BITBUS™ Enhanced Microcontroller (BEM)

The source of the iPCX 344A IBM PC INTERFACE board's controlling and communication capability is Intel's highly integrated 12 MHz 8044 microcontroller. The 8044 consists of the advanced 8-bit, 8051 microcontroller and a SDLC controller called the Serial Interface Unit (SIU). This dual processor architecture provides complex control and high speed communications in a cost-effective, single chip implementation.

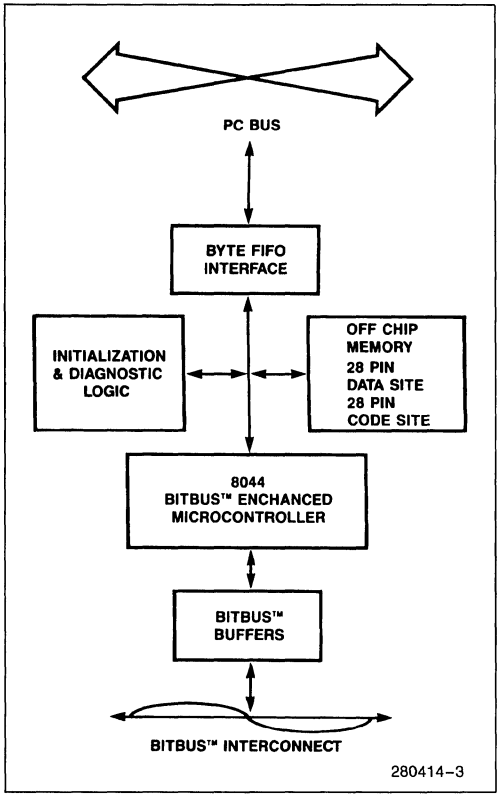


Figure 2. iPCX 344A Block Diagram

Another essential part of the 8044 controller is the integral firmware residing on-chip to implement the BITBUS interface. In the operating environment of the iPCX 344A board, the 8044's SIU acts as an SDLC controller offloading the on-chip 8051 microcontroller of communication tasks; freeing the 8051 to concentrate on real-time control.

The 8044 BEM (8044 microcontroller and on-chip firmware) provides in one package a simple user interface, and high performance communications and control capabilities to efficiently and economically build a complex control system.

Memory

The iPCX 344A IBM PC System INTERFACE board contains both internal and external memory. Internal memory is located in the on-chip memory of the 8044 BEM. The BITBUS firmware includes Intel's powerful iDCX 51, real-time, multitasking, executive. Eight bytes of bit-addressable internal memory are reserved for the user. Additional space is reserved for user programs and data in the board's external memory.

Two 28-pin JEDEC sites comprise the iPCX 344A board's external memory. One site is dedicated to data; the other to code. Table 1 lists the supported memory devices for each site. Intel's 2764 and 27128 are examples. The user can choose one of two memory configurations and specify different memory sizes by configuring the correct jumpers. This configurability provides the user with access to the code site for program download or upload and ensures that an existing system is easily expanded.

Table 1. Supported Memory Devices

Device	Data Site	Code Site
4K x 8–64K x 8 EPROM/ROM	No	Yes
2K x 8–32K x 8 SRAM	Yes	Yes
2K x 8–16K x 8 NVRAM and E^2PROM	No	Yes

BITBUS™ Microcontroller Interconnect

The iPCX 344A IBM PC INTERFACE board fully supports the BITBUS microcontroller interconnect. The BITBUS interconnect is a serial bus optimized for control applications and supports both synchronous and self-clocked modes of operation. Each mode of operation and the different transmission rates are jumper selectable dependent on application requirements.

Table 2 shows different combinations of mode of operation, transmission rate, and distance. The SDLC protocol, BITBUS message format, and compatibility with Intel's other software and hardware products comprise the remainder of this established architecture. These features contribute to BITBUS reliability and usefulness as a microcontroller interconnect.

The BITBUS connection consists of one or two differential user selected pair(s) of wires. The BITBUS interface on the iPCX 344A board consists of a half-duplex RS485 transceiver and an optional clock source for the synchronous mode of operation.

PC System Interface

The iPCX 344A board will operate in any IBM PC XT, PC AT, or compatible system that meets the following requirements:

— An IBM PC, PC XT with an oscillator running at 4.77 MHz (processor running at 4.77 MHz also)

— An IBM PC AT with an oscillator running at 12 or 16 MHz (processor running at 6 or 8 MHz)

— An available I/O channel with addresses that are *not* used by any other boards in the system in the range of 200H to 3FFH on even addresses

— At least one available system interrupt (required ONLY if running the iPCX 344A board in interrupt mode; user selectable from PC Interrupts 2, 3, 4, 5, 6, or 7)

All IBM guidelines have been followed to ensure complete IBM PC system compatibility.

Initialization and Diagnostic Logic

Like the other members of Intel's Distributed Control Modules (iDCM) product line, the iPCX 344A BITBUS IBM PC INTERFACE board includes many features making it well suited for industrial control applications. Power on diagnostics simplify system startup considerably by immediately indicating an 8044 BEM or external bus failure.

INTEGRAL FIRMWARE

The iPCX 344A BITBUS PC-BUS INTERFACE board contains resident firmware located in the 8044 BITBUS ENHANCED MICROCONTROLLER. This on-chip firmware consists of: a pre-configured iDCX 51 Executive for real-time, multitasking control; DCM 44, a Remote Access and Control (RAC) program that enables BITBUS communication and control of I/O points on the BITBUS interconnect; and power up diagnostics.

The iDCX 51 Executive is an event-driven software manager that can respond to the needs of multiple tasks. This real-time multitasking executive provides: task management, timing, interrupt handling, and message passing services. Table 3 shows the iDCX 51 operating system calls. The executive supports up to seven user tasks at each node while making BITBUS operations transparent.

Remote Access and Control (RAC) is a special purpose task that allows the user to transfer commands and program variables to and from BITBUS controllers to obtain the status of I/O or data line(s), or reverse the state of an I/O line or read and write memory, etc. No user code need be written to use this function. See Table 4 for a complete listing of RAC services.

The services provided by the iPCX 344A board's integral firmware simplify the development and implementation of complex real-time control systems.

DEVELOPMENT ENVIRONMENT

Intel provides a variety of development environments for BITBUS applications. Intel's Development Systems and OEM Systems Handbooks provide details on the following development tools.

— BITBUS TOOLBOX—BITBUS Monitor and Interface Handlers

— ASM/PLM 51—Low and High level languages for application code generation on 8044

Table 2. BITBUS™ Microcontroller Interconnect Modes of Operation

	Speed Kb/s	Maximum Distance Between Repeaters M/ft	Maximum # Nodes Per Segment*	Maximum # Repeaters Between a Master and Any Slave
Synchronous	500–2400	30/100	28	0
Self Clocked	375 62.5	300/1000 1200/4000	28 28	2 10

*Segment: Distance between master and repeater or repeater and repeater.
Synchronous mode requires user supplied crystal.

Table 3. iDCX 51 Systems Calls

Call Name	Description
TASK MANAGEMENT CALLS	
RQ$CREATE$TASK	Create and schedule a new task.
RQ$DELETE$TASK	Delete specified task from system.
RQGETFUNCTION IDS	Obtain the function IDs of tasks currently in the system.
INTERTASK COMMUNICATION CALLS	
RQ$ALLOCATE	Obtain a message buffer from the system buffer pool.
RQ$DEALLOCATE	Return a message buffer to the system buffer pool.
RQ$SEND$MESSAGE	Send a message to specified task.
RQ$WAIT	Wait for a message event.

Table 3. iDCX 51 Systems Calls (Continued)

Call Name	Description
MEMORY MANAGEMENT CALLS	
RQGETMEM	Get available SMP memory.
RQ$RELEASE$MEM	Release SMP memory.
INTERRUPT MANAGEMENT CALLS	
RQ$DISABLE$INTERRUPT	Temporarily disable an interrupt.
RE$ENABLE$INTERRUPT	Re-enable an interrupt.
RQ$WAIT	Wait for an interrupt event.
TIMER MANAGEMENT CALLS	
RQSETINTERVAL	Establish a time interval.
RQ$WAIT	Wait for an interval event.

Table 4. RAC Services

RAC Service	Action Taken by Task 0
RESET_STATION	Perform a software reset.
CREATE_TASK	Perform an RQ$CREATE$TASK system call.
DELETE_TASK	Perform an RQ$DELETE$TASK system call.
GET_FUNCTION_ID	Perform an RQGETFUNCTION$IDS call.
RAC_PROJECT	Suspend or resume RAC services.
READ_I/O	Return values from specified I/O ports.
WRITE_I/O	Write to the specified I/O ports.
UPDATE_I/O	Update the specified I/O ports.
UPLOAD_MEMORY	Return the values in specified memory area.
DOWNLOAD_MEMORY	Write values to specified memory area.
OR_I/O	OR values into specified I/O ports.
AND_I/O	AND values into specified I/O ports.
XOR_I/O	XOR values into specified I/O ports.
READ_INTERNAL	Read values at specified internal RAM areas.
WRITE_INTERNAL	Write values to specified internal RAM areas.
NODE_INFO	Return device related information.
OFFLINE	Set node offline.
UPLOAD_CODE	Read values from code memory space.
DOWNLOAD_CODE	Write values to specified EEPROM memory.

SPECIFICATIONS

CPU

8044 BITBUS Enhanced Microcontroller (BEM)

Word Size

Instruction—8 bits
Data—8 bits

Processor Clock

12.0 MHz

Instruction Execution Time

1 μs 60% instructions
2 μs 40% instructions
4 μs Multiply and Divide

Memory Capacity Addressing

iDCM Controller: Up to 64 Kbytes code.

Device	Data	Code
EPROM/ROM		
4K x 8—64K x 8	No	Yes
SRAM		
2K x 8—32K x 8	Yes	Yes
NVRAM and E2PROM		
2K x 8—16K x 8	No	Yes

External I/O Space

0FF00H–0FFFFH (mapped into data memory space)

Termination

Minimum 120Ω each end of BITBUS interconnect with user supplied resistors

Address Ranges

	Option A	Option B
External Data Memory Site	0000H–7FFFH	0000H–7FFFH
External Code Memory Site	1000H–0FFFFH (0000H–0FFFFH If EA Active)	8000H–0FEFFH
Internal Code Memory	0000H–0FFFH	0000H–0FFFH

Option A: Supports maximum amount of external EPROM code memory.
Option B: Supports downloading code into external RAM or EEPROM memory.

Message Size:

Up to 54 bytes

Connectors

Standard 9-pin—D Subminiature socket

Physical Characteristics

IBM PC ADD-ON FORMAT

Height: 3.98 in.

Depth: 6 in.

Interfaces

BITBUS Interconnect: Fully supports synchronous mode at 500 Kbps to 2.4 Mbs and self-clocked modes at 375 Kbs or 62.5 Kbs

Note: On-board ALE clock supports 1 Mbps synchronous operation. All other synchronous mode speeds require user-supplied 2.0 MHz–9.6 MHz crystal.

PC System: Two unidirectional, one-byte-deep, nine-bit FIFO buffers (ninth bit distinguishes between data and command)

Power Requirements

0.9A at +5V ±5% (memory not included)

Environmental Requirements

Operating Temperature: 16°C to 32°C at no air flow
0°C to 55°C at 200 Linear Feet/Minute air velocity

Operating Humidity: 90% Noncondensing

Storage Temperature: −40°C to +70°C

Storage Humidity: 95% Noncondensing

REFERENCE MANUAL

149235-001— iPCX 344A BITBUS IBM PC System Interface Board User's Guide

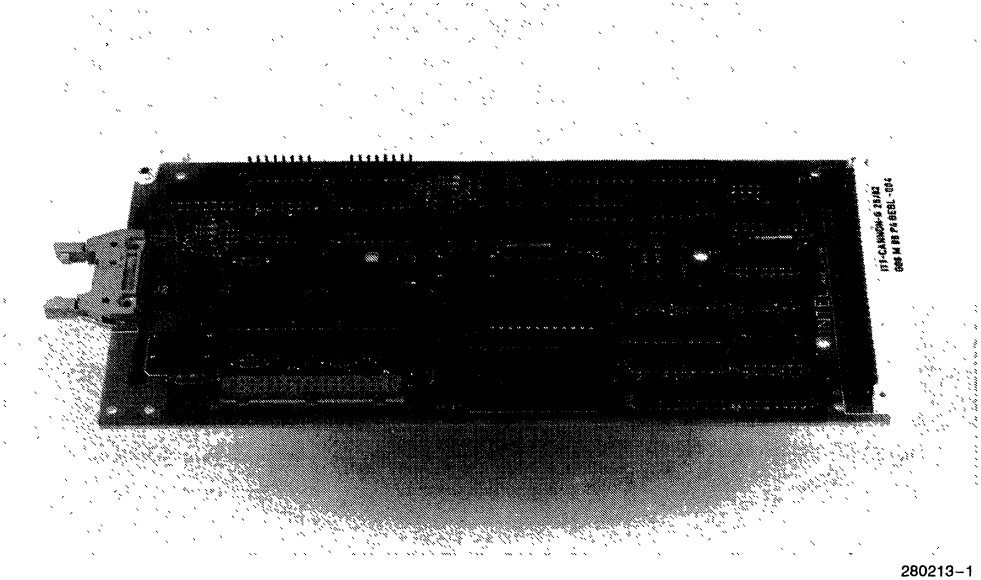
ORDERING INFORMATION

Part Number	Description
iPCX 344A	BITBUS IBM PC System INTERFACE Board

iRCB 44/10A
BITBUS™ DIGITAL I/O REMOTE CONTROLLER BOARD

- **High Performance 12 MHz 8044 Controller**
- **Integral Firmware: iDCX Executive, Optimized for Real-Time Control**
- **Full BITBUS™ Support**
- **Standard Industrial Packaging: Eurocard, DIN Connector**
- **2 28-Pin JEDEC Memory Sites for User's Control Functions**
- **I/O Expansion with 8-Bit iSBX™ Connector**
- **Programmable Control/Monitoring Using 24 Digital I/O Lines**
- **Power Up Diagnostics**
- **Compatible with iRCX 910 Digital Signal Isolation and Termination Module**

The iRCB 44/10A BITBUS™ Digital I/O Remote Controller Board is an intelligent real-time controller and a remote I/O expansion device. Based on the highly integrated 8044 component (an 8 bit 8051 microcontroller and an intelligent SDLC-based controller on one chip) the iRCB 44/10A board provides high performance control capability at low cost. The iRCB 44/10A board can expand Intel's OEM microcomputer system capabilities to include distributed real-time control. Like all members of the iDCM family, the iRCB 44/10A board is well suited for industrial control applications such as data acquisition and monitoring, process control, robotics, and machine control.



280213-1

March 1988
Order Number: 280213-003

OPERATING ENVIRONMENT

Intel's Distributed Control Modules (iDCM) product family contains the building blocks to implement real-time distributed control applications. The iDCM family incorporates the BITBUS interconnect to provide standard high speed serial communication between microcontrollers. The iDCM hardware products, which include the iPCX 344A board, iSBX 344A MULTIMODULE™ board and the iRCB 44/10A BITBUS Remote Controller Board (and other iRCB boards), communicate in an iDCM system via the BITBUS interconnect as shown in Figure 1.

The iRCB 44/10A board can be used as an intelligent remote controller or an I/O expansion device. When performing as an intelligent controller the iRCB 44/10A board not only monitors the status of multiple process points, but it can execute varied user supplied control algorithms. When functioning as an I/O expansion device, the iRCB 44/10A board simply collects data from multiple I/O ports and transmits this information via the BITBUS or iSBX bus interface to the system controller for analysis or updating purposes.

As a member of the iDCM product line, the iRCB 44/10A board fully supports the BITBUS microcontroller interconnect. Typically, the iRCB 44/10A board would be a node in a BITBUS system. The iRCB 44/10A board could be a master or slave node. (The BITBUS system supports a multidrop configuration: one master, many slaves.)

ARCHITECTURE

Figure 2 illustrates the major functional blocks of the iRCB 44/10A board: 8044 BITBUS Enhanced Microcontroller, memory, BITBUS microcontroller interconnect, parallel I/O, iSBX expansion, initialization and diagnostic logic.

8044 BITBUS™ Enhanced Microcontroller

The heart of the iRCB 44/10A board's controlling and communication capability is the highly integrated 12 MHz 8044 microcontroller. The 8044 consists of the advanced 8-bit 8051 microcontroller and a SDLC controller called the Serial Interface Unit (SIU). This dual processor architecture allows complex control and high speed communication functions to be realized cost effectively. The 8044's SIU acts as a SDLC-based controller which offloads the on-chip 8051 microcontroller of communication tasks; freeing the 8051 to concentrate on real-time control.

The 8044 BEM microcontroller also includes, in firmware, a set of procedures known as Remote Access and Control (RAC), a preconfigured version of the DCX 51 Executive, communications software, and power-up diagnostics.

The BEM (8044 microcontroller and on-chip firmware) provides, in one package, a simple user interface, and high performance communications and control capabilities to efficiently and economically build a complex control system.

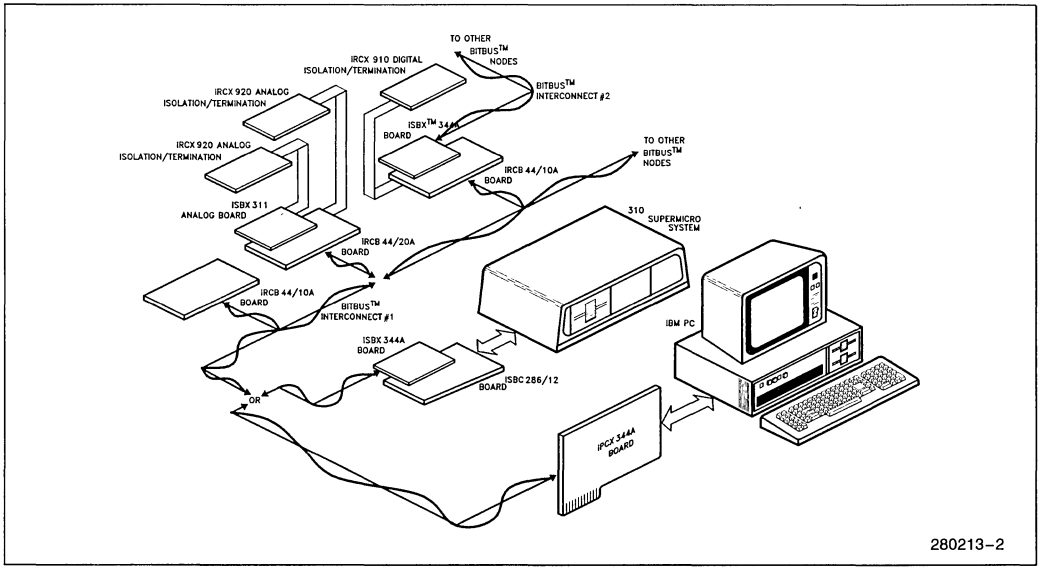


Figure 1. iDCM Operating Environment

Memory

The iRCB 44/10A board memory consists of two sections: internal and external. Internal memory is located in the on-chip memory of the BEM. The iDCX51 Executive and the remaining BEM firmware ration this resource. However, eight bytes of bit addressable internal memory are reserved for the user. Ample space is reserved for user programs and data in the iRCB 44/10A board external memory.

Two 28 pin JEDEC sites comprise the iRCB 44/10A board external memory. One site has been dedicated for data, the other for code. Table 1 lists the supported memory devices for each site. Intel's 2764, and 27128 are examples. The user may choose one of two memory configurations and specify different memory sizes by placing the proper jumpers at system initialization. The most flexible configuration option provides the user with access to the code site for program download or upload. This feature ensures expansion of an existing system is easily accommodated.

Table 1. Supported Memory Devices

Device	Data Site	Code Site
4K × 8-64K × 8 EPROM/ROM	NO	YES
2K × 8-32K × 8 SRAM.	YES	YES
2K × 8-16K × 8 NVRAM and E2PROM	NO	YES

BITBUS™ Microcontroller Interconnect

The iRCB 44/10A board serial interface fully supports the BITBUS microcontroller interconnect. The BITBUS interconnect is a serial bus optimized for

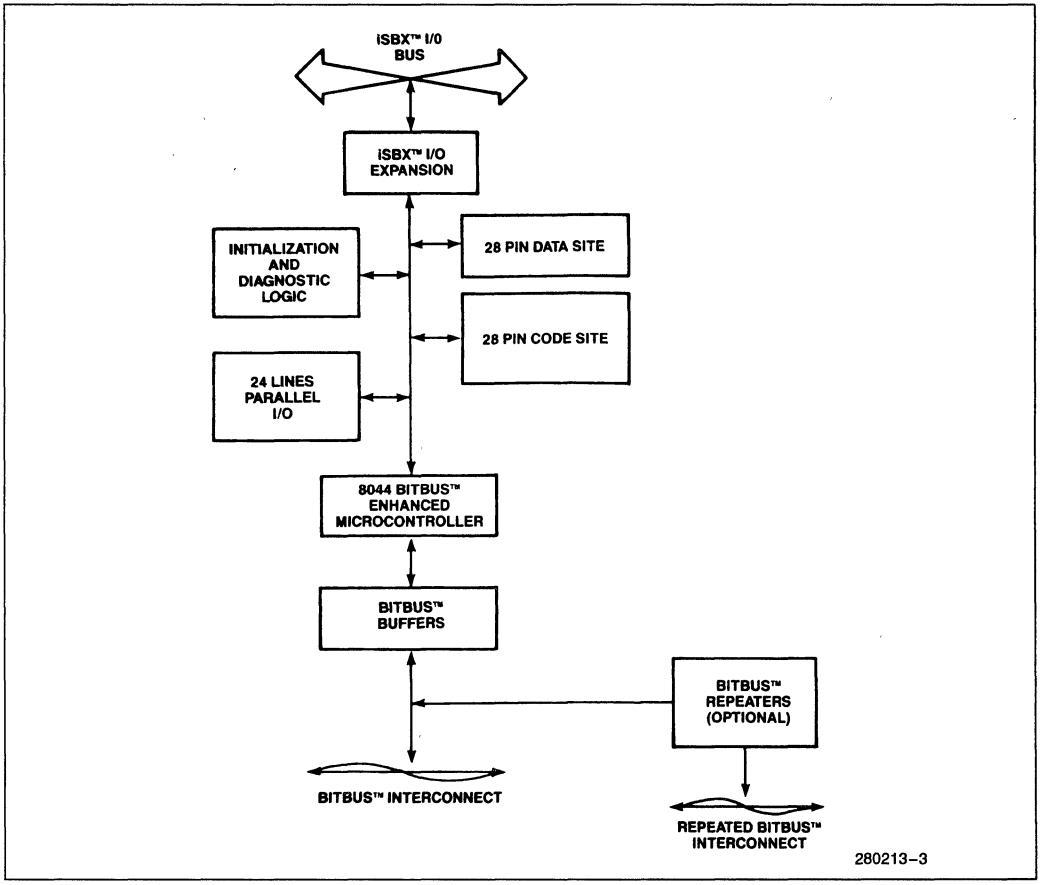


Figure 2. iRCB™ 44/10A Block Diagram

control applications. The bus supports both synchronous and self-clocked modes of operation. These modes of operation are selectable dependent on application requirements as are the transmission speeds. Table 2 shows the different combinations of modes of operation, transmission speeds, and distances. The SDLC-based protocol, BITBUS message format, and compatibility with Intel's other software and hardware products comprise the remainder of the BITBUS architecture. These features contribute to BITBUS system reliability and usefulness as a microcontroller interconnect.

The BITBUS connection consists of one or two differential pair(s) of wires. The serial (BITBUS) interface of the iRCB 44/10A board consists of: a half-duplex RS 485 transceiver, an optional BITBUS repeater and an optional clock source for the synchronous mode of operation.

Digital Parallel I/O

In order to provide an optimal parallel I/O interface for control applications, the iRCB 44/10A board supports 24 software programmable parallel I/O lines. This feature supplies the flexibility and simplicity required for control and data acquisition systems. Sixteen of these lines are fully programmable as inputs or outputs, with loopback, on a bit by bit basis so that bit set, reset, and toggle operations are streamlined. The remaining eight lines are dedicated as inputs. Figure 3 depicts the general I/O port structure.

The parallel I/O lines can be manipulated by using the Remote Access and Control (RAC) function (in BEM firmware) from a supervisory node or locally by a user program. The user program can also access the RAC function or directly operate the I/O lines. Input, output, mixed— input and output, and bit operations are possible simply by reading or writing a particular port.

iSBX™ Expansion

One iSBX I/O expansion connector is provided on the iRCB 44/10A board. This connector can be used to extend the I/O capability of the board. In addition to specialized and custom designed iSBX boards, a full line of compatible high speed, 8-bit expansion MULTIMODULE boards, both single and double wide, are available from Intel. The only incompatible modules are those that require the MWAIT* signal or DMA operation. A few of Intel's iRCB 44/10A board compatible iSBX MULTIMODULE boards include: parallel I/O, serial I/O, BITBUS expansion, IEEE 488 GPIB, analog input and analog output.

With the iSBX 344A BITBUS Controller MULTIMODULE board and user supplied software, the iRCB 44/10A board can act as an intelligent BITBUS repeater facilitating the transition between two BITBUS segments operating at different speeds.

Initialization and Diagnostic Logic

Like the other members of Intel's Distributed Control Modules (iDCM) product line, the iRCB 44/10A board includes many features which make it well suited for industrial control applications. Power up diagnostics is just one of these features. Diagnostics simplify system startup considerably, by immediately indicating an iDCM controller or external bus failure. The LEDs used for power up diagnostics are

Table 2. BITBUS™ Microcontroller Interconnect Modes of Operation

	Speed Kb/s	Maximum Distance Between Repeaters M/ft	Maximum #Nodes Per Segment*	Maximum # Repeaters Between A Master And Any Slave
Synchronous	500–2400	30/100	28	0
Self Clocked	375 62.5	300/1000 1200/4000	28 28	2 10

*Segment: Distance between master and repeater or repeater and repeater. Synchronous mode requires user supplied crystal.

available for user diagnostics after power up as well to further contribute to reliable operation of the system.

Initial iRCB 44/10A board parameters are set by positioning jumpers. The jumpers determine the BITBUS mode of operation: synchronous, self clocked, transmission speed, and address of the iRCB 44/10A board in the BITBUS system. This minimizes the number of spare boards to be stocked for multiple nodes, decreasing stocking inventory and cost.

INTEGRAL FIRMWARE

The iRCB 44/10A board contains resident firmware located in the 8044 BEM. The on-chip firmware consists of: a pre-configured iDCX 51 Executive for user program development; a Remote Access and Controller (RAC) function that enables user communication and control of different microcontrollers and I/O points; a communications gateway to connect the BITBUS interconnect, iSBX bus, iPCX bus and iDCX 51 tasks; and power up diagnostics.

Table 3. iDCX 51 Executive Calls

Call Name	Description
TASK MANAGEMENT CALLS	
RQ$CREATE$TASK	Create and schedule a new task.
RQ$DELETE$TASK	Delete specified task from system.
RQGETFUNCTION$IDS	Obtain the function IDs of tasks currently in the system.
INTERTASK COMMUNICATION CALLS	
RQ$ALLOCATE	Obtain a message buffer from the system buffer pool.
RQ$DEALLOCATE	Return a message buffer to the system buffer pool.
RQ$SEND$MESSAGE	Send a message to specified task.
RQ$WAIT	Wait for a message event.
MEMORY MANAGEMENT CALLS	
RQGETMEM	Get available SMP memory.
RQ$RELEASE$MEM	Release SMP memory.
INTERRUPT MANAGEMENT CALLS	
RQ$DISABLE$INTERRUPT	Temporarily disable an interrupt.
RQ$ENABLE$INTERRUPT	Re-enable an interrupt.
RQ$WAIT	Wait for an interrupt event.
TIMER MANAGEMENT CALLS	
RQSETINTERVAL	Establish a time interval.
RQ$WAIT	Wait for an interval event.

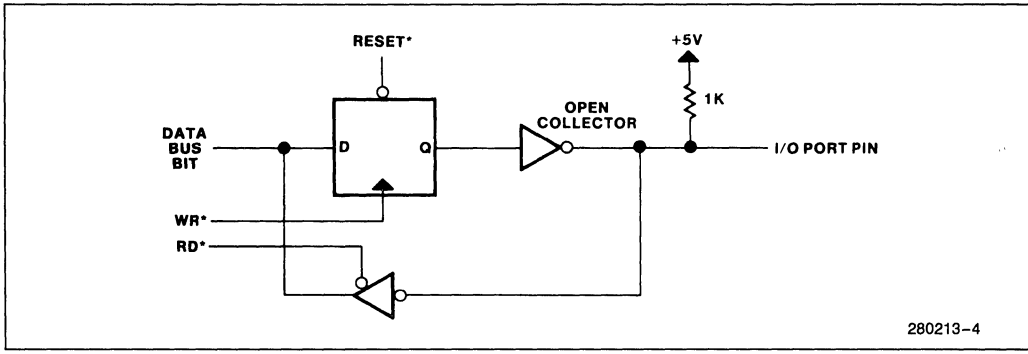


Figure 3. I/O Port Structure

The iDCX 51 Executive is an event-driven software manager that can respond to the needs of multiple tasks. This real-time multitasking executive provides: task management, timing, interrupt handling, and message passing services. Table 3 shows the iDCX 51 calls. Both the Executive and the communications gateway allow for the addition of up to seven user tasks at each node while making BITBUS operation transparent.

The Remote Access and Control Function is a special purpose task that allows the user to transfer commands and program variables to remote BITBUS controllers, obtain the status of a remote I/O line(s), or reverse the state of a remote I/O line. Table 4 provides a complete listing of the RAC services. No user code need be written to use this function. Power up tests provide a quick diagnostic service.

The services provided by the iRCB 44/10A board integral firmware simplify the development and implementation of complex real-time control application systems. All iDCM hardware products contain integral firmware thus supplying the user with a total system solution.

Table 4. RAC Services

RAC Service	Action Taken by Task 0
RESET_STATION	Perform a software reset.
CREATE_TASK	Perform an RQ$CREATE$TASK system call.
DELETE_TASK	Perform an RQ$DELETE$TASK system call.
GET_FUNCTION_ID	Perform an RQGETFUNCTION$IDS call.
RAC_PROTECT	Suspend or resume RAC services.
READ_IO	Return values from specified I/O ports.
WRITE_IO	Write to the specified I/O ports.
UPDATE_IO	Update the specified I/O ports.
UPLOAD_MEMORY	Return the values in specified memory area.
DOWNLOAD_MEMORY	Write values to specified memory area.
OR_I/O	OR values into specified I/O ports.
AND_I/O	AND values into specified I/O ports.
XOR_I/O	XOR values into specified I/O ports.
READ_INTERNAL	Read values at specified internal RAM areas.
WRITE_INTERNAL	Write values at specified internal RAM areas.
NODE_INFO	Return device related information.
OFFLINE	Set node offline.
UPLOAD_CODE	Read values from code memory space.
DOWNLOAD_CODE	Write values to specified EEPROM memory.

INDUSTRIAL PACKAGING

The iRCB 44/10A form factor is a single high, 220 mm deep Eurocard and supports most standard industrial packaging schemes as well as Intel's RCX 910 Digital Signal Conditioning, Isolation and Termination Module (see below). The Eurocard form factor specifies reliable DIN connectors. A standard 64 pin connector is included on the iRCB 44/10A board.

Physical Characteristics

Single high, 220 mm deep Eurocard Form Factor

Dimensions

Width: 13.77 mm (0.542 in) maximum component height

Height: 100 mm (3.93 in.)

Depth: 220 mm (8.65 in.)

Weight: 169 gm (6 ounces)

DIGITAL SIGNAL CONDITIONING, ISOLATION, AND TERMINATION

The RCB 44/10A is fully compatible with the RCX 910 Digital Signal Conditioning, Isolation and Termination Panel. The RCX 910 panel provides integral mounting for one RCB 44/10A, with connectors for power, the BITBUS interconnect signals, and 24 Industry Standard I/O isolation and signal conditioning modules. These modules, available from a number of vendors worldwide, typically provide greater than 1500V isolation and support signal conditioning in a number of voltages including 5–60 VDC, 120 and 240 VAC.

SPECIFICATIONS

Word Size

Instruction: 8 bits

Data: 8 bits

Processor Clock 12 MHz

Instruction Execution Times

1 μsec 60% instructions
2 μsec 40% instructions
4 μsec Multiply & Divide

Memory Capacity/Addressing

iDCM Controller: Up to 64 Kbytes code

DEVELOPMENT ENVIRONMENT

Intel provides a complete development environment for the iRCB 44/10A board.

BITBUS™ Development Environments

	BITBUS™ TOOLS								NODE CODE			ICE™	EPROM PROG.	
	DCS 100 TOOLBOX													
	BBM	UBI	BIH	PC Bridge	OBJHEX	UDI2DOS	DCS 110	DCS 120	ASM 51	PL/M 51	RL 51, LIB 51	ICE 5100/044	iUP200A/201A with iUPF87/44A module and iPPS sw	iPDS with iUPF87/44A module and iPPS sw
Series II									C	C	C		X	
III			X				X	X	X	X	X	X	X	
IV							X	X	X	X	X	X	X	
iPDS	A	A	A				X	X	C	C	C			X
iRMX 5¼"	X	X	X	X	X		X	X	D	D	D			
8"	X	X	X	X	X		X	X	D	D	D			
XENIX 5¼"	X	X		B	X									
8"	X	X		B	X									
DOS	X	X	X	X	X	X	X	X	X	X	X	X	X	

NOTES:
A. iPDS uses Release 1 Toolbox.
B. Supports operation with XENIX. XENIX disks not required.
C. Down-revision version.
D. Available for iRMX® 86.

Address Ranges

Memory	Option A	Option B
External —Data	0000H-7FFFH	0000H-7FFFH
—Code	1000H-0FFFFH	8000H-0FEFFH
Internal	0000H-0FFFH	0000H-0FFFH

NOTES:
Option A: Supports maximum amount of external EPROM code memory.
Option B: Supports downloading code into RAM or EEPROM memory.

Interrupt Sources

Two external: iSBX I/O Expansion bus sources or other sources.
BITBUS Microcontroller Interconnect.

8044 BITBUS™ Enhanced Microcontroller I/O Addressing

Function	Address	Read	Write	Bit
PORT A	FFCOH	✓	✓	
PORT B	FFC1H	✓		
PORT C	FFC2H	✓	✓	
MCSO	FF80H-FF87H FF00, FF01	✓	✓	
MSC1	FF88H-FF8F	✓	✓	
LED #1	90H	✓	✓	✓
LED #2	91H	✓	✓	✓
RDY/NE*	B4H	✓	✓	✓
NODE ADDRESS	FFFFH	✓		
CONFIGURATION	FFFEH	✓		
OPT0	92H	✓	✓	✓
OPT1	93H	✓	✓	✓
INT0	B2H	✓		✓
INT1	B3H	✓		✓

PARALLEL I/O

Number: 2 8-Bit Bi-directional Ports
1 8-Bit Input Port

Table 5. Parallel I/O Electrical Specification

Parameter	Condition	Min	Max	Units
V_{OL}	$I_{OL}=16$ mA		0.5	V
V_{OH}	$I_{OH}=-2$ mA	2.4		V
V_{IH}		2.0	7.0	V
V_{IL}		−1.0	0.8	V
I_{IL}	$V_{IL}=0.5V$		6.0	mA
I_{IH}	V_{IH} = logic high		0.0	mA
I_I	$V_{IH}=7V$		−2.2	mA

Terminations

Sockets provided on board for ¼ Watt 5% Carbon type resistors. Resistor value to match characteristic impedance of cable as closely as possible—120Ω or greater.

Repeaters

Sockets provided on board: Devices 75174 and 75175

Connector Options

10 PIN PLUG

Flat Cable: **3M 3473-6010,** TB Ansley 609-1001M, **or** equal

Discrete Wire: **BERG** 65846-007, ITT Cannon 121-**7326-105, or equal**

DIN CONNECTOR PLUG

Flat Cable: GW Elco 00-8259-096-84-124, Robinson **Nugent** RNE-IDC64C-TG30, or equal

Discrete Wire: ITT Cannon G06 M96 P3 BDBL-004 **GW Elco 60 8257** 3017, or equal

10 Pin Repeater Connector Pin Out

Pin	Signal
1	+12V
2	+12V
3	GND
4	GND
5	DATA*
6	DATA
7	DCLK*/RTS*
8	DCLK/RTS
9	RGND
10	RGND

Electrical Characteristics

Interfaces

iSBX I/O expansion bus: supports the standard I/O Expansion Bus Specification with compliance level D8/8F

Memory Sites: Both code and data sites support the electrical Universal Memory Site specification

BITBUS™ Interconnect: The iRCB 44/10A Remote Controller Board supports the BITBUS Specification as follows:

Fully supported synchronous mode at 2.4 Mbits/second and self clocked mode for 375 kbits/second and 62.5 kbits/second

The iRCB 44/10A Remote Controller Board presents one standard load to the BITBUS without repeaters, with repeaters two standard loads

Message length up to 54 bytes supported

RAC Function support as shown in Table 4

Parallel I/O: See the Table 5 for Electrical Specifications of the interface.

Power Requirements

0.9A at +5V ±5% iRCB 44/10 board only (power to memory, repeater, or iSBX board NOT included)

Environmental Characteristics

Operating Temperature: 0°C to 55°C at 200 Linear Feet/Minute Air Velocity

Humidity: 90% non-condensing

Reference Manual (NOT Supplied)

iRCB 44/10 Digital I/O Remote 148100-001 Controller Board User's Guide

Ordering Information

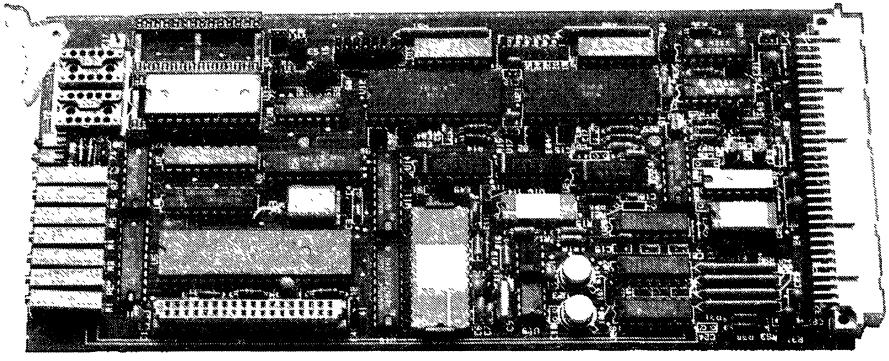
Part Number	Description
iRCB 44/10A	BITBUS Digital I/O Remote Controller Board

iRCB 44/20A
ANALOG I/O CONTROLLER

- Distributed Intelligence via BITUS™ Serial Bus
- 8044 8-bit Microcontroller at 12 MHz
- 12-bit Analog Resolution
- Up To 20 KHz Aquisition Rate (50 ms)
- Software Programmable Gain: 1, 10, 100, 500
- Two 28-pin JEDEC Memory Sites
- 16 Single-ended or 8 Differential Input Channels
- 2 Outputs Channels
- ±10V Range or 4–20 mA Current Loop
- I/O Expandable via iSBX™ Connector
- Compact Single-Eurocard Packaging
- Low Power Consumption
- Compatible with iRCX 920 Analog Signal Conditioning, Isolation and Termination Panel

The iRCB 44/20A is a fully programmable analog I/O subsystem on a single-Eurocard form-factor board. The resident 8044 microcontroller operating at 12 MHz provides a means of executing data aquisition and control routines remote from the host computer. Real-time capability is made possible by the iDCX 51 Distributed Control Executive, resident in the 8044 microcontroller. Distribution of real-time control is implemented by the BITBUS Serial Bus protocol, which is also managed integrally by the 8044.

Offering high performance, low-cost, and improved system bandwidth via distributed intelligence, the iRCB 44/20A Analog I/O Controller is ideal for data acquisition and control in both laboratory and industrial environments.



280721-1

MULTIMODULE™ is a trademark of Intel Corporation.

IBM® PC is a registered trademark of International Business Machines Corporation.

November 1988
Order Number: 280721-002

APPLICATION ENVIRONMENT

Intel's Distributed Control Modules (iDCM) product family contains the building blocks to implement real-time distributed control applications. The iDCM family incorporates the BITBUS interconnect to provide standard high-speed serial communication between microcontrollers. The iRCB 44/20A may communicate with other nodes in a distributed system via the BITBUS interconnect as shown in Figure 1. Other nodes in the system may be the iSBX 344A BITBUS Controller MULTIMODULE™, the iPCX 344A BITBUS IBM® PC Interface, the iRCB 44/10A BITBUS Digital I/O Controller Board, or other BITBUS compatible products.

The iRCB 44/20A board, can be used as an intelligent remote controller or an I/O expansion device. When performing as an intelligent controller the iRCB 44/20A board not only monitors the status of multiple sensors, it can also locally execute user developed control algorithms. When functioning as an I/O expansion device the iRCB 44/20A board manages the multiple I/O ports, transmitting this information via the BITBUS bus or iSBX interface to the system controller for analysis or data logging purposes.

Typically, the iRCB 44/20A board will operate as a node in a BITBUS system. BITBUS communication supports a multidrop configuration with one master, and multiple subordinate nodes. The iRCB 44/20A board may be either a master or slave node to manage a wide variety of analog input or output tasks.

FUNCTIONAL DESCRIPTION

The major functional blocks of the iRCB 44/20A board, shown in Figure 2, include the 8044 microcontroller and BITBUS interconnect, local memory, Analog I/O, and iSBX expansion.

Distributed Intelligence

The heart of the iRCB 44/20A board's controlling and communication capability is the highly integrated 8044 microcontroller which operates at 12 MHz. The 8044 contains the advanced 8-bit, 8051 microcontroller and a complimentary SDLC controller, called the Serial Interface Unit (SIU). This dual processor architecture provides complex control and high speed communication functions at a low cost.

Another essential part of the 8044 controller is the on-chip firmware that exercises the BITBUS interface. The 8044's SIU acts as an SDLC controller, off loading the on-chip microcontroller of communication tasks so it may concentrate on real-time control.

The 8044 microcontroller simplifies the user interface, and offers high performance communications and control capabilities in a single component package. Many interconnected Distributed Control Modules can form a powerful platform to efficiently and economically administer a complete control system.

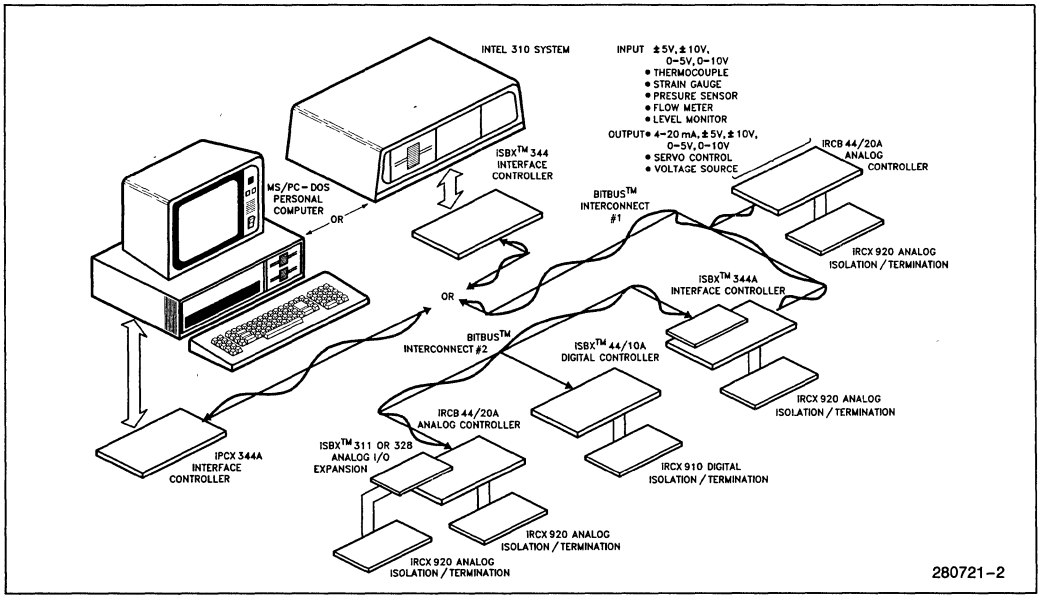


Figure 1. BITBUS Distributed Control Example

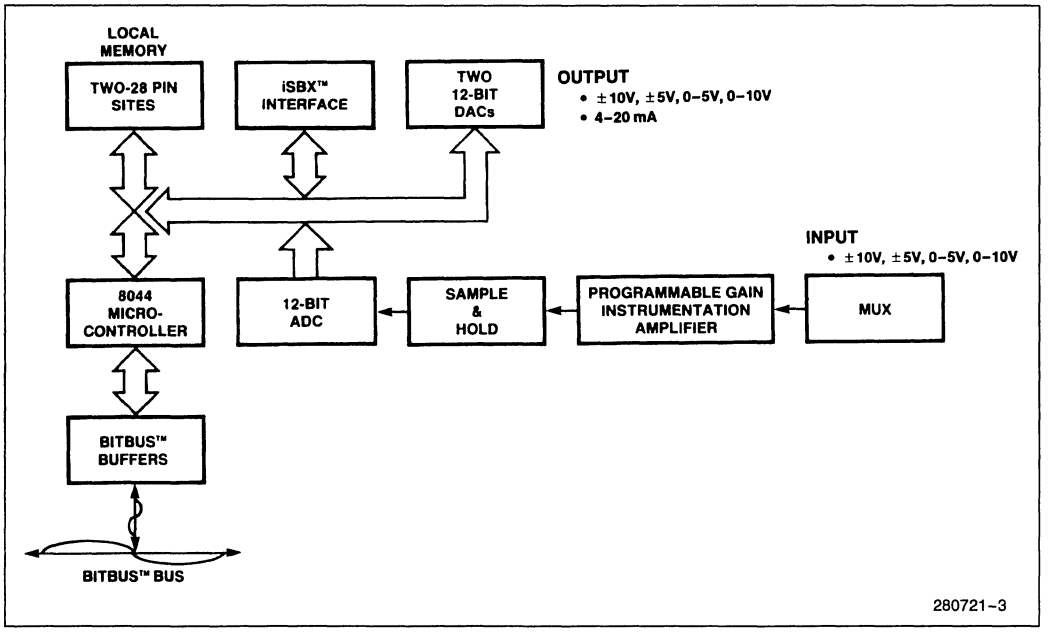


Figure 2. iRCB 44/20A Functional Diagram

BITBUS™ Microcontroller Interconnect

The iRCB 44/20A board fully supports the BITBUS microcontroller interconnect. BITBUS is a serial bus optimized for control applications. Both synchronous and self-clocked modes of operation are supported as well as multiple transmission rates. Table 1 shows the different combinations of modes of operation, transmission speeds, and distances. The SDLC protocol and BITBUS message format comprise the data-link level of the BITBUS architecture. Use of these standards maximizes system reliability and flexibility.

The physical connection to BITBUS uses either one or two pairs of wires across which differential signals travel. The iRCB 44/20A board contains a half-duplex RS 485 tranceiver and an optional clock source for the synchronous mode of operation.

Local Memory

The iRCB 44/20A board contains both internal and external local memory. Internal memory is located within the 8044 controller and is used by the iDCX 51 Executive and the SIU. Eight bytes of bit-addressable internal memory have been reserved for the user.

Two 28-pin JEDEC sites provide the iRCB 44/20A board with memory that is external to the 8044. One site has been dedicated for data, the other for application code. Table 2 lists the supported memory devices for each site. The user may select one of two memory configurations using jumpers. One option provides the user with access to the application code site for uploading or downloading programs, which allows expansion or modification of an existing system from a remote site.

Table 1. BITBUS™ Microcontroller Interconnect Modes Of Operation

	Speed Kb/S	Maximum Distance Between Repeaters M/ft	Maximum # Nodes Per Segment*	Maximum # Repeaters Between A Master And Any Slave
Synchronous	500–2400	30/100	28	0
Self Clocked	375 62.5	300/1000 1200/4000	28 28	2 10

* Segment: Distance between master and repeater or repeater and repeater. Synchronous Mode requires user supplied crystal.

Table 2. Supported Memory Devices

Device	Data Site	Code Site
4K x 8–64K x 8 EPROM/ROM	NO	YES
2K x 8–32K x 8 SRAM	YES	YES
2K x 8–16K x 8 NVRAM and E2PROM	NO	YES

Analog I/O

The iRCB 44/20A has been designed to manage a wide variety of analog functions. The jumper-selectable voltage or current ranges plus software programmable gain allows the iRCB 44/20A to acquire data from a combination of up to 16 thermocouples, strain gauges, pressure transducers, flow meters, level sensors, or any devices that operate on a 4–20 mA current loop. Two analog output channels provide the capability to adjust system parameters locally through servo control, voltage-driven devices, or other actuators that respond to 4–20 mA signals.

The 8044 microcontroller on the iRCB 44/20A allows Proportional Integral/Derivative (PID) algorithms, event timing, or averaging tasks to operate independent of the host computer or programmable controller. By off-loading the host in this manner, the overall system performance can be improved significantly.

The analog I/O lines can be manipulated from a remote supervisor by communicating with the Remote Access and Control (RAC) functions, which are included in the 8044 controller firmware. The local application program running on the iRCB 44/20A can also access the RAC functions or directly operate the I/O lines.

iSBX™ Expansion

One 8-bit iSBX I/O expansion connector is provided to expand the functionality of the iRCB 44/20A board. A full line of compatible expansion MULTIMODULE boards are available from Intel; both single- and double-wide versions are supported by the iRCB 44/20A. Parallel I/O, serial I/O, IEEE 488, magnetic-bubble memory, or additional analog I/O may be added in this manner.

Also, the iSBX 344A BITBUS Controller MULTIMODULE can be used to implement another BITBUS hierarchy with the iRCB 44/20A functioning as the master. With user supplied software, this product combination can operate as an intelligent BITBUS repeater, facilitating the transmission between two BITBUS segments operating at different speeds.

Initialization and Diagnostic Logic

Like the other members of the Intel's Distributed Control Modules (iDCM) product line, the iRCB 44/20A board includes many features which make it well suited for industrial control applications. Power-up diagnostics simplify system initialization by immediately indicating a failure in either the 8044 microcontroller or external bus. On-board LEDs indicate diagnostic status and are available after power-up for user developed diagnostic routines.

Initial iRCB 44/20A board parameters are manually set with jumpers. These jumpers specify the mode of operation (synchronous or self clocked), and transmission speed. The address of the iRCB 44/20A board within the BITBUS system is also declared in this manner. Therefore, spare board inventory is reduced, since the iRCB 44/20A may be positioned at any node address.

INTEGRAL iDCX 51 FIRMWARE

The iRCB 44/20A board contains resident firmware located within the 8044 controller. This on-chip firmware, known as DCM 44, consists of a pre-configured iDCX 51 Distributed Control Executive for user program development and execution, a library of Remote Access and Control (RAC) functions for internode communications and I/O control, plus an iSBX communications gateway, and power-up diagnostics.

The iDCX 51 Executive is an event-driven software manager that can respond to the needs of multiple tasks. This real-time multitasking executive provides task management and timing, interrupt handling, and message passing services. Table 3 shows the iDCX 51 user command library. Both the executive and the communications gateway allow for the addition of seven user tasks at each node that are independent of BITBUS bus management operations.

Remote Access and Control (RAC) functions are special purpose tasks that allow the host system to transfer commands and program variables to remote BITBUS controllers and read/write to the remote I/O lines. Table 4 provides a complete listing of the RAC commands. No user code need be written to use this function. Power-up tests provide a quick diagnostic service.

The DCM 44 firmware, integral to the iRCB 44/20A board, simplifies the development and implementation of complex real-time control applications. All iDCM hardware products contain this integral firmware, providing the user with application code portability.

Table 3. iDCX 51 Executive Calls

Call Name	Description
TASK MANAGEMENT CALLS	
RQ$CREATE$TASK	Create and schedule a new task.
RQ$DELETE$TASK	Delete specified task from system.
RQGETFUNCTION$IDS	Obtain the function IDs of tasks currently in the system.
INTERTASK COMMUNICATION CALLS	
RQ$ALLOCATE	Obtain a message buffer from the system buffer pool.
RQ$DEALLOCATE	Return a message buffer to the system buffer pool.
RQ$SEND$MESSAGE	Send a message to specified task.
RQ$WAIT	Wait for a message event.
MEMORY MANAGEMENT CALLS	
RQGETMEM	Get available SMP memory.
RQ$RELEASE$MEM	Release SMP memory.
INTERRUPT MANAGEMENT CALLS	
RQ$DISABLE$INTERRUPT	Temporarily disable an interrupt.
RQ$ENABLE$INTERRUPT	Re-enable an interrupt.
RQ$WAIT	Wait for an interrupt event.
TIMER MANAGEMENT CALLS	
RQSETINTERVAL	Establish a time interval.
RQ$WAIT	Wait for an interval event.

Table 4. RAC Services

RAC Service	Action Taken by Task 0
RESET_STATION	Perform a software reset.
CREATE_TASK	Perform an RQ$CREATE$TASK system call.
DELETE_TASK	Perform an RQ$DELETE$TASK system call.
GET_FUNCTION_ID	Perform an RQGETFUNCTION$IDS call.
RAC_PROTECT	Suspend or resume RAC services.
READ_IO	Return values from specified I/O ports.
WRITE_IO	Write to the specified I/O ports.
UPDATE_IO	Update the specified I/O ports.
UPLOAD_MEMORY	Return the values in specified memory area.
DOWNLOAD_MEMORY	Write values to specified memory area.
OR_I/O	OR values into specified I/O ports.
AND_I/O	AND values into specified I/O ports.
XOR_I/O	XOR values into specified I/O ports.
READ_INTERNAL	Read values at specified internal RAM areas.
WRITE_INTERNAL	Write values to specified internal RAM areas.
NODE_INFO	Return device related information.
OFFLINE	Set node offline.
UPLOAD_CODE	Read values from code memory space.
DOWNLOAD_CODE	Write values to specified EEPROM memory.

NOTES:
Internal memory locations are included in the 192 bytes of data RAM provided in the microcontroller. External memory refers memory outside the microcontroller—the 28-pin sockets of iRCB 44/20A board. Each RAC Access Function may refer to 1, 2, 3, 4, 5 or 6 individual I/O or memory locations in a single command.

INDUSTRIAL PACKAGING

The iRCB 44/20A board conforms to a single-wide (3V), 220 mm deep Eurocard form-factor. This allows the iRCB 44/20A to fit within standard industrial racks or chassis as well as Intel's RCX 920 Analog Signal Conditioning, Isolation & Termination Panel (see below). The Eurocard specification references DIN 41612 connectors, which are used on the iRCB 44/20A board.

ANALOG SIGNAL CONDITIONING, ISOLATION AND TERMINATION

The RCB 44/20A is fully compatible with the RCX 920 Analog Signal Conditioning, Isolation and Termination Panel. The RCX 920 panel provides integral mounting for one RCB 44/20A, with connectors for power, the BITBUS interconnect signals, and 18 Analog Devices 5B Series Signal Conditioning and Isolation Modules. These modules provide 240V RMS field wiring protection, and 1500V RMS common mode voltage isolation and support signal conditioning in a wide range of analog voltages and currents including thermocouple and RTD sensors, millivolt and volt inputs and 4–20 mA and 0–20 mA outputs.

SPECIFICATIONS

CPU

8044 BITBUS Enhanced Microcontroller (BEM)

Word Size

Instruction—8 bits
Data—8 bits

Processor Clock

12 MHz

Instruction Execution Times

1 μsec 60% instructions
2 μsec 40% instructions
4 μsec Multiply & Divide

Memory Addressing

iDCM Controller
Up to 64K bytes code

Address Ranges

	Option A	Option B
External Data Memory Site	0000H–7FFFH	0000H–7FFFH
External Code Memory Site	1000H–0FFFFH (0000H–0FFFFH if EA Active)	8000H–0FFEFH
Internal Code Memory	0000H–0FFFH	0000H–0FFFH

NOTES:
Option A: Supports maximum amount of external EPROM code memory
Option B: Supports downloading code into external RAM or EEPROM memory

BITBUS™ Development Environments

HOST SYSTEM	BITBUS™ TOOLS								NODE CODE			ICE™	EPROM PROG.	
	DCS 100 TOOLBOX													
	BBM	UBI	BIH	PC Bridge	OBJHEX	UDI2DOS	DCS 110	DCS 120	ASM 51	PL/M 51	RL 51, LIB 51	ICE 5100/044	iUP200A/201A with iUPF87/44A module and iPPS sw	iPDS with iUPF87/44A module and iPPS sw
Series II									C	C	C		X	
III			X				X	X	X	X	X	X	X	
IV							X	X	X	X	X	X	X	
iPDS™	A	A	A				X	X	C	C	C			X
iRMX® 5¼″	X	X	X	X	X		X	X	D	D	D			
8″	X	X	X	X	X		X	X	D	D	D			
XENIX 5¼″	X	X		B	X									
8″	X	X		B	X									
DOS	X	X	X	X	X	X	X	X	X	X	X	X	X	

NOTES:
A. iPDS™ uses Release 1 Toolbox.
B. Supports operation with XENIX. XENIX disks not required.
C. Down-revision version.
D. Available for iRMX® 86.

I/O Capability

Analog—16 single-ended or 8 differential channels and 2 outputs channels

Expansion—one single-or double-wide iSBX MULTIMODULE (MWAIT * or DMA not supported by iRCB 44/20)

Interrupt Sources

Two external: iSBX I/O Bus or BITBUS Interconnect sources

Bus Termination

Jumper selectable resistors provide termination capability for cable with an impedance of 120Ω or greater.

Analog Input Specifications

Number of channels—16 single-ended or 8 differential
Input ranges—0 to 5V, 0 to 10V (unipolar) ±5V, ±10V (bipolar)
Gain ranges—1, 10, 100, 500, (software programmable)
Input impedance—100MΩ
Input bias current—±50 nA
Overvoltage protection—±32V power on
±20V power off

Accuracy

Resolution—12 bits
Linearity and Noise—±¾ LSB (trimmable)
System Accuracy
Gain = 1—±0.035% full-scale range (trimmable)
Gain = 500—±0.15% full-scale range (trimmable)

Stability

Gain tempco—32 ppm/°C (gain = 11)
75 ppm/°C (gain = 500)
Offset tempco—100 microvolts/°C max.

Dynamic Performance

Aggregate throughout—20 KHz (gain = 1, 10)
7.5 KHz (gain = 100, 500)
Common mode rejection—70 dB (gain = 1)
100 dB (gain = 500)
A/D conversion time—30 microseconds

Analog Output Specifications

Number of channels—2
Output ranges—0 to 5V, 0 to 10V (unipolar) ±5V, ±10V (bipolar)
Current-loop range—4 to 20 mA (unipolar mode only)
Output impedance—0.2Ω min. (voltage)
5 MΩ max. (current)
Output current—±5 mA (short-circuit protected)

Accuracy

Resolution—12 bits
Linearity and Noise—±¾ SB (trimmable)
System Accuracy—
Gain = 1—−0.35% full-scale range (trimmable)
Gain = 500—±0.15% full-scale range (trimmable)

Stability

Full-scale temperature coefficient
150 microvolts/°C (unipolar)
300 microvolts/°C (bipolar)
0.6 microamps/°C (current-loop)
Offset temperature coefficient
30 microvolts/°C (unipolar)
180 microvolts/°C (bipolar)
0.3 microamps/°C (current-loop)

Mating Connectors

Function	# of Pins	Type	Vendor	Part Number
BITBUS Connector	64	Flat Cable	GW Elco Robinson Nugent	00-8259-096-84-124 RNE-IDC-64C-TG30
iSBX Connector	36	Wire Wrap	ITT Cannon GW Elco	G06 M96 P3 BDBL-004 60 8257 3017
		Solder	Viking	000292-0001

Dynamic Performance

Aggregate throughput—20 KHz (gain = 1, 10)
7.5 KHz (gain = 100, 500)
Settling Time—15 microseconds to ±½ LSB

Electrical Characteristics

Interface Compliance

iSBX BUS (through level D8/8F):
Memory sites—code and data sites are JEDEC compatible

BITBUS:

— Synchronous and self-clocked mode support for 500 Kbps to 2.4 Mbps, 375K and 62.5K bits/sec

NOTE:
On-board ALE clock supports 1 Mbps synchronous operation. All other synchronous mode speeds require user-supplied 2.0–9.6 MHz crystal.

— Equivalent to 1.1 standard (RS 485) loads

— Message length up to 54 bytes maximum

Power Requirement (exclusive of optional memory or iSBX MULTIMODULE)

Voltage	Current (amps)	Max, Power (watts)
+5V ±5% +12V ±5% −12V ±5%	0.9 max. 0.7 typ 100 mA max. 100 mA max.	4.5

NOTE:
+15V and −15V required for 0 to 10V and ±10V ranges; for ±15V operation, the iRCB 44/20A cannot be used with iSBX MULTIMODULES that use ±12V power sources.

Physical Characteristics

Width: 3.77 mm (0.542 in) maximum component height
Height: 100 mm (3.93 in)
Depth: 220 mm (8.65 in)
Weight: 169 gm (6 ounces)

Environmental Characteristics

Operating Temperature: 0°C to +60°C at 0.8 CFM air volume
Relative Humidity: 90% non-condensing

Reference Manual (Not Supplied)

148816— iRCB 44/20A Hardware Reference Manual

ORDERING INFORMATION

Part Number	Description
iRCB 44/20A	BITBUS Analog I/O Controller Board

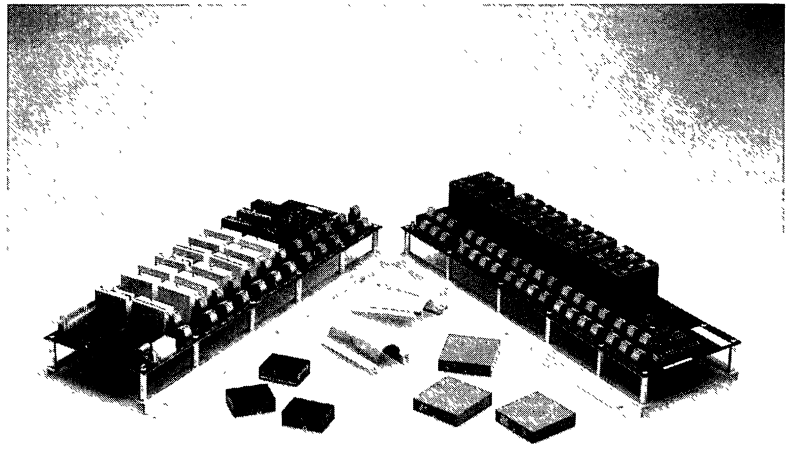
iRCX 910/920 DIGITAL/ANALOG SIGNAL CONDITIONING ISOLATION AND TERMINATION PANELS

iRCX-910

- Digital termination for BITBUS™ iRCB 44/10A digital remote controller board, iSBX™ 350 digital MULTIMODULE™ and Multibus® digital I/O single board computers (SBCs)
- Sockets for 24 industry-standard, optically coupled isolation and signal conditioning I/O modules
- LEDs indicate status of each module
- Separate connectors for BITBUS, Power, RCB and Expansion I/O
- Integral mounting site for one 24 channel digital iRCB 44/10A

iRCX 920

- Analog termination for iRCB 44/20A, iSBX 311, iSBX 328, and iSBC 88/40
- Sockets accepting up to 18 Analog Devices Corporation's 5B Series of isolation and signal conditioning modules
- Separate connectors for BITBUS, Power, RCB and Expansion I/O
- Integral mounting site for one 18-channel analog iRCB 44/20A



AUGUST 1988
ORDER NUMBER: 280443-001

More Convenient BITBUS™ System Integration

Intel now provides one more building block for developing BITBUS™ networks: the iRCX 910/920 Digital/Analog Signal Isolation and Termination Panels. These boards provide remote node termination and isolation in a design that's easy to install and service. They work with Intel's RCB 44/10A and 44/20A, which provide analog/digital control, and with Intel's BITBUS Monitor, DCX-51 Real-time Multitasking Executive and BITBUS Toolbox, which provide the software support.

Intel makes BITBUS system integration easier and more convenient than ever.

Compatible With a Wide Range of Intel MULTIBUS® Boards

The iRCX 910 and iRCX 920 not only work with the iRCB 44/10A and 44/20A controller boards but also with a wide range of MULTIBUS® boards both from Intel and MULTIBUS Manufacturing Group vendors. The 50-pin expansion connection on the iRCX panels makes iSBC and iSBX board connection easy.

Table 1 shows the Intel iSBC and iSBX boards currently compatible with the iRCX products.

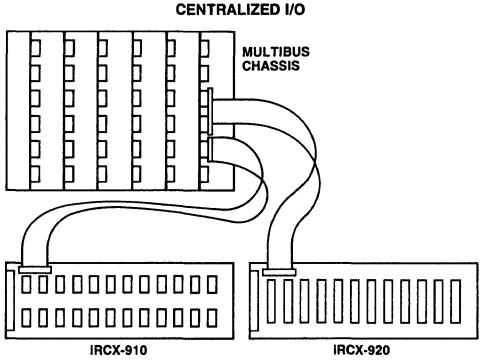
iRCX 910: Compatible With Industry-Standard I/O Modules

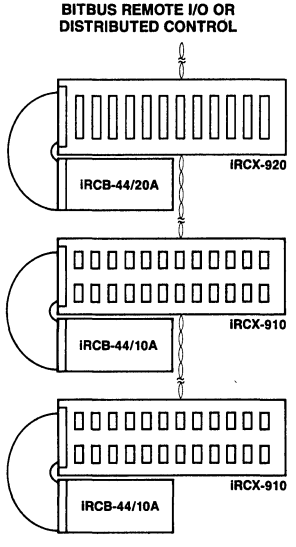
The iRCX 910 provides sockets for 24 channels of user-provided digital I/O to perform AC or DC switching and isolation. The user can configure up to 24 I/O channels, filling only the channels needed for the application. The iRCX 910 accepts a wide range of industry-standard I/O modules, including those from Gordos, Opto 22, Crydom, Potter-Brumfield and others[1]. The input modules convert high-level inputs from such sources as limit or proximity switches to TTL levels. The output modules convert TTL to high-level signals for driving motor starters, solenoids, indicating lights, and the like. Regardless of whether input or output, these modules typically provide greater than 1500 volt isolation, 2-3 KV of transient noise protection, and signal conditioning in a wide range of voltages. An LED for each channel shows on/off status, and a 5 amp fuse provides overcurrent protection.

[1]Mention of these companies in no way constitutes an endorsement by Intel of their products

Table 1: Intel Boards Compatible with iRCX 910 and iRCX 920.

iRCX 910		iRCX 920	
Intel iSBCs	**iSBX**	**Intel iSBC**	**Intel iSBX**
80/10B 80/20-4 80/24A 80/30 80/05A 86/14 86/30 86/35 88/25 88/40A 517 519	350	88/40A	311 328





Examples of how iRCX 910 and 920s can be used in MULTIBUS and BITBUS systems.

iRCX 920: Uses Analog Devices Corporation Modules to Provide State-of-the-Art Signal Conditioning and Isolation

The iRCX 920 terminates 18 analog signals going to and from field wiring and provides signal conditioning and isolation using Analog Devices Corporation's 5B Series of analog isolation and signal conditioning modules (purchased separately). These modules provide 240 volt RMS field wiring protection, 1500 volt RMS common mode voltage isolation and signal conditioning in a wide range of analog voltage and currents, including thermocouple and RTD sensors, millivolt and volt inputs, and 0-20 ma and 4-20 ma process current outputs. Possible connections include temperature and pressure sensors, frequency counters and many others. The iRCB 44/20A when used in conjunction with the iRCX 920 provides up to 16 analog inputs and 2 analog outputs.

The iRCX 920 also contains an integral temperature sensor isothermal barrier strip (RTD) to provide a temperature reference for thermocouple modules doing cold junction compensation. Because this compensation is implemented in hardware rather than software, it simplifies the controller software's task, allowing superior software performance.

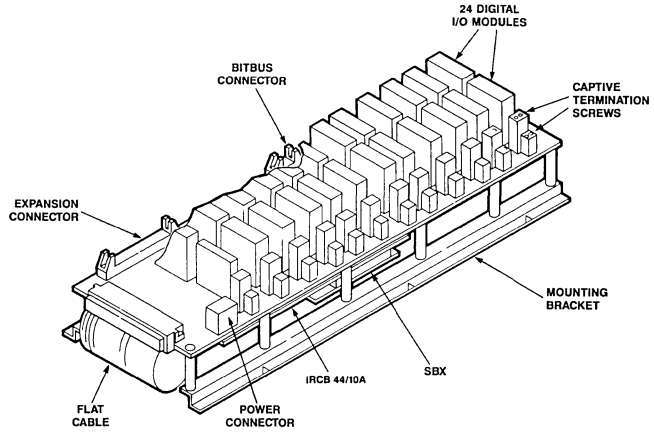
Easy to Install, Easy to Use

The iRCX panels provide a quick and easy, plug-in solution to remote node BITBUS interconnection. They can be mounted to an industrial panel or in a standard RETMA 19″ rack when used with a customer provided 19″L × 7″W pan. Quick access to the RCB 44/10A and 44/20A boards is accomplished by loosening six screws and shifting the iRCX slightly. Field wiring connections are made using captive screw terminals, positioned to allow easy wire routing. Installation is quick, service is easy.

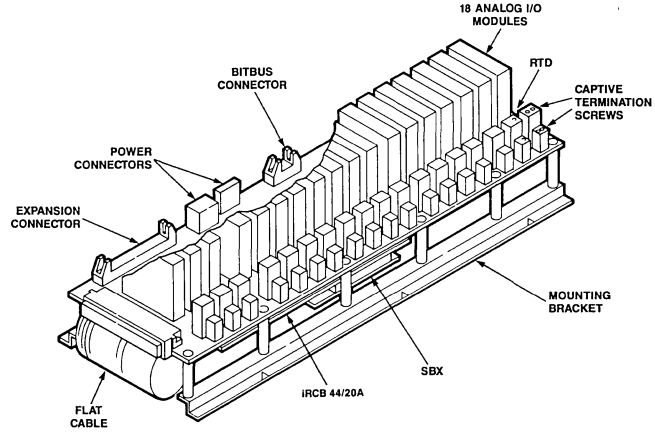
The I/O modules on the iRCX 910 are color-coded for easy identification. LEDs provide on/off status, allowing the operator a quick verification of I/O operation, and are also useful for start-up testing, debugging and troubleshooting a process or machine breakdown.

Increased Reliability

The iRCX 910 and 920 feature improved noise immunity through judicious component placement and the inclusion of a ground terminal. Also, they're mounted in front of the iRCB 44/10A and 44/20A boards, protecting the heart of the remote node from accidental damage.



iRCX 910: The Complete Remote Node Solution for Distributed I/O and Control.



iRCX 920: The Complete Remote Node Solution for Distributed I/O and Control.

Specifications

POWER REQUIREMENTS (TYPICAL)

RCX 910

V_{CC} = +5 VDC ±5%

I_{CC} = 0.03 a/module installed
+1.00 a (if RCB 44/10A installed)
+ current requirements of any installed SBX

RCX 920

V_{CC} = + 5 VDC ±5%

I_{CC} = 0.03 a/input module
+0.17 a/output module
+1.00 a (if RCB 44/20A is installed)
+ current requirements of any installed SBX

Additional Power Requirements when used with RCB 44/20A
100 ma @ +12 VDC ±4%
100 ma @ −12 VDC ±4%

DIMENSIONS

RCX 910
Width: 17.00″ (43.18 cm)
Height: 6.20″ (15.75 cm)
Depth: 3.25″ (8.26 cm) with user-provided modules installed

RCX 920
Width: 17.00″ (43.18 cm)
Height: 6.20″ (15.75 cm)
Depth: 4.25″ (10.80 cm) with user-provided modules installed

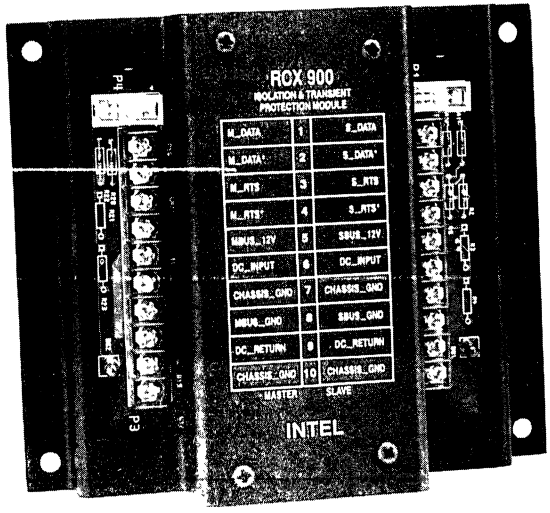
ENVIRONMENTAL REQUIREMENTS

RCX 910/RCX 920 Standalone
Operating Temperature: 0° to 70° C (32° to 158° F)
Operating Humidity: 0–90% R.H. (non-condensing)

RCX 910/RCX 920 with mounted RCB
In still air:
0° to 55° C (32° to 131° F)
With 200 linear feet/minute forced air:
0° to 60° C (32° to 140° F)

Ordering Information

Part Name	Description
iRCX 910	BITBUS Digital Signal Conditioning, Isolation, and Termination Panel
iRCX 920	BITBUS Analog Signal Conditioning, Isolation, and Termination Panel
Analog I/O Modules	Analog Devices 5B Series To order contact: Analog Devices Corporation One Technology Way Norwood, MA 02062 (617) 329-4700



ISOLATION AND TRANSIENT PROTECTION FOR BITBUS™ NETWORKS

The Intel iRCX 900 Isolation Module protects against ground loops, voltage differences between nodes, and transient voltage spikes in BITBUS™ networks operating in the 62.5 to 375 Kbps asynchronous (self-clocked) modes. The iRCX 900 also serves as an isolated, standalone repeater, driving up to 28 remote BITBUS nodes.

FEATURES:

- Common-mode isolation of 1500 VAC for a working voltage of 480 VAC.
- Transient protection meeting the IEEE 472/ANSI-C37.90a-1974 Surge Withstand Compatibility Test.
- On-board DC-DC switching voltage regulator with 1500 VAC isolation.
- Power-on indicator LEDs.
- Rugged industrial packaging with multiple mounting options
- Dual termination with 10-pin BITBUS connector and 10-position terminal block.

October, 1989
Order Number 280639-001

FEATURES

STANDALONE ISOLATION AND TRANSIENT PROTECTION

The iRCX 900 provides BITBUS network protection against voltage surges and transients in industrial environments. This protection is provided through the use of optocouplers, transient suppressors, and an isolated DC-DC voltage regulator.

DOUBLES AS A NETWORK REPEATER

The iRCX 900 also serves as a network repeater, with each iRCX 900 module capable of driving up to 28 remote BITBUS nodes. The iRCX 900 can be powered either from the 12V BITBUS signal line or a separate 12-24V power source.

EASY TO INSTALL AND UPDATE

Installation and change of configuration are easy, because the iRCX 900 is a standalone module. It can be placed anywhere in the network without piggybacking it to another board. The iRCX 900 can be panel-mounted or placed on a desk or lab benchtop. Wiring instructions are clearly outlined on the module packaging.

SERVICE, SUPPORT AND TRAINING

Intel provides worldwide support for board repair or on-site service.

INTEL QUALITY AND RELIABILITY

The iRCX 900 is designed and manufactured in accordance with Intel's high quality standards. We then verify quality through rigorous testing in our state-of-the-art Environmental Test Laboratory.

SPECIFICATIONS

POWER REQUIREMENTS

Input Voltage: 10.8–26.4 VDC
Input Power: 2.0 watts (typical)

May be powered from a remote BITBUS node or from a user-provided power supply.

DIMENSIONS:

Width: 6.48 in. (16.46 cm)
Height: 5.7 in. (14.48 cm)
Depth: 1.78 in. (4.52 cm)

ENVIRONMENTAL REQUIREMENTS

Storage Temperature: – 40° to 70°C (– 40° to 158°F)
Operating Temperature: 0° to 55°C (32° to 131°F)
Storage Humidity: 5%–95% non-condensing at 55°C
Operating Humidity: 8%–90% non-condensing at 55°C

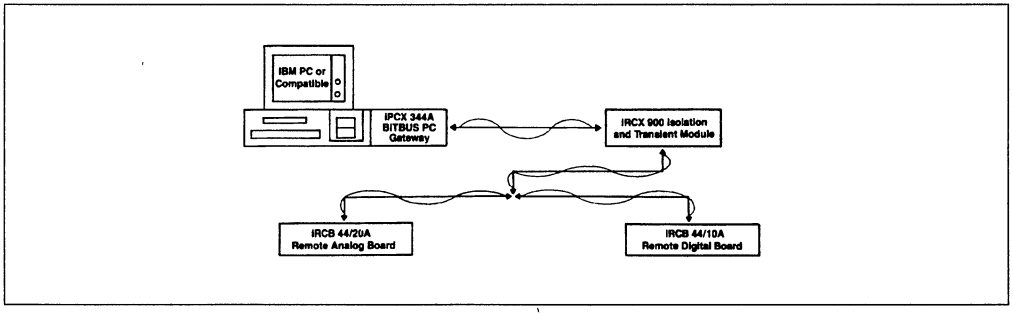


Figure 1: Example Configuration

ORDERING INFORMATION

Order Code:

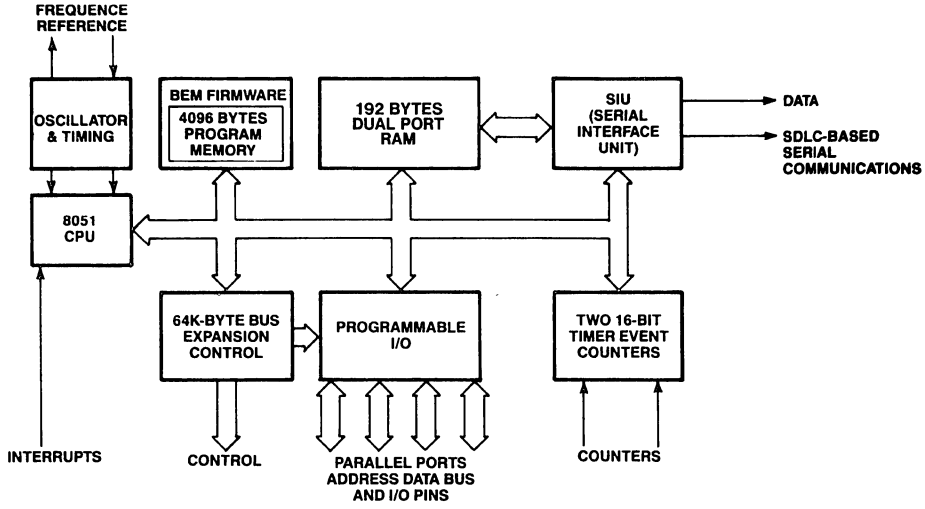
iRCX 900 BITBUS Isolation and Transient Protection Module

For more information or the number of your nearest sales office call 800-548-4725 (good in the U.S. and Canada).

8044 BITBUS™ ENHANCED MICROCONTROLLER

- **Dual Processor Microcontroller Architecture**
- **High Performance 8-Bit CPU**
- **Embedded Parallel Communications Firmware**
- **Tuned for Distributed Real-Time Control**
- **BITBUS™ Firmware Included On-Chip**
- **Power-Up Diagnostics**
- **DCX 51 Distributed Control Executive Included On-Chip**
- **MCS®-51 Software Compatible**

The 8044 BITBUS Enhanced Microcontroller (BEM) is a powerful 8-bit microcontroller with on-chip firmware. The dual processor architecture of the 8044 combined with the inherent the processing power of an 8051 CPU is well suited for distributed data acquisition and control applications in both the factory and laboratory. The firmware integral includes facilities for: diagnostics, task management, message passing, and user-transparent parallel and serial communication services.



280129-1

Figure 1. BEM Block Diagram

*IBM is a trademark of International Business Machines Corporation.

October 1987
Order Number: 280129-003

OPERATING ENVIRONMENT

Introduction

The BITBUS Interconnect Serial Control Bus Specification defines an integrated architecture optimized for implementing real-time distributed control systems. The architecture includes a message structure and protocol for multitasking environments, and a predefined interface for I/O access and control. As with traditional bus specifications the mechanical, electrical, and data protocols have been defined. Over a twisted pair of wires the bus can support up to 250 nodes at three different bit rates dependent on application performance requirements. Figure 2 illustrates the BITBUS Interconnect architecture.

The 8044 BITBUS Enhanced Microcontroller (BEM) or DCM Controller provides the user with the smallest BITBUS building block—a BITBUS component solution. With its dual processor architecture, this unique single chip provides both communication and computational engines (Figure 3). Real-time control and computational power are provided by the on-chip 8-bit 8051 CPU. The Serial Interface Unit (SIU) executes a majority of the communications functions in hardware resulting in a high performance solution for distributed control applications where communication and processing power are equally important. The BEM's firmware implements the BITBUS message structure and protocol, and the pre-defined I/O command set.

Firmware

The 8044 microcontroller requires specific hardware to interface to BITBUS. The BEM's firmware also requires a particular hardware environment in order to execute correctly, just as the iDCX 86 Operating System or other operating systems required a specific hardware environment, i.e., interrupt controller, timers, etc. Based upon the hardware provided, Basic or Extended firmware environments result.

The Basic firmware environment supports the minimum configuration for the BEM to execute as a

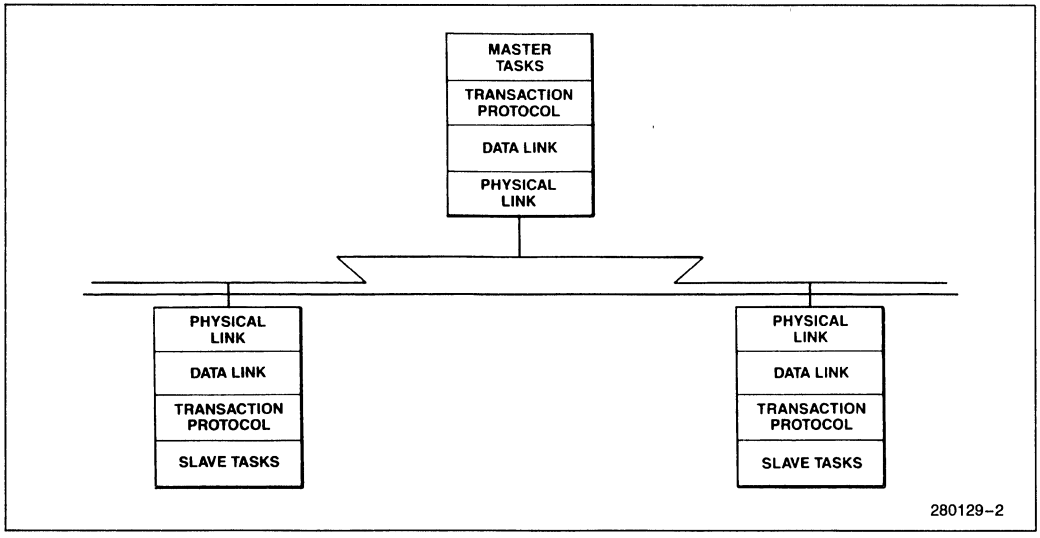


Figure 2. BITBUS™ Architecture

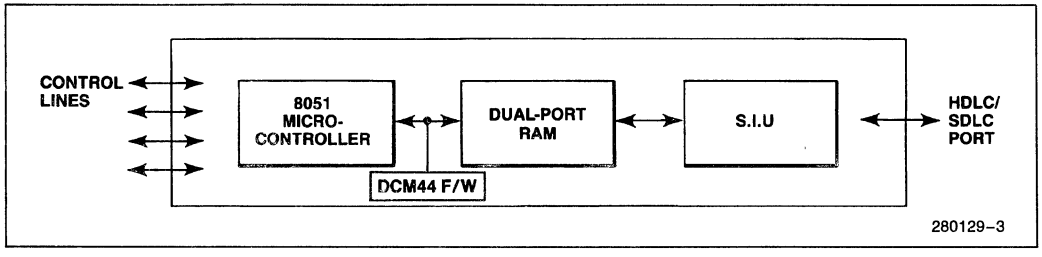


Figure 3. 8044's Dual Processor Architecture

BITBUS device. The Extended firmware environment requires hardware incremental to the Basic environment and allows the user to take full advantage of all the features included in the BEM's firmware. The designer may implement the Basic or Extended firmware environment as desired as long as the programmatic requirements of the firmware are met (see below).

Figure 4 shows one example of an Extended firmware environment. This particular example represents the BITBUS Core as used on Intel's iSBX™ 344A BITBUS Controller MULTIMODULE™ Board and iRCB 44/10A BITBUS Remote Controller Board.

BASIC FIRMWARE ENVIRONMENT	
Memory Bus	Parallel ports of 8044
BITBUS Node Address	0FFFFH external data space
Configuration	0FFFEH external data space
System RAM	0–02FFH external data space
Diagnostic LED #1	Port 1.0 (Pin 1)
Diagnostic LED #2	Port 1.1 (Pin 2)
EXTENDED FIRMWARE ENVIRONMENT	
Memory Bus	Parallel ports of 8044
BITBUS Node Address	0FFFFH external data space

EXTENDED FIRMWARE ENVIRONMENT (Continued)	
Configuration	0FFFEH external data space
System RAM	0–02FFH external data space
Diagnostic LED #1	Port 1.0 (Pin 1)
Diagnostic LED #2	Port 1.1 (Pin 2)
User Task Interface	First Task Descriptor—0FFF0H to 0FFFFH in External data space Other Task Descriptors and User Code—01000H to 0FFEFH in external code space
User RAM Availability	On-Chip—02AH to 02FH bit space Off-Chip—BITBUS Master: 0400H to 0FFEFH external data space BITBUS Slave: 0100H to 0FFEFH external data space
Remote Access and Control Interface	Memory-Mapped I/O—0FF00H to 0FFFFH external data space

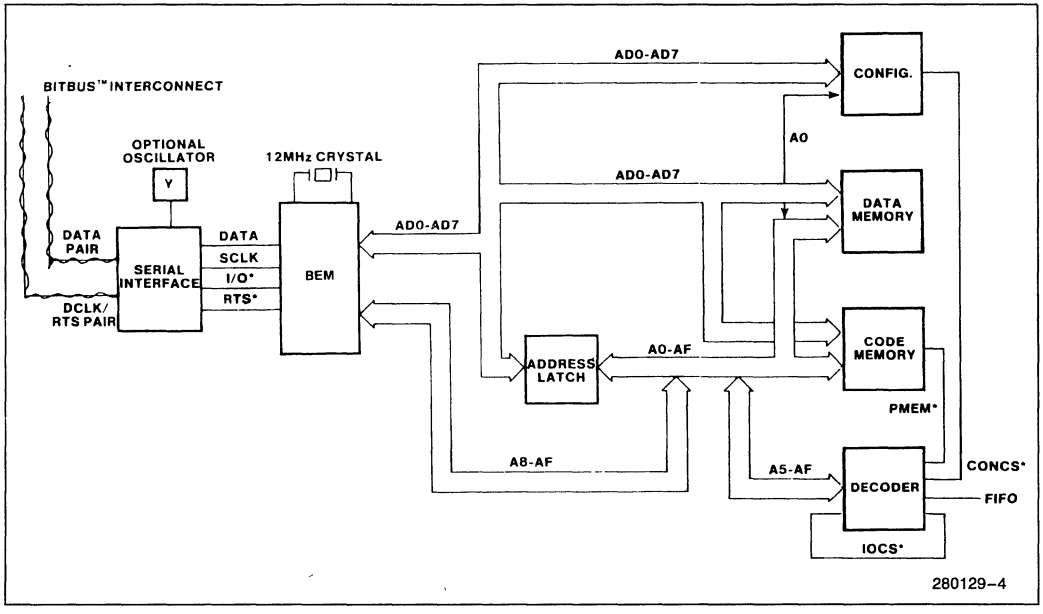


Figure 4. Extended Firmware Environment Example

EXTENDED FIRMWARE ENVIRONMENT (Continued)	
Parallel Interface to Extension Device	FIFO Command Byte—0FF01H external data space FIFO Data Byte—0FF00H external data space Receive Data Intr—INT0 (pin 12) Transmit Data Intr—INT1 (pin 13) Command/Data Bit—P1.2

FUNCTIONAL DESCRIPTION

High Performance 8044 Microcontroller

The 8044 combines the powerful 8051 microcontroller with an intelligent serial communications controller to provide a single-chip solution that efficiently implements distributed processing or distributed control systems. The microcontroller is a self-sufficient unit containing ROM, RAM, ALU, and peripherals. The 8044's architecture and instruction set are identical to the 8051's. The serial interface of the 8051 is replaced with an intelligent communications processor, the Serial Interface Init (SIU), on the 8044. This unique dual processor architecture results in high performance and reliability for distributed control and processing environments. The intelligent SIU offloads the CPU from communication tasks, thus dedicating more of its compute power to external processes.

Major features of the 8051 microcontroller are:

- 8-bit CPU
- On-chip oscillator
- 4K bytes of RAM
- 192 bytes of ROM
- 32 I/O lines
- 64K address space external data memory
- 64K address space external program memory
- Two Programmable 16-bit counters
- Five source interrupt structure with two priority levels
- Bit addressability for Boolean functions
- 1 μs instruction cycle time for 60% instructions
  2 μs instruction cycle time for 40% instructions
- 4 μs cycle time for 8 by 8 unsigned multiple and divide

As noted in the Operating Environment discussion, the BITBUS firmware requires various CPU resources, i.e., memory, timers, and I/O dependent upon the firmware environment selected.

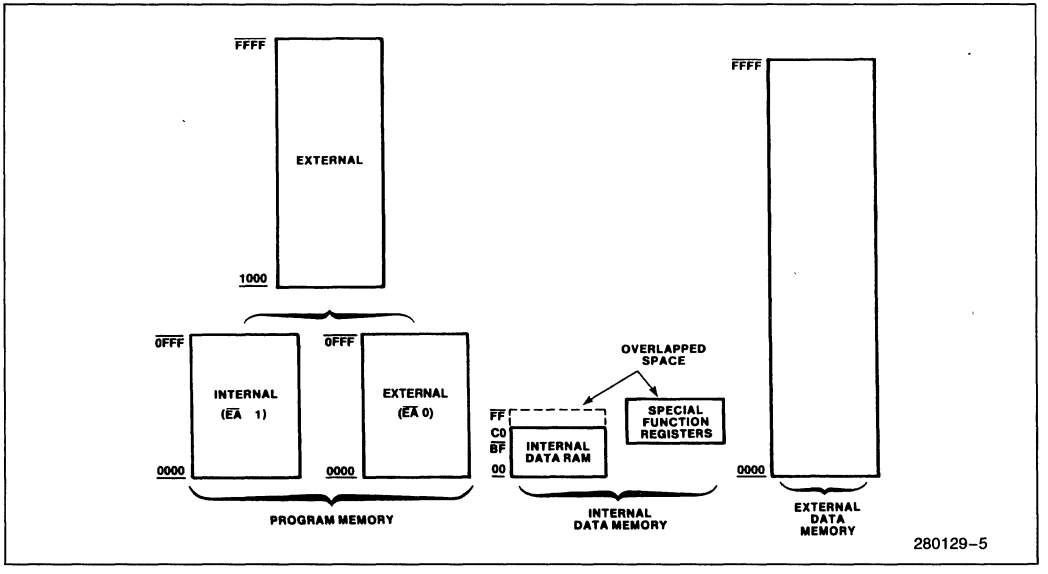


Figure 5. BEM Memory Map

Memory Architecture

The 8044 microcontroller maintains separate data and code memory spaces. Internal data memory and program memory reside on the controller. External memory resides outside the controller. The BEM firmware uses the available internal code memory space and most of the remaining internal data memory with the exception of bit space 02AH to 02FH. Figure 5 shows the BEM's memory map.

I/O ADDRESSING REQUIREMENTS

The table below provides the BEM's I/O port addresses.

Table 1. BEM I/O Addressing

Function	Address	Bit	Byte
Red LED P1.0	90H	X	
Green LED P1.1	91H	X	
TCMD	92H	X	
RFNF#	B3H	X	
TFNF#	B2H	X	
RDY/NE*	B4H	X	
Node Address	FFFFH		X
Configuration	FFFEH		X
Reserved	FFE0H–FFFDH		X
Digital I/O	FFC0H–FFDFH		X
SBX #4	FFB0H–FFBFH		X
SBX #3	FFB0H–FFAFH		X
SBX #2	FF90H–FF9FH		X
SBX #1	FF80H–FF8FH		X
User Defined	FF40H–FF7FH		X
Reserved	FF02H–FF3FH		X
FIFO Command	FF01H		X
FIFO Data	FF00H		X

SIGNAL FUNCTIONS

The 8044 BEM's pin configuration and pin description follow.

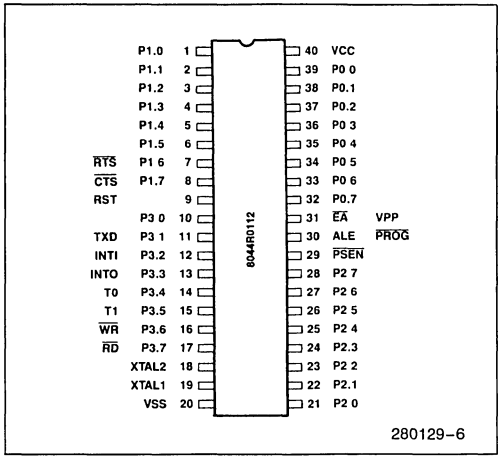


Figure 6A. BEM DIP Pin Configuration

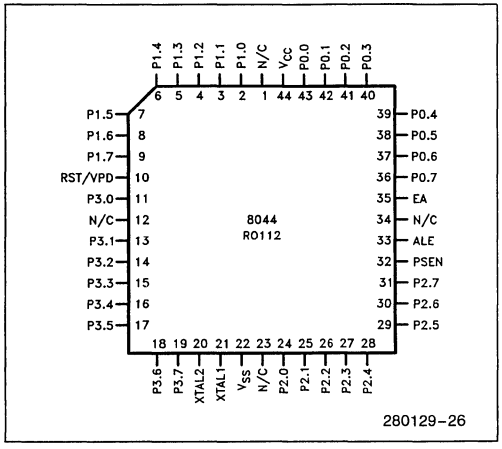


Figure 6B. BEM PLCC Pin Configuration

Table 2. BEM Pin Description

Name	Description
VSS	Circuit ground potential.
V_{CC}	+5V power supply during operation and program verification.
PORT 0	Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source eight LS TTL loads.
PORT 1	Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during program verification. Port 1 can sink/source four LS TTL loads. In non-loop mode two of the I/O lines serve alternate functions: —$\overline{RTS}$ (P1.6) Request-to Send output. A low indicates that the 8044 is ready to transmit. —$\overline{CTS}$ (P1.7) Clear-to-Send input. A low indicates that a receiving station is ready to receive.

Table 2. BEM Pin Description (Continued)

Name	Description
PORT 2	Port 2 is an 8-bit quasi-bidirection I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source four LS TTL loads.
PORT 3	Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and RD and WR pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source four LS TTL loads. In addition to I/O some of the pins also serve alternate functions as follows: • I/$\overline{O}$ R x D (P3.0). In point-to-point or multipoint configurations, this pin controls the direction of pin P3.1. Serves as Receive Data input in loop and diagnostic modes. • DATA T x D (P3.1). In point-to-point or multipoint configurations, this pin functions as data input/output. In loop mode, it serves as transmit pin. A '0' written to this pin enables diagnostic mode. • $\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0. • $\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1. • TO (P3.4). Input to counter 0. • SCLK T1 (P3.5). In addition to I/O, this pin provides input to counter 1 or serves as SCLK (serial clock) input. • $\overline{WR}$ (P3.6). The write control signal latches the data byte from Port 0 into the External Data Memory. • $\overline{RD}$ (P3.7). The read control signal enables External Data Memory to Port 0.
RST	A high on this pin for two machine cycles while the oscillator is running resets the device. A small external pulldown resistor ($\approx$ 8.2 KΩ) from RST to VSS permits power-on reset when a capacitor ($\approx$ 10 μf) is also connected from this pin to V_{CC}.
ALE/$\overline{PROG}$	Provides Address Latch Enable output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access. It also receives the program pulse input for programming the EPROM version.
$\overline{PSEN}$	The Program Store Enable output is a control signal that enables the external Program Memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.
$\overline{EA}$/VPP	When held at a TTL high level, the 8044 executes instructions from the internal ROM when the PC is less than 4096. When held at a TTL low level, the 8044 fetches all instructions from external Program Memory. The pin also receives the 21V EPROM programming supply voltage on the 8744.
XTAL 1	Input to the oscillator's high gain amplifier. Required when a crystal is used. Connect to VSS when external source is used on XTAL 2.
XTAL 2	Output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.

Firmware

The BEM's Basic firmware environment provides two services: BITBUS Communications and Power-Up Diagnostics. The Extended firmware environment provides the Basic firmware services plus Parallel Communications and User Software Services (iDCX 51 Executive, Remote Access and Control functions). A discussion of each service follows.

Basic Firmware Services

POWER-UP DIAGNOSTICS INCREASE RELIABILITY

For added reliability and simplified system start up, the BEM firmware includes power-up diagnostics. At chip reset the BEM diagnostic firmware checks the integrity of the 8044's instruction set, ROM, internal

RAM, and external RAM. LED indicator lights may be used to show the progress of the diagnostics. Intel's BITBUS boards use one red LED, and one green LED as indicators for test progress. Since the test halts if a fault is found, the last LED state indicates the trouble area.

No programmatic interface exists for the power-up diagnostics. Only LEDs (or other indicators) connected to the outputs of Port 1 of the 8044 are required. For the test sequence shown in Table 3, the red LED is connected to pin P1.0, and the green LED is connected to pin P1.1.

Table 3. Power-Up Test Sequence

Test Sequence	State of Port* After Test Completion	
	Red LED (Pin 1.0)	Green LED (Pin 1.1)
Power-on	On	On
Prior to Start of Tests	Off	Off
Test 1—Instruction Set	On	On
Test 2—ROM Checksum Test	On	Off
Test 3—Internal RAM	Off	Off
Test 4—External RAM	Off	On

*Ports are Active Low.

BITBUS™ INTERFACE SIMPLIFIES DESIGN OF DISTRIBUTED CONTROL SYSTEMS

The BITBUS Serial Control Bus is a serial bus optimized for high speed transfer of short messages in a hierarchical system. From the perspective of systems using the BITBUS bus there are three external protocols that must be adhered to: physical, data link, and transaction control as shown in Figure 2. The physical interface includes all bus hardware requirements, e.g. cable and connector definition, transceiver specification. The data link interface refers to the device to device transfer of frames on the bus. The transaction control interface indentifies the rules for transmitting messages on the bus as well as the format of the messages passed.

For maximum reliability and to facilitate standardization the following existing standards were chosen as portions of the BITBUS Specification: International Electrotechnical Commission (IEC) mechanical board and connector specifications, the Electronic Industry Association (EIA) RS-485 Electrical Specification and IBM*'s Serial Data Link Control protocol for the physical and data link levels of the BITBUS interface.

BITBUS™ Physical Interface

Implementation of the electrical interface to BITBUS requires external hardware. Specifically, an EIA Standard RS-485 driver and transceiver and an optional clock source for the synchronous mode of operation. A self clocked mode of operation is also available. Different modes of operation facilitate a variety of performance/distance options as noted in Table 4. Figure 7 illustrates the BEM's BITBUS interface hardware requirements.

Table 4. BITBUS™ Interconnect Modes of Operation

	Speed Kb/s	Max. Dist Between Repeaters M/ft	Max # Nodes Between Repeaters	Max # Repeaters
Synchronous	2400	30/100	28	0
Self-Clocked	375 62.5	300/1000 1200/4000	28 28	2 10

BITBUS™ Data Link Service

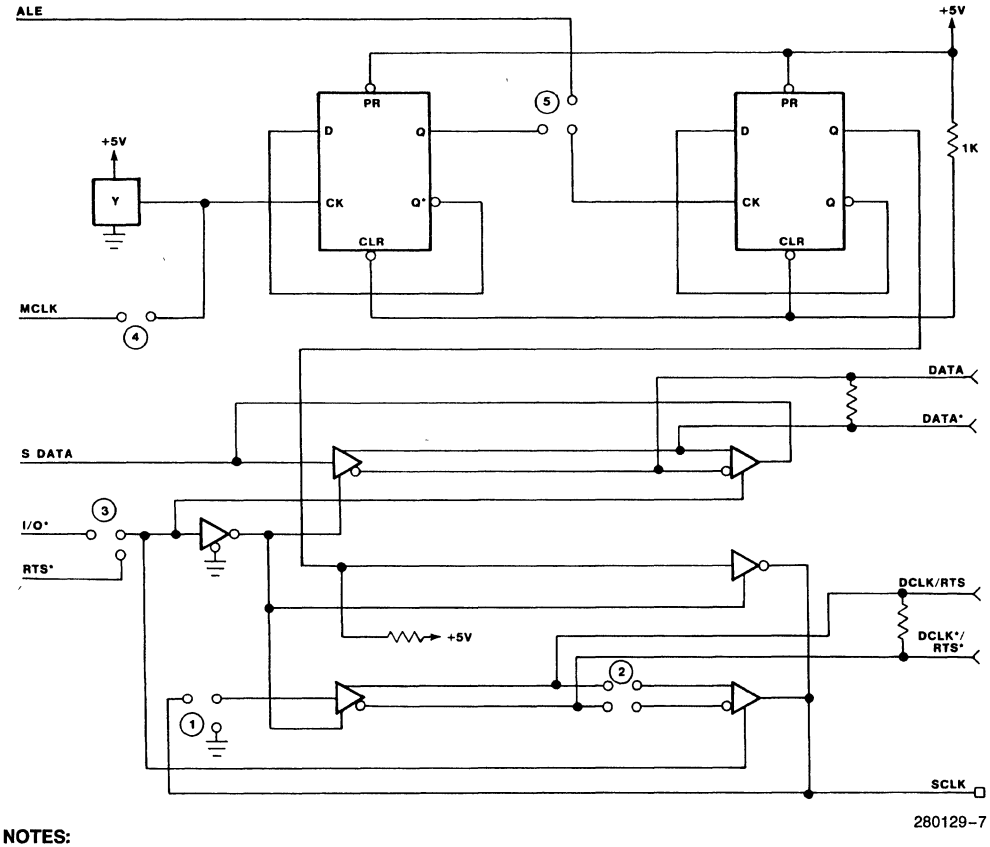
The 8044's serial interface unit (SIU) implements a majority of the data link interface, a subset of IBM's Serial Data Link Protocol (SDLC), in hardware resulting in a significant performance advantage compared with multichip solutions. Multichip solutions require both hardware and software glue that degrade performance, decrease reliability, and increase cost. This portion of the BITBUS interface requires no user involvement for execution.

For a detailed discussion of the protocol executed by the BITBUS data link service refer to "The BITBUS Interconnect Serial Control Bus Specification". A basic subset of SDLC with the REJECT option is implemented. The standard frame format transferred across the BITBUS is shown in Figure 8. The information field carries the BITBUS message.

BITBUS™ Transaction Control Service

For added reliability, the BITBUS interface incorporates error checking at the message level in addition to the imbedded error checking provided by SDLC at the data link level. The message control interface defines the format and function of messages transmitted in frames across the BITBUS bus. (Figure 9)

The transaction protocol requires that for every order message transmitted across the bus a reply message must be transmitted in return. Error types and error detection mechanisms are also designated by this interface.



NOTES:
1. Connect to ground for self-clocked mode and SCLK for synchronous mode.
2. Remove for self-clocked operation with repeater(s).
3. Connect to RTS* for synchronous mode or I/O* for self-clocked mode.
4. Selects MCLK as serial clock source.
5. Selects ALE or oscillator as serial clock source.

Figure 7. BITBUS™ Interface Hardware Requirements

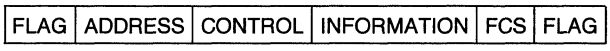


Figure 8. BITBUS™ Frame Format

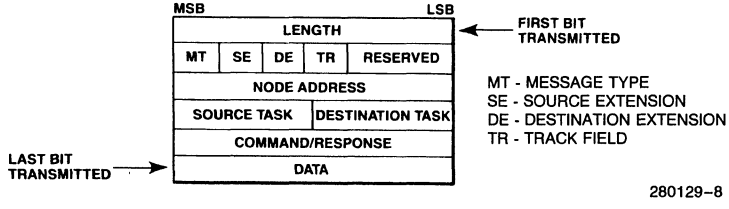


Figure 9. BITBUS™ Message Format

BITBUS™ Interface Configuration

The BEM's firmware also simplifies designation of the bus mode of operation (Speed/distance option) as well as the node address, memory configuration and parallel interface parameters by reading two external locations for this information as shown in Figure 10. The designer no longer needs to directly manipulate the 8044's serial mode register (SMD), status/command register (STS), and send/receive counter register (NSNR). These two 8-bit locations are derived by multiplexing the 8044's port 0 address lines AD0–AD7.

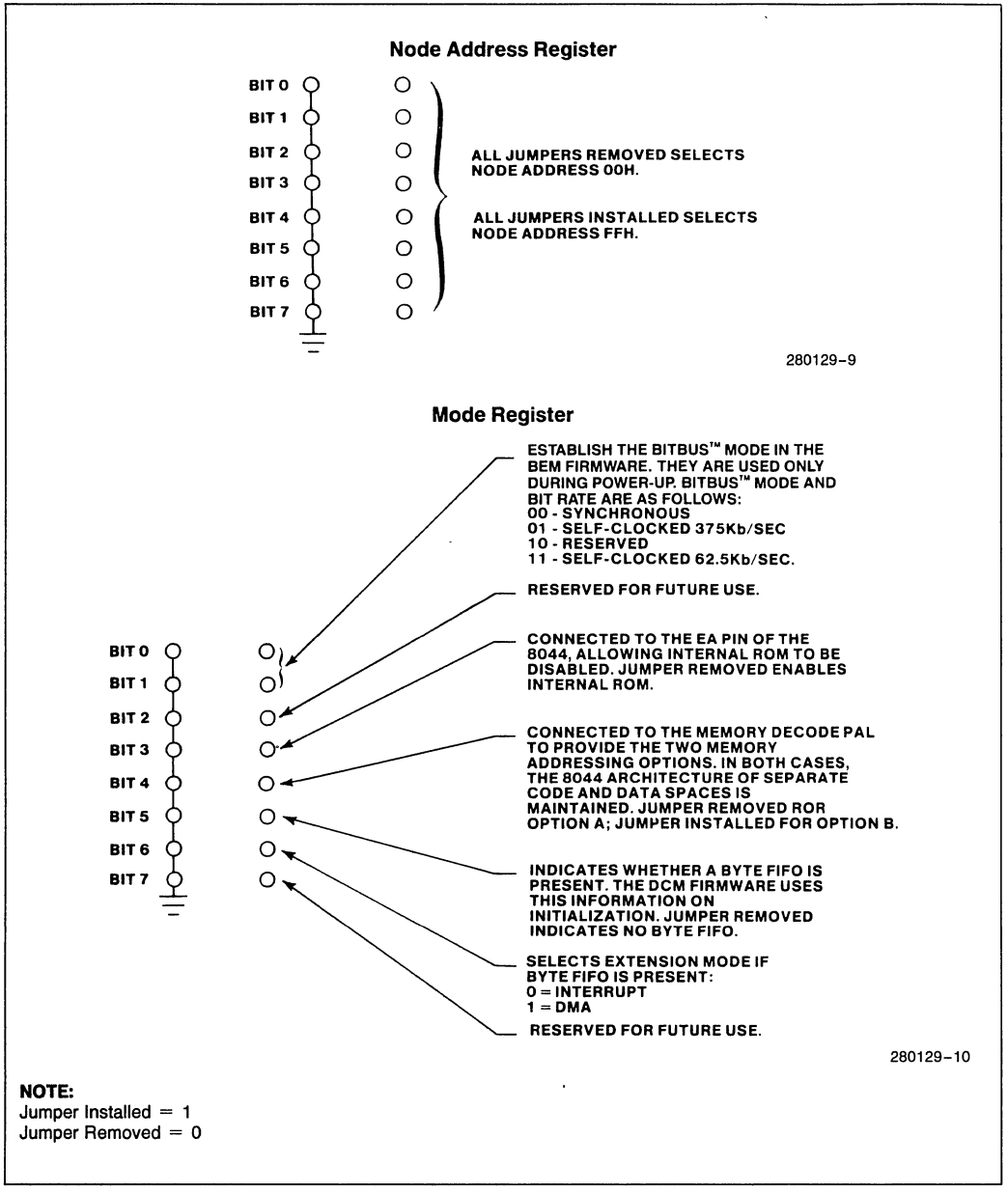


Figure 10. BITBUS™ Firmware Configuration

Extended Firmware Services

PARALLEL COMMUNICATION INTERFACE EXTENDS DISTRIBUTED CONTROL CAPABILITY

The BEM's firmware also includes a parallel interface for expanding the capabilities of distributed systems. For example, this interface allows other processors to be employed in BITBUS systems if more processing power is required as shown in Figure 11. This interface provides the means for connection to other buses: iSBX bus, STD bus, IBM's PC bus.

The interface consists of a byte-FIFO queue through which BITBUS messages can be passd via embedded communications firmware. From the BEM's perspective the user simply designates the correct routing information in the BITBUS message header and the message is directed to the communications firmware and passed through the parallel interface. One example of an implementation that uses this interface is the iSBX BITBUS Controller MULTIMODULE Board via the iSBX bus.

Parallel Interface Hardware

To implement the Parallel Interface, the user must provide hardware for two FIFOs (one byte minimum) in external data memory, and control signals to/from the 8044's Pins: INT0 (P3.2), INT1 (P3.3), and P1.2. Key hardware elements required are: decoder for the registers' external addresses, temporary storage for bytes passing through the interface, a way to designate bytes as command or data, and a means to generate the control signals. FIFO's must be used to move the data through the interface although the depth of the FIFO need not exceed one byte.

Interface hardware must also be provided for the "extension" side of the interface. Implementation of this hardware is left to the user with the restriction that the operation of the BEM side remains independent.

Parallel Byte Stream and Message Protocol

The two byte registers (FIFOs) provide the path for bytes to move through the parallel interface. Bytes are read or written from the registers designated: FIFO Data Byte (FF00H) and FIFO Command Byte (FF01H). INT0, INT1 and P1.2 provide control signals to the firmware for moving the bytes through the registers. These signals are referred to as the Parallel Interface Control Bits:

Pin	Function	Internal Bit Address
INT0	RFNF	B3H
INT1	TFNE	B2H
P1.2	TCMD	92H

The hardware uses RFNF to control the output of bytes from the BEM. RFNF is set when the FIFO Data or FIFO Command Byte Registers can receive information. RFNF remains clear when the FIFO Data or Command Bytes are not available. Transmission of a BITBUS message across the parallel interface consists of successively outputing message bytes to the FIFO Data Byte Register until all bytes are sent. The firmware then writes a value of 0 to the Command Byte register indicating all the message bytes have been sent. The first data byte in the message indicates the number of bytes in the message.

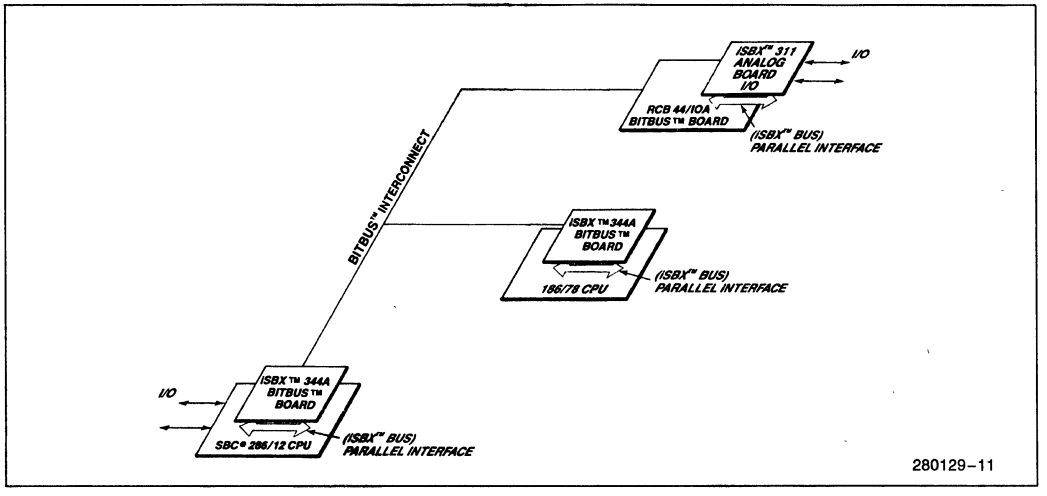


Figure 11. Extending the Capability of BITBUS™ System with the Parallel Communications Interface

TFNE controls the input of data bytes to the BEM. This bit is set when bytes are available for reading. When no bytes are available this bit is clear. TCMD indicates whether the next byte read is a Data Byte or Command Byte. BITBUS messages are received by inputing data bytes until a command byte is received. Data bytes are read from the FIFO Data Byte Register. Command Bytes are read from the FIFO Command Byte Register.

Figure 12 provides one example of a Byte FIFO Interface. This specific example illustrates the interface provided on the iSBX 344A BITBUS Controller MULTIMODULE Board. Figure 13 shows transmission of bytes from the BEM across the parallel interface. Figure 14 shows transmission of bytes to the BEM.

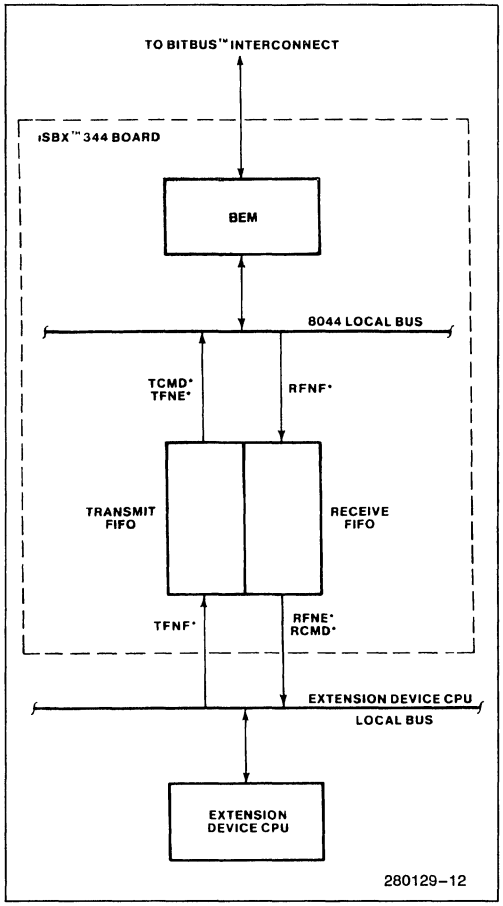


Figure 12. Byte FIFO Interface Example

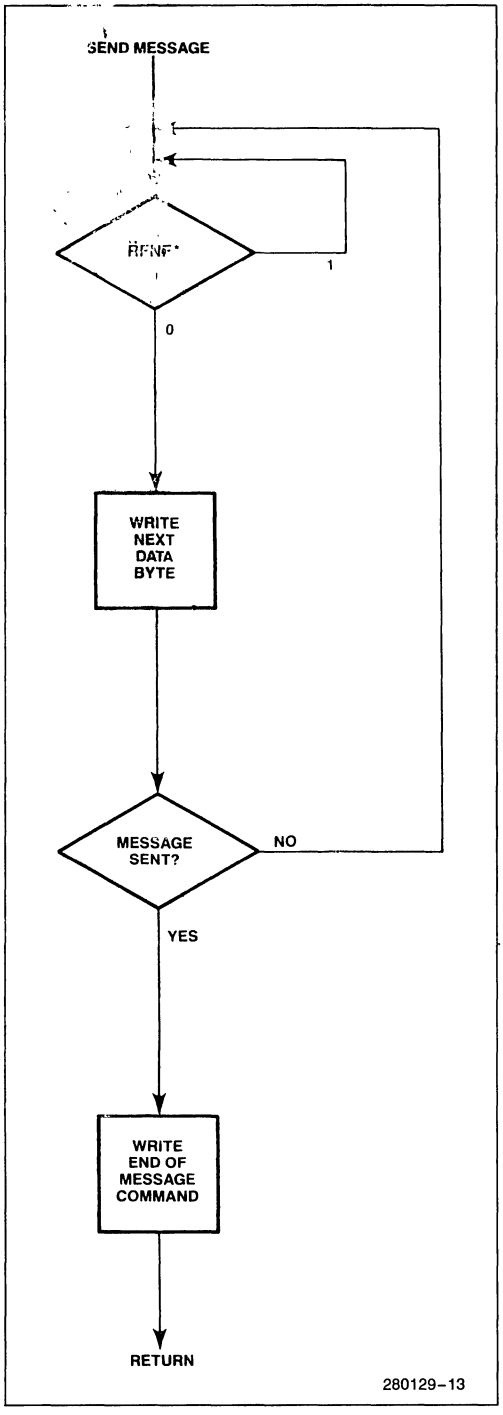


Figure 13. Transmitting a Message from BEM

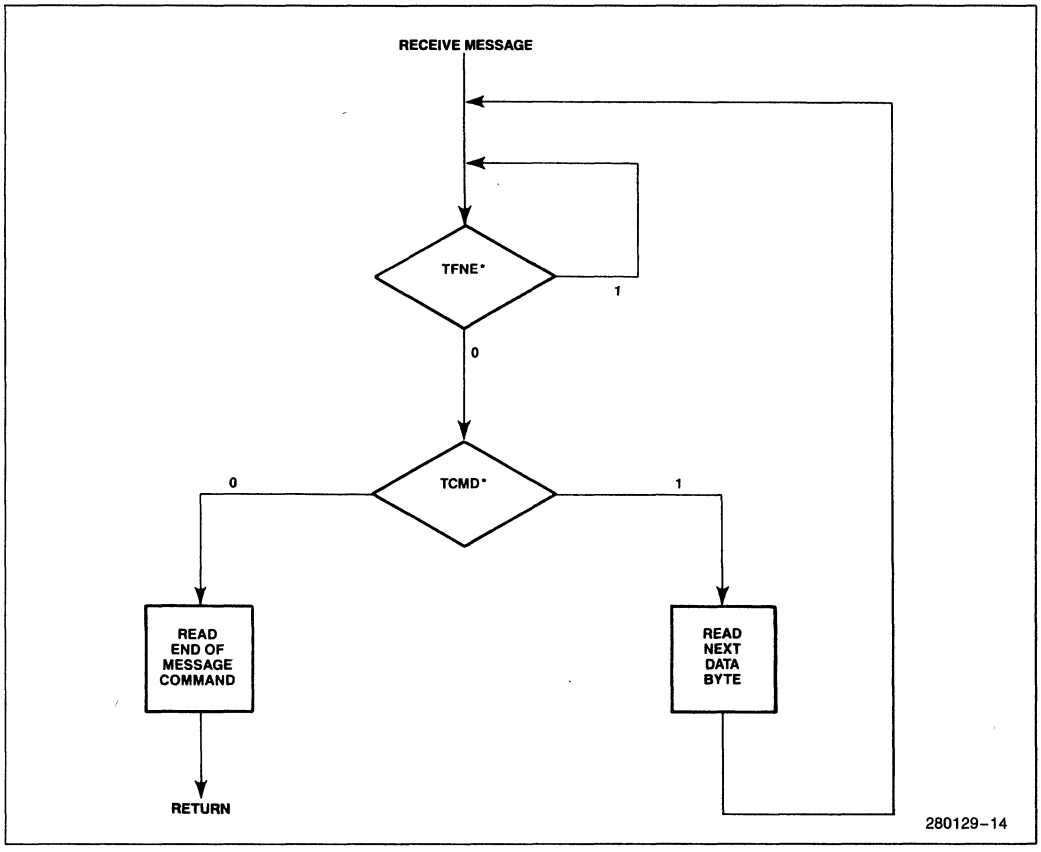


Figure 14. Transmitting a Message to BEM

USER SOFTWARE SERVICES

Multitasking, I/O Access and Control Capabilities

The Extended firmware environment of the BEM provides a multitasking facility via the iDCX 51 Real-time, Multitasking Executive. Operating system calls are listed in Table 5. Other services provided by the Executive: interrupt handling, task scheduling, and intertask communication facilitate smooth development of distributed systems. In addition to the Executive's intertask communication service provided by the RQSENDMESSAGE call, other portions of the firmware extend the communication capability across the parallel and BITBUS interfaces. This embedded communications firmware greatly simplifies and speeds sending messages to different microcontrollers or microprocessors in the system.

To further ease the development of distributed control applications, a pre-defined task (Remote Access and Control Task) provides the means of invoking iDCX 51 Executive services, or accessing I/O and memory from tasks on other devices. The Remote Access and Control functions execute under the iDCX 51 Executive as Task 0. Figure 13 illustrates this concept in a BITBUS system. Table 6 shows the functions provided by the RAC task. All I/O command accesses are memory mapped to locations 0FF00H to 0FFFFH in the BEM's external memory.

Table 5. iDCX™ 51 Calls

Call Name	Description
TASK MANAGEMENT CALLS	
RQ$CREATE$TASK	Create and schedule a new task.
RQ$DELETE$TASK	Delete specified task from system.
RQGETFUNCTION$IDS	Obtain the function IDs of tasks currently in system.
INTERTASK COMMUNICATION CALLS	
RQ$ALLOCATE	Obtain a message buffer from the system buffer pool.
RQ$DEALLOCATE	Return a message buffer to the system buffer pool.
RQ$SEND$MESSAGE	Send a message to specified task.
RQ$WAIT	Wait for a message event.
MEMORY MANAGEMENT CALLS	
RQGETMEM	Get available system memory pool memory.
RQ$RELEASE$MEM	Release system memory pool memory.
INTERRUPT MANAGEMENT CALLS	
RQ$DISABLE$INTERRUPT	Temporarily disable an interrupt.
RQ$ENABLE$INTERRUPT	Re-enable an interrupt.
RQ$WAIT	Wait for an interrupt event.
TIMER MANAGEMENT CALLS	
RQSETINTERVAL	Establish a time interval.
RQ$WAIT	Wait for an interval event.

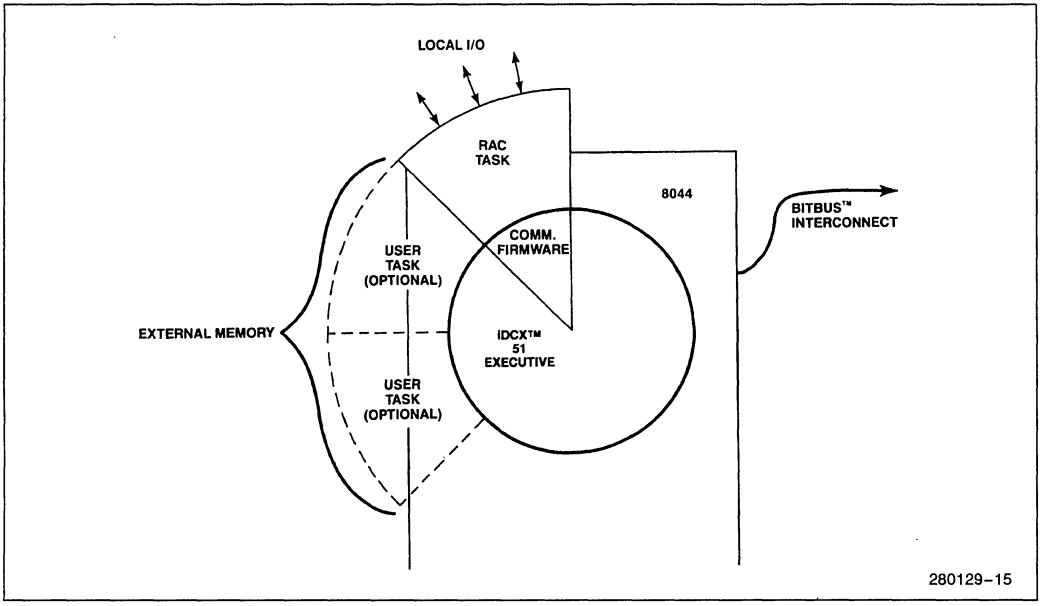


Figure 15. BEM Communication Firmware

Table 6. RAC Functions

Name	Function
RESET_STATION	Perform a software reset.
CREATE_TASK	Perform an RQ$CREATE$TASK system call.
DELETE_TASK	Perform an RQ$DELETE$TASK system call.
GET_FUNCTION_ID	Perform an RQGETFUNCTION$IDS call.
RAC_PROTECT	Suspend or resume RAC services.
READ_IO	Return values from specified I/O ports.
WRITE_IO	Write to the specified I/O ports.
UPDATE_IO	Update the specified I/O ports.
UPLOAD_MEMORY	Return the values in specified memory area.
DOWNLOAD_MEMORY	Write values to specified memory area.
OR_I/O	OR values into specified I/O ports.
AND_I/O	AND values into specified I/O ports.
XOR_I/O	XOR values into specified I/O ports.
READ_INTERNAL	Read values at specified internal RAM areas.
WRITE_INTERNAL	Write values to specified internal RAM areas.
NODE_INFO	Return device related information.
OFFLINE	Set node offline.
UPLOAD_CODE	Read values from code memory space.
DOWNLOAD_CODE	Write values to specified EEPROM memory.

NOTES:
Internal memory locations are included in the 192 bytes of data RAM provided in the microcontroller. External memory refers memory outside the microcontroller—the 28-pin sockets of the iSBX 344A module and the iRCB 44/10A and iRCB 44/20A boards. Each RAC Access Function may refer to 1, 2, 3, 4, 5, or 6 individual I/O or memory locations in a single command.

In addition to allowing creation and deletion of tasks on remote system nodes, the RAC functions allow memory upload and download. This feature eases programming changes in distributed systems and enhances overall system flexibility. Diagnostics can also be downloaded to remote nodes to facilitate system debug.

Another feature optimized for distributed control environments is the GET FUNCTION IDS service. The function ID capability provides the user with the ability to identify specific tasks by function rather than node address and task number. This constant identifier facility remains valid even if functions are moved to different physical locations, eg. another system node.

Aside from the iDCX 51 Executive system calls the user interfaces to the BEM through the task initialization interface; the Initial Task Descriptor. The first user task descriptor must be located at location 0FFF0H in external memory code space so that on power up user code may be automatically detected.

The Initial Task Descriptor (ITD) allows the user to specify the original attributes of a task. Table 7 shows the ITD task structure.

Table 7. ITD Structure

Pattern	Word	value identifying an ITD: "AA55H"
Initial PC	Word	address of first task instruction
Stack-Length	Byte	# bytes of system RAM for tasks stack
Function ID	Byte	value 1–255 associates task w/function
Register Bank	Bit(4)	assigns one register bank to task
Priority	Bit(4)	task priority level
Interrupt Vector	Word	specifies interrupt associated w/task
Next ID	Word	address of the next ITD in linked-list

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0 to 70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin with
Respect to Ground (V_{SS}) .....−0.5V to +7V
Power Dissipation .....2 Watts

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%, V_{SS} = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions
VIL	Input Low Voltage	−0.5	0.8	V	
VIH	Input High Voltage (Except RST and XTAL2)	2.0	V_{CC} + 0.5	V	
VIH1	Input High Voltage to PST For Reset, XTAL2	2.5	V_{CC} + 0.5		XTAL1 = V_{SS}
VOL	Output Low Voltage Ports 1, 2, 3 (Note 1)		0.45	V	I_{OL} = 1.6 mA
VOL1	Output Low Voltage Port 0, ALE, \$\overline{PSEN}$ (Note 1)		0.45	V	I_{OL} = 3.2 mA
VOH	Output High Voltage Ports 1, 2, 3	2.4		V	I_{OH} = −80 μA
VOH1	Output High Voltage Port 0, ALE, \$\overline{PSEN}$	2.4		V	I_{OH} = −400 μA
IIL	Logical 0 Input Current Ports 1, 2, 3		−500	μA	XTAL1 at V_{SS} Vin = 0.45V
IIH1	Input High Current to RST/VPD For Reset		500	μA	Vin < V_{CC} − 1.5V
ILI	Input Leakage Current to Port 0, \$\overline{EA}$		±10	μA	0.45V < Vin < V_{CC}
ICC	Power Supply Current		170	mA	All Outputs Disconnected, EA = V_{CC}
CIO	Capacitance of \$\overline{I/O}$ Buffer		10	pF	fc = 1 MHz
IIL2	Logical 0 Input Current XTAL2		−3.6	mA	XTAL1 at V_{SS} Vin = 0.45V

NOTE:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VOLs of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.

A.C. CHARACTERISTICS T_A to 0°C to 70°C, V_{CC} = 5V ±10%, V_{SS} = 0V, C_L for Port 0, ALE and $\overline{PSEN}$ Outputs = 100 pF; C_L for All Other Outputs = 80 pF

PROGRAM MEMORY

Symbol	Parameter	12 MHz Clock			Variable Clock 1/TCLCL = 3.5 MHz to 12 MHz		
		Min	Max	Units	Min	Max	Units
TLHLL	ALE Pulse Width	127		ns	2TCLCL−40		ns
TAVLL	Address Setup to ALE	43		ns	TCLCL−40		ns
TLLAX(1)	Address Hold after ALE	48		ns	TCLCL−35		ns
TLLIV	ALE to Valid Instr in		233	ns		4TCLCL−100	ns
TLLPL	ALE to $\overline{PSEN}$	58		ns	TCLCL−25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	215		ns	3TCLCL−35		ns
TPLIV	$\overline{PSEN}$ to Valid Instr in		125	ns		3TCLCL−125	ns
TPXIX	Input Instr Hold after $\overline{PSEN}$	0		ns	0		ns
TPXIZ(2)	Input Instr Float after $\overline{PSEN}$		63	ns		TCLCL−20	ns
TPXAV(2)	Address Valid after $\overline{PSEN}$	75		ns	TCLCL−8		ns
TAVIV	Address to Valid Instr in		302	ns		5TCLCL−115	ns
TAZPL	Address Float to $\overline{PSEN}$	−25		ns	−25		ns

NOTES:
1. TLLAX for access to program memory is different from TLLAX for data memory.
2. Interfacing RUPI-44 devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to Port 0 drivers.

EXTERNAL DATA MEMORY

Symbol	Parameter	12 MHz Clock			Variable Clock 1/TCLCL = 3.5 MHz to 12 MHz		
		Min	Max	Units	Min	Max	Units
TRLRH	$\overline{RD}$ Pulse Width	400		ns	6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		ns	6TCLCL−100		ns
TLLAX(1)	Address Hold after ALE	48		ns	TCLCL−35		
TRLDV	$\overline{RD}$ to Valid Data in		252	ns		5TCLCL−165	ns
TRHDX	Data Hold after $\overline{RD}$	0		ns	0		ns
TRHDZ	Data Float after $\overline{RD}$		97	ns		2TCLCL−70	ns
TLLDV	ALE to Valid Data in		517	ns		8TCLCL−150	ns
TAVDV	Address to Valid Data in		585	ns		9TCLCL−165	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	200	300	ns	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	203		ns	4TCLCL−130		ns
TWHLH	$\overline{WR}$ or $\overline{RD}$ High to ALE High	43	123	ns	TCLCL−40	TCLCL+40	ns
TQVWX	Data Valid to $\overline{WR}$ Transition	23		ns	TCLCL−60		ns
TQVWH	Data Setup before $\overline{WR}$	433		ns	7TCLCL−150		ns
TWHQX	Data Hold after $\overline{WR}$	33		ns	TCLCL−50		ns
TRLAZ	RD Low to Address Float		25	ns		25	ns

NOTE:
1. TLLAX for access to program memory is different from TLLAX for access data memory.

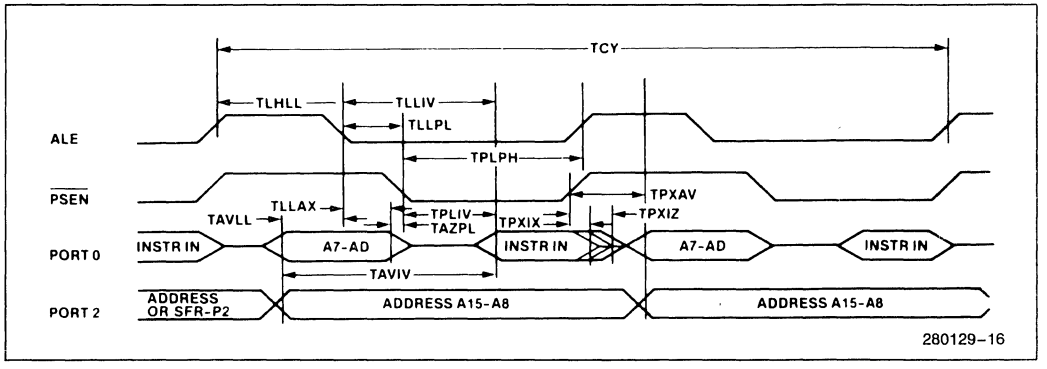
SERIAL INTERFACE

Symbol	Parameter	Min	Max	Units
TDCY	Data Clock	420		ns
TDCL	Data Clock Low	180		ns
TDCH	Data Clock High	100		ns
tTD	Transmit Data Delay		140	ns
tDSS	Data Setup Time	40		ns
tDHS	Data Hold Time	40		ns

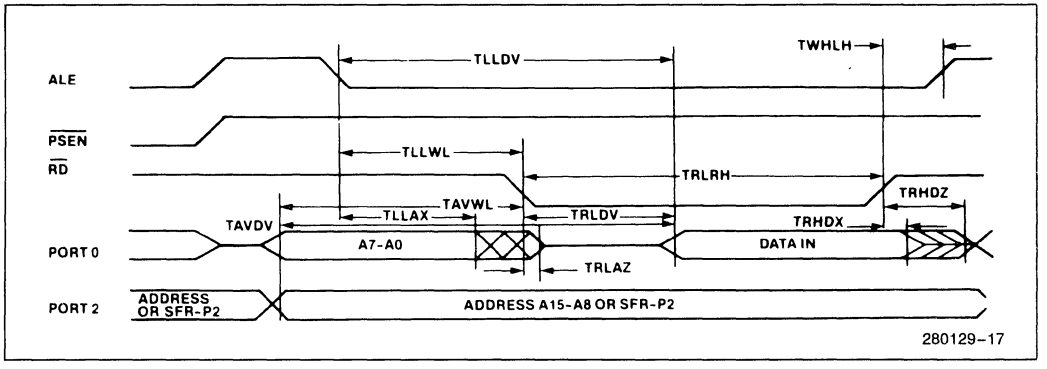
WAVEFORMS

Memory Access

PROGRAM MEMORY READ CYCLE

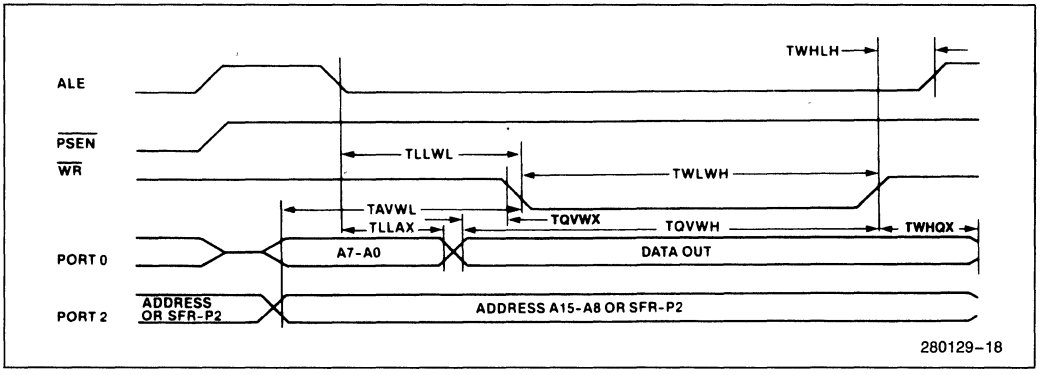


DATA MEMORY READ CYCLE



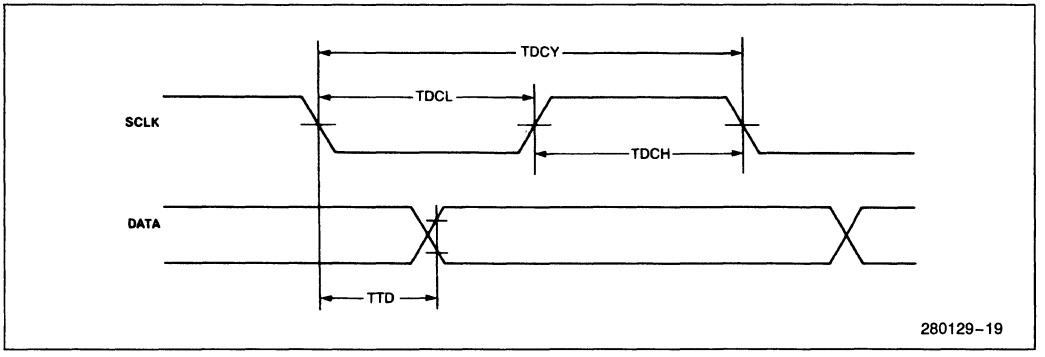
WAVEFORMS (Continued)

DATA MEMORY WRITE CYCLE

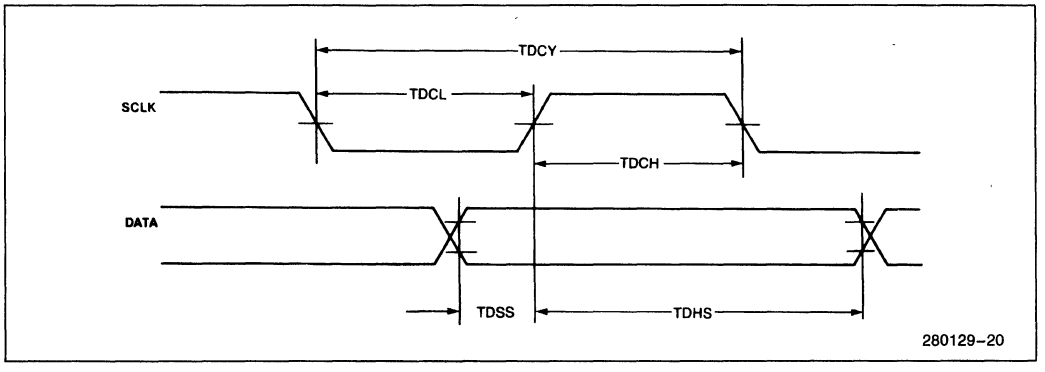


SERIAL I/O WAVEFORMS

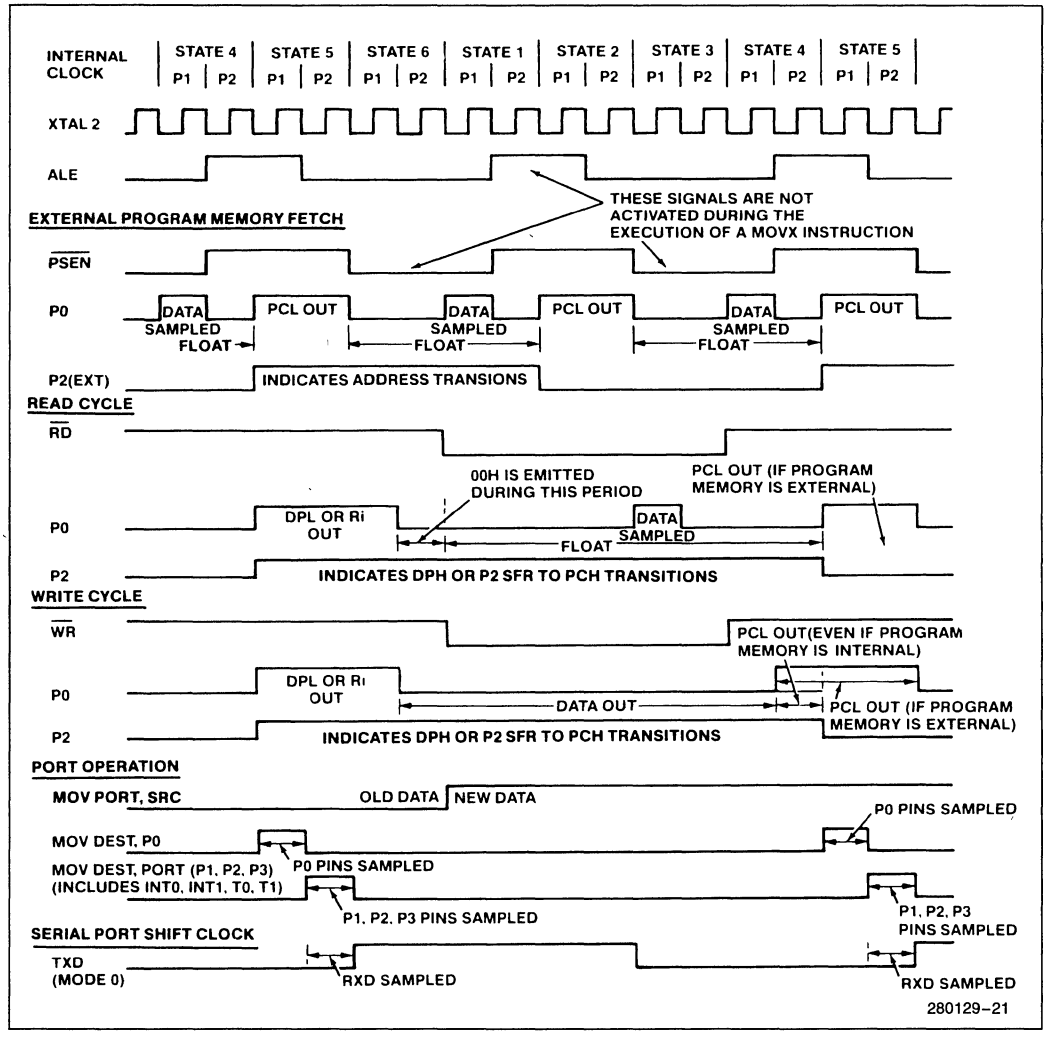
SYNCHRONOUS DATA TRANSMISSION



SYNCHRONOUS DATA RECEPTION

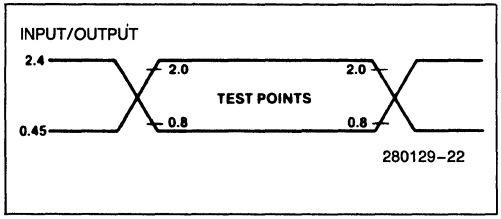


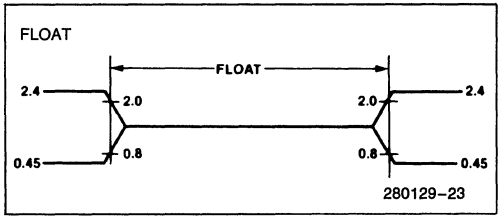
CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component to component. Typically though, (T_A = 25°C, fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

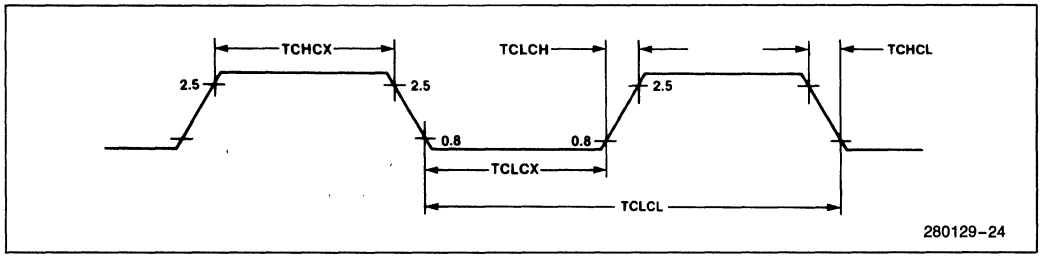
A.C. TESTING INPUT, OUTPUT, FLOAT WAVEFORMS





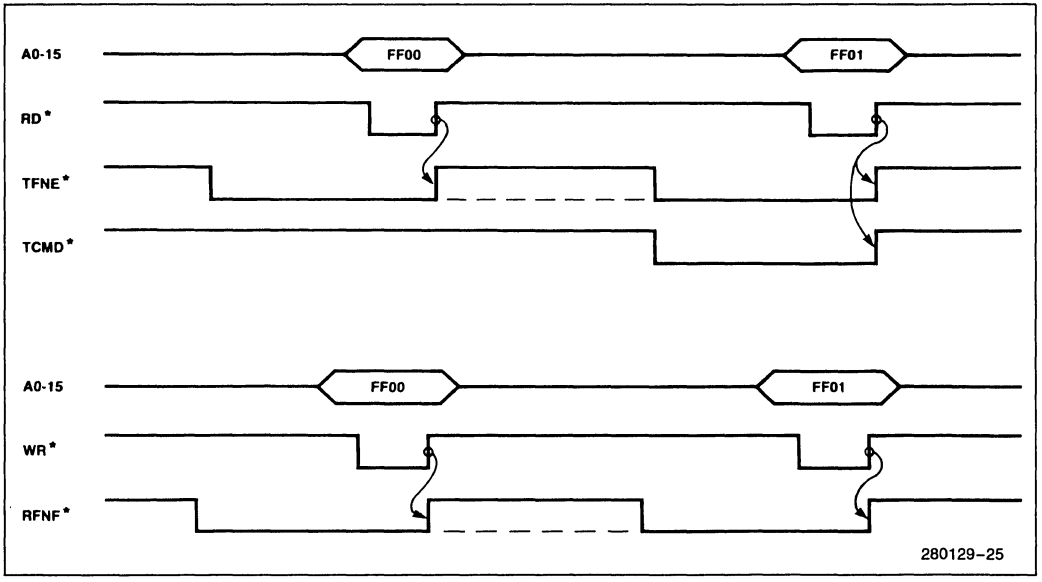
NOTES:
1. A.C. testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0".
2. Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".

EXTERNAL CLOCK DRIVE XTAL2



Symbol	Parameter	Variable Clock Freq = 3.5 MHz to 12 MHz		Units
		Min	Max	
TCLCL	Oscillator Period	83.3	285.7	ns
TCHCX	High Time	30	TCLCL−TCLCX	ns
TCLCX	Low Time	20	TCLCL−TCHCX	ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

BEM PARALLEL INTERFACE LOGIC TIMING



SPECIFICATIONS

Package: 40 pin DIP, 44 pin PLCC
Process: +5V, silicon gate HMOSII

Related Documents (Not Supplied)

Order Number

146312-001— Guide to Using the Distributed Control Modules

231663-002— 8044AH/8344AH/8744H Data Sheet

210941-002 — OEM System Handbook

210918-006 — Embedded Controller Handbook

231166-001 — VLSI Solutions for Distributed Control Applications

ORDERING INFORMATION

Part Number	Description
P,N8044AH,R 0112	BITBUS Enhanced Microcontroller

8044AH/8344AH/8744H
HIGH PERFORMANCE 8-BIT MICROCONTROLLER WITH ON-CHIP SERIAL COMMUNICATION CONTROLLER

- **8044AH—Includes Factory Mask Programmable ROM**
- **8344AH—For Use with External Program Memory**
- **8744H—Includes User Programmable/Eraseable EPROM**

8051 MICROCONTROLLER CORE

- **Optimized for Real Time Control 12 MHz Clock, Priority Interrupts, 32 Programmable I/O Lines, Two 16-bit Timer/Counters**
- **Boolean Processor**
- **4K × 8 ROM, 192 × 8 RAM**
- **64K Accessible External Program Memory**
- **64K Accessible External Data Memory**
- **4 μs Multiply and Divide**

SERIAL INTERFACE UNIT (SIU)

- **Serial Communication Processor that Operates Concurrently to CPU**
- **2.4 Mbps Maximum Data Rate**
- **375 Kbps using On-Chip Phase Locked Loop**
- **Communication Software in Silicon:**
  - **Complete Data Link Functions**
  - **Automatic Station Response**
- **Operates as an SDLC Primary or Secondary Station**

The RUPI-44 family integrates a high performance 8-bit Microcontroller, the Intel 8051 Core, with an Intelligent/high performance HDLC/SDLC serial communication controller, called the Serial Interface Unit (SIU). See Figure 1. This dual architecture allows complex control and high speed data communication functions to be realized cost effectively.

Specifically, the 8044's Microcontroller features: 4K byte On-Chip program memory space; 32 I/O lines; two 16-bit timer/event counters; a 5-source; 2-level interrupt structure; a full duplex serial channel; a Boolean processor; and on-chip oscillator and clock circuitry. Standard TTL and most byte-oriented MCS-80 and MCS-85 peripherals can be used for I/O amd memory expansion.

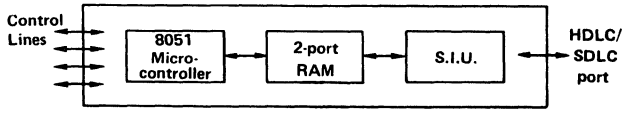
The Serial Interface Unit (SIU) manages the interface to a high speed serial link. The SIU offloads the On-Chip 8051 Microcontroller of communication tasks, thereby freeing the CPU to concentrate on real time control tasks.

The RUPI-44 family consists of the 8044, 8744, and 8344. All three devices are identical except in respect of on-chip program memory. The 8044 contains 4K bytes of mask-programmable ROM. User programmable EPROM replaces ROM in the 8744. The 8344 addresses all program memory externally.

The RUPI-44 devices are fabricated with Intel's reliable +5 volt, silicon-gate HMOSII technology and packaged in a 40-pin DIP.

The 8744H is available in a hermetically sealed, ceramic, 40-lead dual in-line package which includes a window that allows for EPROM erasure when exposed to ultraviolet light (See Erasure Characteristics). During normal operation, ambient light may adversely affect the functionality of the chip. Therefore applications which expose the 8744H to ambient light may require an opaque label over the window.

8044's Dual Controller Architecture



231663-1

Figure 1. Dual Controller Architecture

October 1987
Order Number: 231663-004

Table 1. RUPI™-44 Family Pin Description

VSS

Circuit ground potential.

VCC

+5V power supply during operation and program verification.

PORT 0

Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source eight LS TTL loads (six in 8744).

PORT 1

Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during program verification. Port 1 can sink/source four LS TTL loads.

In non-loop mode two of the I/O lines serve alternate functions:

— $\overline{RTS}$ (P1.6). Request-to-Send output. A low indicates that the RUPI-44 is ready to transmit.

— $\overline{CTS}$ (P1.7) Clear-to-Send input. A low indicates that a receiving station is ready to receive.

PORT 2

Port 2 is an 8-bit quasi-bidirection I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source four LS TTL loads.

PORT 3

Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and RD and WR pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source four LS LTT loads.

In addition to I/O, some of the pins also serve alternate functions as follows:

— I/$\overline{O}$ RxD (P3.0). In point-to-point or multipoint configurations, this pin controls the direction of pin P3.1. Serves as Receive Data input in loop and diagnostic modes.

— DATA TxD (P3.1) In point-to-point or multipoint configurations, this pin functions as data input/output. In loop mode, it serves as transmit pin. A '0' written to this pin enables diagnostic mode.

— $\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0.

— $\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1.

— TO (P3.4). Input to counter 0.

— SCLK T1 (P3.5). In addition to I/O, this pin provides input to counter 1 or serves as SCLK (serial clock) input.

— $\overline{WR}$ (P3.6). The write control signal latches the data byte from Port 0 into the External Data Memory.

— $\overline{RD}$ (P3.7). The read control signal enables External Data Memory to Port 0.

RST

A high on this pin for two machine cycles while the oscillator is running resets the device. A small external pulldown resistor ($\approx 8.2K\Omega$) from RST to V_{SS} permits power-on reset when a capacitor ($\approx 10\mu f$) is also connected from this pin to V_{CC}.

ALE/$\overline{PROG}$

Provides Address Latch Enable output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access. It also receives the program pulse input for programming the EPROM version.

$\overline{PSEN}$

The Program Store Enable output is a control signal that enables the external Program Memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.

$\overline{EA}$/VPP

When held at a TTL high level, the RUPI-44 executes instructions from the internal ROM when the PC is less than 4096. When held at a TTL low level, the RUPI-44 fetches all instructions from external Program Memory. The pin also receives the 21V EPROM programming supply voltage on the 8744.

Table 1. RUPI™-44 Family Pin Description (Continued)

XTAL 1	XTAL 2
Input to the oscillator's high gain amplifier. Required when a crystal is used. Connect to VSS when external source is used on XTAL 2.	Output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.

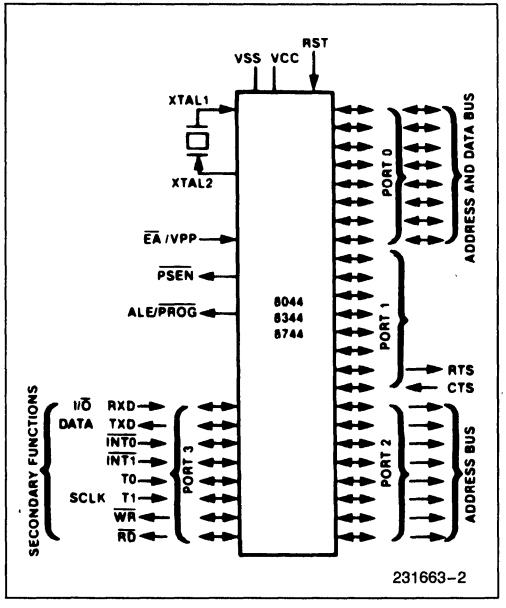


Figure 2. Logic Symbol

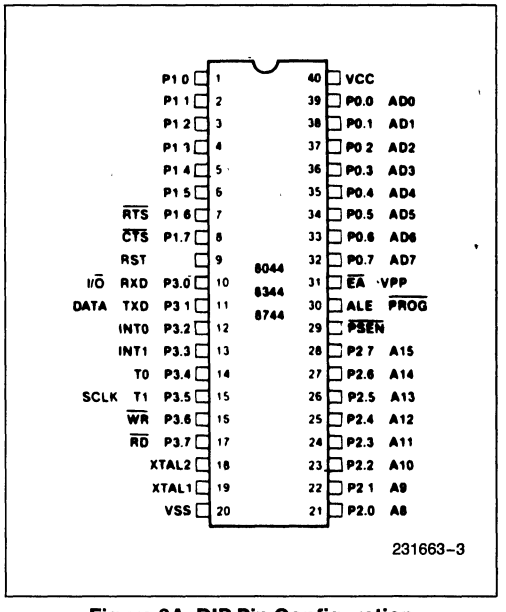


Figure 3A. DIP Pin Configuration

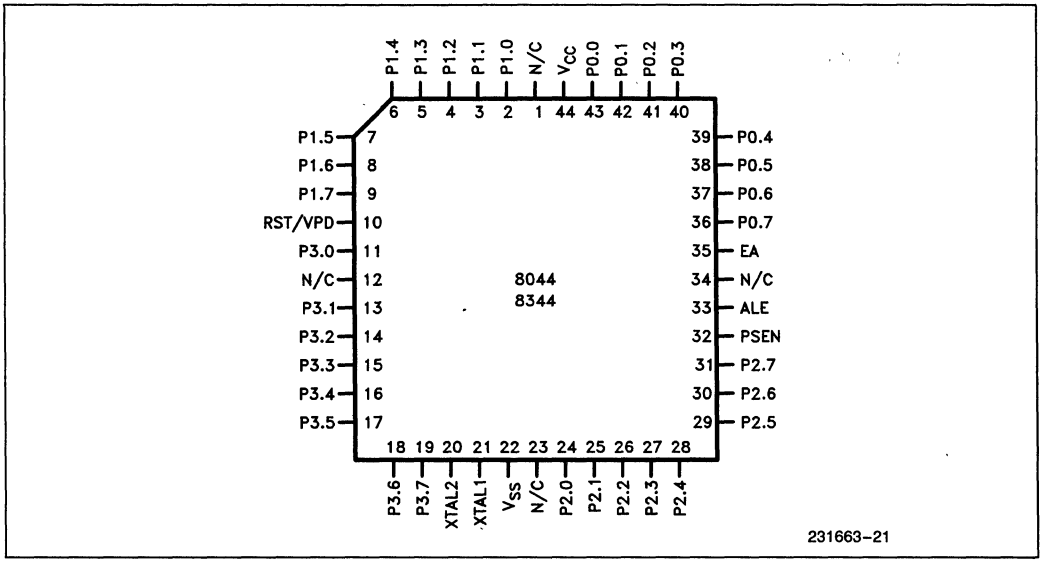


Figure 3B. PLCC Pin Configuration

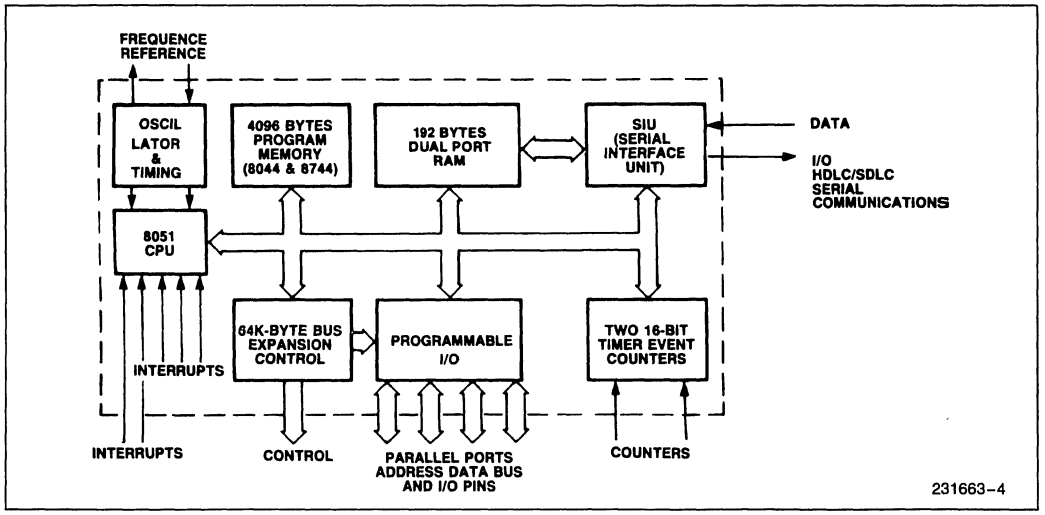


Figure 4. Block Diagram

FUNCTIONAL DESCRIPTION

General

The 8044 integrates the powerful 8051 microcontroller with an intelligent Serial Communication Controller to provide a single-chip solution which will efficiently implement a distributed processing or distributed control system. The microcontroller is a self-sufficient unit containing ROM, RAM, ALU, and its own peripherals. The 8044's architecture and instruction set are identical to the 8051's. The 8044 replaces the 8051's serial interface with an intelligent SDLC/HDLC Serial Interface Unit (SIU). 64 more bytes of RAM have been added to the 8051 RAM array. The SIU can communicate at bit rates up to 2.4 M bps. The SIU works concurrently with the Microcontroller so that there is no throughput loss in either unit. Since the SIU possesses its own intelligence, the CPU is off-loaded from many of the communications tasks, thus dedicating more of its computing power to controlling local peripherals or some external process.

The Microcontroller

The microcontroller is a stand-alone high-performance single-chip computer intended for use in sophisticated real-time application such as instrumentation, industrial control, and intelligent computer peripherals.

The major features of the microcontroller are:

- 8-bit CPU
- on-chip oscillator
- 4K bytes of ROM
- 192 bytes of RAM
- 32 I/O lines
- 64K address space for external Data Memory
- 64K address space for external Program Memory
- two fully programmable 16-bit timer/counters
- a five-source interrupt structure with two priority levels
- bit addressability for Boolean processing

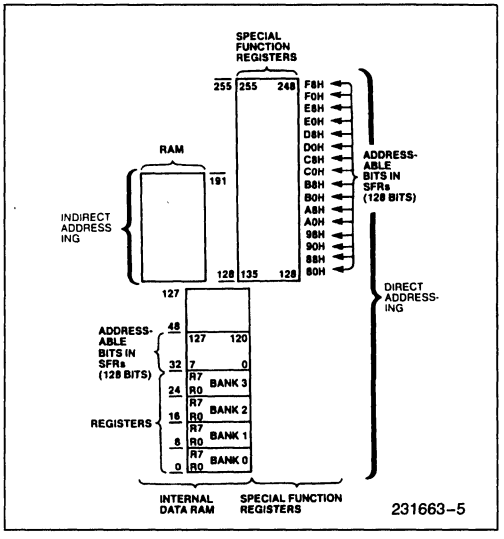


Figure 5. Internal Data Memory Address Space

- 1 μs instruction cycle time for 60% of the instructions 2 μs instruction cycle time for 40% of the instructions
- 4 μs cycle time for 8 by 8 bit unsigned Multiply/Divide

INTERNAL DATA MEMORY

Functionally the Internal Data Memory is the most flexible of the address spaces. The Internal Data Memory space is subdivided into a 256-byte Internal Data RAM address space and a 128-bit Special Function Register address space as shown in Figure 5.

The Internal Data RAM address space is 0 to 255. Four 8-Register Banks occupy locations 0 through 31. The stack can be located anywhere in the Internal Data RAM address space. In addition, 128 bit locations of the on-chip RAM are accessible through Direct Addressing. These bits reside in Internal Data RAM at byte locations 32 through 47. Currently locations 0 through 191 of the Internal Data RAM address space are filled with on-chip RAM.

Parallel I/O

The 8044 has 32 general-purpose I/O lines which are arranged into four groups of eight lines. Each group is called a port. Hence there are four ports; Port 0, Port 1, Port 2, and Port 3. Up to five lines from Port 3 are dedicated to supporting the serial channel when the SIU is invoked. Due to the nature of the serial port, two of Port 3's I/O lines (P3.0 and P3.1) do not have latched outputs. This is true whether or not the serial channel is used.

Port 0 and Port 2 also have an alternate dedicated function. When placed in the external access mode, Port 0 and Port 2 become the means by which the 8044 communicates with external program memory. Port 0 and Port 2 are also the means by which the 8044 communicates with external data memory. Peripherals can be memory mapped into the address space and controlled by the 8044.

Table 2. MCS®-51 Instruction Set Description

Mnemonic		Description	Byte	Cyc
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1

Table 2. MCS®-51 Instruction Set Description (Continued)

Mnemonic		Description	Byte	Cyc
LOGICAL OPERATIONS				
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@RI	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@RI	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@RI	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2

Table 2. MCS®-51 Instruction Set Description (Continued)

Mnemonic		Description	Byte	Cyc
DATA TRANSFER (Continued)				
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit) addr	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order Digit ind RAM w A	1	1
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry Flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry Flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C,/bit	AND complement of direct bit to Carry	2	2
ORL	C/bit	OR direct bit to Carry flag	2	2
ORL	C,/bit	OR complement of direct bit to Carry	2	2
MOV	C,/bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2
PROGRAM AND MACHINE CONTROL				
ACALL	addr11	Absolute Subroutine Call	2	2
LCALL	addr16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute Jump	2	2
LJMP	addr16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Comp, immed, to A & Jump if Not Equal	3	2

Table 2. MCS®-51 Instruction Set Description (Continued)

Mnemonic	Description	Byte	Cyc
PROGRAM AND MACHINE CONTROL (Continued)			
CJNE Rn,#data,rel	Comp, immed, to reg & Jump if Not Equal	3	2
CJNE @Ri,#data, rel	Comp, immed, to ind. & Jump if Not Equal	3	2
DJNZ Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ direct,rel	Decrement direct & Jump if Not Zero	3	2
NOP	No operation	1	1

Notes on data addressing modes:

Rn — Working register R0-R7

direct — 128 internal RAM locations, any I/O port, control or status register

@Ri — Indirect internal RAM location addressed by register R0 or R1

Notes on data addressing modes:

(Continued)

#data — 8-bit constant included in instruction

#data16 — 16-bit constant included as bytes 2 & 3 of instruction

bit — 128 software flags, any I/O pin, controll or status bit

Notes on program addressing modes:

addr16 — Destination address for LCALL & LJMP may be anywhere within the 64-K program memory address space

Addr11 — Destination address for ACALL & AJMP will be within the same 2-K page of program memory as the first byte of the following instruction

rel — SJMP and all conditional jumps include an 8-bit offset byte, Range is +127 −128 bytes relative to first byte of the following instruction

Timer/Counters

The 8044 contains two 16-bit counters which can be used for measuring time intervals, measuring pulse widths, counting events, generating precise periodic interrupt requests, and clocking the serial communications. Internally the Timers are clocked at 1/12 of the crystal frequency, which is the instruction cycle time. Externally the counters can run up to 500 KHz.

Interrupt System

External events and the real-time driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, a sophisticated multiple-source, two priority level, nested interrupt system is provided. Interrupt response latency ranges from 3 μsec to 7 μsec when using a 12 MHz clock.

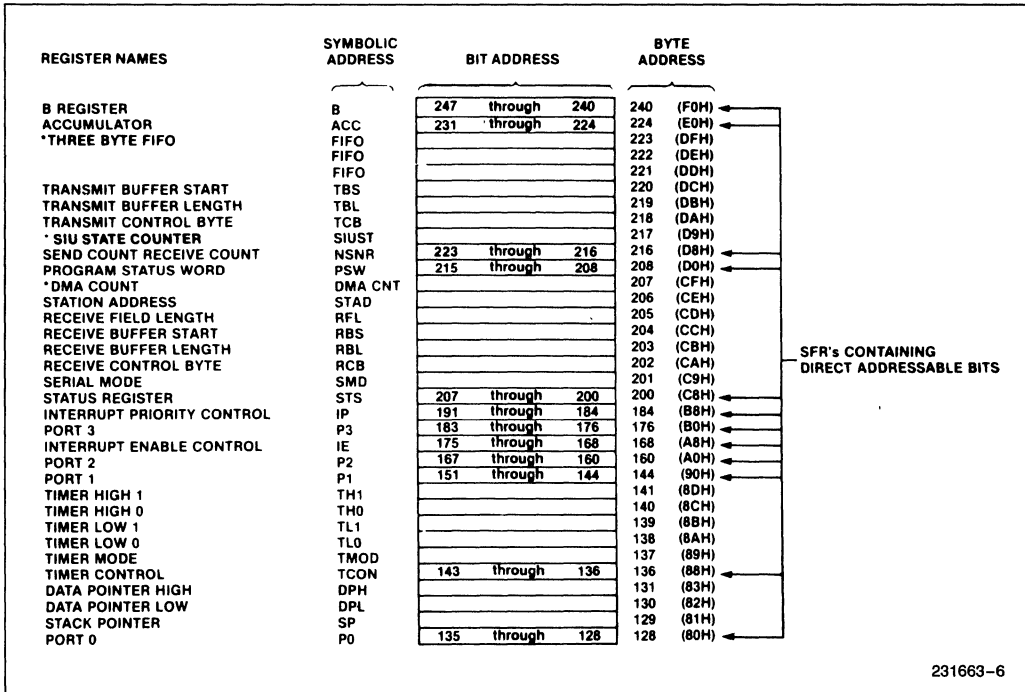
All five interrupt sources can be mapped into one of the two priority levels. Each interrupt source can be enabled or disabled individually or the entire interrupt system can be enabled or disabled. The five interrupt sources are: Serial Interface Unit, Timer 1, Timer 2, and two external interrupts. The external interrupts can be either level or edge triggered.

Serial Interface Unit (SIU)

The Serial Interface Unit is used for HDLC/SDLC communications. It handles Zero Bit Insertion/Deletion, Flags automatic access recognization, and a 16-bit cyclic redundancy check. In addition it implements in hardware a subset of the SDLC protocol certain applications it is advantageous to have the CPU control the reception or transmission of every single frame. For this reason the SIU has two modes of operation: "AUTO" and "FLEXIBLE" (or "NON-AUTO"). It is in the AUTO mode that the SIU responds to SDLC frames without CPU intervention; whereas, in the FLEXIBLE mode the reception or transmission of every single frame will be under CPU control.

There are three control registers and eight parameter registers that are used to operate the serial interface. These registers are shown in Figure 5 and Figure 6. The control register set the modes of operation and provide status information. The eight parameter registers buffer the station address, receive and transmit control bytes, and point to the on-chip transmit and receive buffers.

Data to be received or transmitted by the SIU must be buffered anywhere within the 192 bytes of on-chip RAM. Transmit and receive buffers are not allowed to "wrap around" in RAM; a "buffer end" is generated after address 191 is reached.



NOTE:
*ICE Support Hardware registers. Under normal operating conditions there is no need for the CPU to access these registers.

Figure 5. Mapping of Special Function Registers

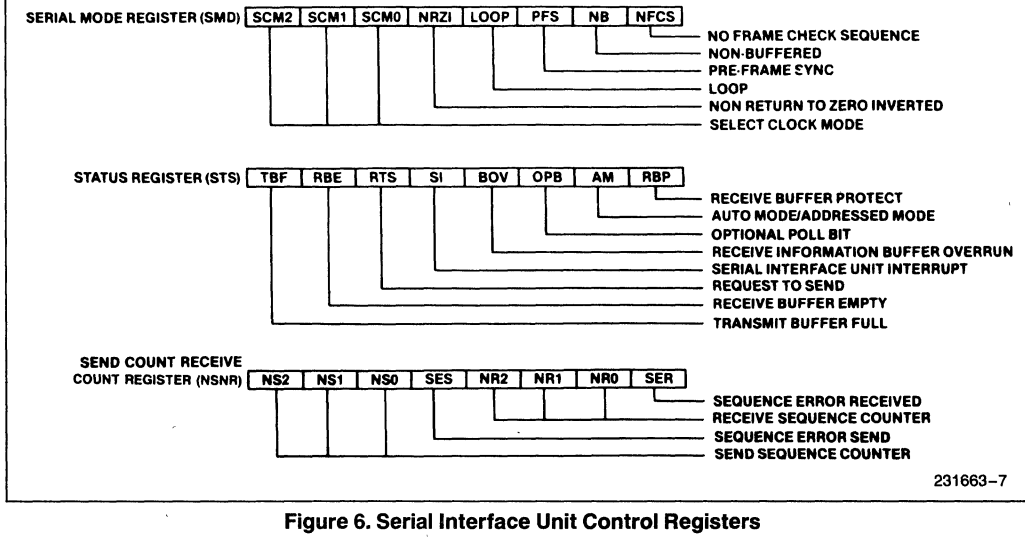


Figure 6. Serial Interface Unit Control Registers

With the addition of only a few bytes of code, the 8044's frame size is not limited to the size of its internal RAM (192 bytes), but rather by the size of external buffer with no degradation of the RUPI's features (e.g. NRZI, zero bit insertion/deletion, address recognition, cyclic redundancy check). There is a special function register called SIUST whose contents dictates the operation of the SIU. At low data rates, one section of the SIU (the Byte Processor) performs no function during known intervals. For a given data rate, these intervals (stand-by mode) are fixed. The above characteristics make it possible to program the CPU to move data to/from external RAM and to force the SIU to perform some desired hardware tasks while transmission or reception is taking place. With these modifications, external RAM can be utilized as a transmit and received buffer instead of the internal RAM.

AUTO Mode

In the AUTO mode the SIU implements in hardware a subset of the SDLC protocol such that it responds to many SDLC frames without CPU intervention. All AUTO mode responses to the primary station will comform to IBM's SDLC definition. The advantages of the AUTO mode are that less software is required to implement a secondary station, and the hardware generated response to polls is much faster than doing it in software. However, the Auto mode can not be used at a primary station.

To transmit in the AUTO mode the CPU must load the Transmit Information Buffer, Transmit Buffer Start register, Transmit Buffer Length register, and set the Transmit Buffer Full bit. The SIU automatically responds to a poll by transmitting an information frame with the P/F bit in the control field set. When the SIU receives a positive acknowledgement from the primary station, it automatically increments the Ns field in the NSNR register and interrupts the CPU. A negative acknowledgement would cause the SIU to retransmit the frame.

To receive in the AUTO mode, the CPU loads the Receive Buffer Start register, the Receive Buffer Length register, clears the Receive Buffer Protect bit, and sets the Receive Buffer Empty bit. If the SIU is polled in this state, and the TBF bit indicates that the Transmit Buffer is empty, an automatic RR response will be generated. When a valid information frame is received the SIU will automatically increment Nr in the NSNR register and interrupt the CPU.

While in the AUTO mode the SIU can recognize and respond to the following commands without CPU intervention: I (Information), RR (Receive Ready), RNR (Receive Not Ready), REJ (Reject), and UP (Unnumbered Poll). The SIU can generate the following responses without CPU intervention: I (Information), RR (Receive Ready), and RNR (Receive Not Ready).

When the Receive Buffer Empty bit (RBE) indicates that the Receive Buffer is empty, the receiver is enabled, and when the RBE bit indicates that the Receive Buffer is full, the receiver is disabled. Assuming that the Receiver Buffer is empty, the SIU will respond to a poll with an I frame if the Transmit Buffer is full. If the Transmit Buffer is empty, the SIU will respond to a poll with a RR command if the Receive Buffer Protect bit (RBP) is cleared, or an RNR command if RBP is set.

FLEXIBLE (or NON-AUTO) Mode

In the FLEXIBLE mode all communications are under control of the CPU. It is the CPU's task to encode and decode control fields, manage acknowledgements, and adhere to the requirements of the HDLC/SDLC protocols. The 8044 can be used as a primary or a secondary station in this mode.

To receive a frame in the FLEXIBLE mode, the CPU must load the Receive Buffer Start register, the Receive Buffer Length register, clear the Receive Buffer Protect bit, and set the Receive Buffer Empty bit. If a valid opening flag is received and the address field matches the byte in the Station Address register or the address field contains a broadcast address, the 8044 loads the control field in the receive control byte register, and loads the I field in the receive buffer. If there is no CRC error, the SIU interrupts the CPU, indicating a frame has just been received. If there is a CRC error, no interrupt occurs. The Receive Field Length register provides the number of bytes that were received in the information field.

To transmit a frame, the CPU must load the transmit information buffer, the Transmit Buffer Start register, the Transmit Buffer Length register, the Transmit Control Byte, and set the TBF and the RTS bit. The SIU, unsolicited by an HDLC/SDLC frame, will transmit the entire information frame, and interrupt the CPU, indicating the completion of transmission. For supervisory frames or unnumbered frames, the transmit buffer length would be 0.

CRC

The FCS register is initially set to all 1's prior to calculating the FCS field. The SIU will not interrupt the CPU if a CRC error occurs (in both AUTO and FLEXIBLE modes). The CRC error is cleared upon receiving of an opening flag.

Frame Format Options

In addition to the standard SDLC frame format, the 8044 will support the frames displayed in Figure 7. The standard SDLC frame is shown at the top of this figure. For the remaining frames the information field will incorporate the control or address bytes and the frame check sequences; therefore these fields will be stored in the Transmit and Receive buffers. For example, in the non-buffered mode the third byte is treated as the beginning of the information field. In the non-addressed mode, the information field begins after the opening flag. The mode bits to set the frame format options are found in the Serial Mode register and the Status register.

FRAME OPTION	NFCS	NB	AM[1]	FRAME FORMAT
Standard SDLC NON-AUTO Mode	0	0	0	F A C I FCS F
Standard SDLC AUTO Mode	0	0	1	F A C I FCS F
Non-Buffered Mode NON-AUTO Mode	0	1	1	F A I FCS F
Non-Addressed Mode NON-AUTO Mode	0	1	0	F I FCS F
No FCS Field NON-AUTO Mode	1	0	0	F A C I F
No FCS Field AUTO Mode	1	0	1	F A C I F
No FCS Field Non-Buffered Mode NON-AUTO Mode	1	1	1	F A I F
No FCS Field Non-Addressed Mode NON-AUTO Mode	1	1	0	F I F

Mode Bits:
AM — "AUTO" Mode/Addressed Mode
NB — Non-Buffered Mode
NFCS — No FCS Field Mode

Key to Abbreviations:
F = Flag (01111110)
A = Address Field
C = Control Field
I = Information Field
FCS= Frame Check Sequence

Note 1:
The AM bit function is controlled by the NB bit. When NB = 0, AM becomes AUTO mode select, when NB = 1, AM becomes Address mode select.

Figure 7. Frame Format Options

Extended Addressing

To realize an extended control field or an extended address field using the HDLC protocol, the FLEXIBLE mode must be used. For an extended control field, the SIU is programmed to be in the non-buffered mode. The extended control field will be the first and second bytes in the Receive and Transmit Buffers. For extended addressing the SIU is placed in the non-addressed mode. In this mode the CPU must implement the address recognition for received frames. The addressing field will be the initial bytes in the Transmit and Receive buffers followed by the control field.

The SIU can transmit and receive only frames which are multiples of 8 bits. For frames received with other than 8-bit multiples, a CRC error will cause the SIU to reject the frame.

SDLC Loop Networks

The SIU can be used in an SDLC loop as a secondary or primary station. When the SIU is placed in the Loop mode it receives the data on pin 10 and transmits the data one bit time delayed on pin 11. It can also recognize the Go ahead signal and change it into a flag when it is ready to transmit. As a secondary station the SIU can be used in the AUTO or FLEXIBLE modes. As a primary station the FLEXIBLE mode is used; however, additional hardware is required for generating the Go Ahead bit pattern. In the Loop mode the maximum data rate is 1 Mbps clocked or 375 Kpbs self-clocked.

SDLC Multidrop Networks

The SIU can be used in a SDLC non-loop configuration as a secondary or primary station. When the SIU is placed in the non-loop mode, data is received and transmitted on pin 11, and pin 10 drives a tri-state buffer. In non-loop mode, modem interface pins, RTS and CTS, become available.

Data Clocking Options

The 8044's serial port can operate in an externally clocked or self clocked system. A clocked system provides to the 8044 a clock synchronization to the data. A self-clocked system uses the 8044's on-chip Digital Phase Locked Loop (DPLL) to recover the clock from the data, and clock this data into the Serial Receive Shift Register.

In this mode, a clock synchronized with the data is externally fed into the 8044. This clock may be generated from an External Phase Locked Loop, or possibly supplied along with the data. The 8044 can transmit and receive data in this mode at rates up to 2.4 Mbps.

This self clocked mode allows data transfer without a common system data clock. An on-chip Digital Phase Locked Loop is employed to recover the data clock which is encoded in the data stream. The DPLL will converge to the nominal bit center within eight bit transitions, worst case. The DPLL requires a reference clock of either 16 times (16x) or 32 times (32x) the data rate. This reference clock may be externally applied or internally generated. When internally generated either the 8044's internal logic clock (crystal frequency divided by two) or the timer 1 overflow is used as the reference clock. Using the internal timer 1 clock the data rates can vary from 244 to 62.5 Kbps. Using the internal logic clock at a 16x sampling rate, receive data can either be 187.5 Kbps, or 375 Kbps. When the reference clock for the DPLL is externally applied the data rates can vary from 0 to 375 Kbps at a 16x sampling rate.

To aid in a Phase Locked Loop capture, the SIU has a NRZI (Non Return to Zero Inverted) data encoding and decoding option. Additionally the SIU has a preframe sync option that transmits two bytes of alternating 1's and 0's to ensure that the receive station DPLL will be synchronized with the data by the time it receives the opening flag.

Control and Status Registers

There are three SIU Control and Status Registers:

Serial Mode Register (SMD)

Status/Command Register (STS)

Send/Receive Count Register (NSNR)

The SMD, STS, and NSNR, registers are all cleared by system reset. This assures that the SIU will power up in an idle state (neither receiving nor transmitting).

These registers and their bit assignments are described below.

SMD: Serial Mode Register (byte-addressable)

Bit 7:	6	5	4	3	2	1	0
SCM2	SCM1	SCM0	NRZI	LOOP	PFS	NB	NFCS

The Serial Mode Register (Address C9H) selects the operational modes of the SIU. The 8044 CPU can both read and write SMD. The SIU can read SMD but cannot write to it. To prevent conflict between CPU and SIU access to SMD, the CPU should write SMD only when the Request To Send (RTS) and

Receive Buffer Empty (RBE) bits (in the STS register) are both false (0). Normally, SMD is accessed only during initialization.

The individual bits of the Serial Mode Register are as follows:

Bit#	Name	Description
SMD.0	NFCS	No FCS field in the SDLC frame.
SMD.1	NB	Non-Buffered mode. No control field in the SDLC frame.
SMD.2	PFS	Pre-Frame Sync mode. In this mode, the 8044 transmits two bytes before the first flag of a frame, for DPLL synchronization. If NRZI is enabled, 00H is sent; otherwise, 55H is sent. In either case, 16 preframe transitions are guaranteed.
SMD.3	LOOP	Loop configuration.
SMD.4	NRZI	NRZI coding option. If bit = 1, NRZI coding is used. If bit = 0, then it is straight binary (NRZ).
SMD.5	SCM0	Select Clock Mode—Bit 0
SMD.6	SCM1	Select Clock Mode—Bit 1
SMD.7	SCM2	Select Clock Mode—Bit 2

The SCM bits decode as follows:

SCM				Data Rate
2	1	0	Clock Mode	(Bits/sec)*
0	0	0	Externally clocked	0-2.4M**
0	0	1	Reserved	
0	1	0	Self clocked, timer overflow	244-62.5K
0	1	1	Reserved	
1	0	0	Self clocked, external 16x	0-375K
1	0	1	Self clocked, external 32x	0-187.5K
1	1	0	Self clocked, internal fixed	375K
1	1	1	Self clocked, internal fixed	187.5K

NOTES:
*Based on a 12 Mhz crystal frequency
**0-1 M bps in loop configuration

STS: Status/Command Register (bit-addressable)

Bit:	7	6	5	4	3	2	1	0
	TBF	RBE	RTS	SI	BOV	OPB	AM	RBP

The Status/Command Register (Address C8H) provides operational control of the SIU by the 8044 CPU, and enables the SIU to post status information for the CPU's access. The SIU can read STS, and can alter certain bits, as indicated below. The CPU can both read and write STS asynchronously. However, 2-cycle instructions that access STS during both cycles ('JBC/B, REL' and 'MOV/B, C.') should not be used, since the SIU may write to STS between the two CPU accesses.

The individual bits of the Status/Command Register are as follows:

Bit#	Name	Description
STS.0	RBP	Receive Buffer Protect. Inhibits writing of data into the receive buffer. In AUTO mode, RBP forces an RNR response instead of an RR.
STS.1	AM	AUTO Mode/Addressed Mode. Selects AUTO mode where AUTO mode is allowed. If NB is true, (= 1), the AM bit selects the addressed mode. AM may be cleared by the SIU.
STS.2	OPB	Optional Poll Bit. Determines whether the SIU will generate an AUTO response to an optional poll (UP with P = 0). OPM may be set or cleared by the SIU.
STS.3	BOV	Receive Buffer Overrun. BOV may be set or cleared by the SIU.
STS.4	SI	SIU Interrupt. This is one of the five interrupt sources to the CPU. The vector location = 23H. SI may be set by the SIU. It should be cleared by the CPU before returning from an interrupt routine.
STS.5	RTS	Request To Send. Indicates that the 8044 is ready to transmit or is transmitting. RTS may be read or written by the CPU. RTS may be read by the SIU, and in AUTO mode may be written by the SIU.
STS.6	RBE	Receive Buffer Empty. RBE can be thought of as Receive Enable. RBE is set to one by the CPU when it is ready to receive a frame, or has just read the buffer, and to zero by the SIU when a frame has been received.
STS.7	TBF	Transmit Buffer Full. Written by the CPU to indicate that it has filled the transmit buffer. TBF may be cleared by the SIU.

NSNR: Send/Receive Count Register (bit-addressable)

Bit:	7	6	5	4	3	2	1	0
	NS2	NS1	NS0	SES	NR2	NR1	NR0	SER

The Send/Receive Count Register (Address D8H) contains the transmit and receive sequence numbers, plus tally error indications. The SIU can both read and write NSNR. The 8044 CPU can both read and write NSNR asynchronously. However, 2-cycle instructions that access NSNR during both cycles ('JBC /B, REL,' and 'MOV /B,C') should not be used, since the SIU may write to NSMR between the two 8044 CPU accesses.

The individual bits of the Send/Receive Count Register are as follows:

Bit#	Name	Description
NSNR.0	SER	Receive Sequence Error: NS (P) ≠ NR (S)
NSNR.1	NR0	Receive Sequence Counter—Bit 0
NSNR.2	NR1	Receive Sequence Counter—Bit 1
NSNR.3	NR2	Receive Sequence Counter—Bit 2
NSNR.4	SES	Send Sequence Error: NR (P) ≠ NS (S) and NR (P) ≠ NS (S) + 1
NSNR.5	NS0	Send Sequence Counter—Bit 0
NSNR.6	NS1	Send Sequence Counter—Bit 1
NSNR.7	NS2	Send Sequence Counter—Bit 2

Parameter Registers

There are eight parameter registers that are used in connection with SIU operation. All eight registers may be read or written by the 8044 CPU. RFL and RCB are normally loaded by the SIU.

The eight parameter registers are as follows:

STAD: Station Address Register (byte-addressable)

The Station Address register (Address CEH) contains the station address. To prevent acess conflict, the CPU should access STAD only when the SIU is idle (RTS = 0 and RBE = 0). Normally, STAD is accessed only during initialization.

TBS: Transmit Buffer Start Address Register (byte-addressable)

The Transmit Buffer Start address register (Address DCH) points to the location in on-chip RAM for the beginning of the I-field of the frame to be transmitted. The CPU should access TBS only when the SIU is not transmitting a frame (when TBF = 0).

TBL: Transmit Buffer Length Register (byte = addressable)

The Transmit Buffer Length register (Address DBH) contains the length (in bytes) of the I-field to be transmitted. A blank I-field (TBL = 0) is valid. The CPU should access TBL only when the SIU is not transmitting a frame (when TBF = 0).

NOTE:
The transmit and receive buffers are not allowed to "wrap around" in the on-chip RAM. A "buffer end" is automatically generated if address 191 (BFH) is reached.

TCB: Transmit Control Byte Register (byte-addressable)

The Transmit Control Byte register (Address DAH) contains the byte which is to be placed in the control field of the transmitted frame, during NON-AUTO mode transmission. The CPU should access TCB only when the SIU is not transmitting a frame (when TBF = 0). The N_S and N_R counters are not used in the NON-AUTO mode.

RBS: Receive Buffer Start Address Register (byte-addressable)

The Receive Buffer Start address register (Address CCH) points to the location in on-chip RAM where the beginning of the I-field of the frame being received is to be stored. The CPU should write RBS only when the SIU is not receiving a frame (when RBE = 0).

RBL: Receive Buffer Length Register (byte-addressable)

The Receive Buffer Length register (Address CBH) contains the length (in bytes) of the area in on-chip RAM allocated for the received I-field. RBL = 0 is valid. The CPU should write RBL only when RBE = 0.

RFL: Receive Field Length Register (byte-addressable)

The Receive Field Length register (Address CDH) contains the length (in bytes) of the received I-field that has just been loaded into on-chip RAM. RFL is loaded by the SIU. RFL = 0 is valid. RFL should be accessed by the CPU only when RBE = 0.

RCB: Receive Control Byte Register (byte-addressable)

The Received Control Byte register (Address CAH) contains the control field of the frame that has just been received. RCB is loaded by the SIU. The CPU can only read RCB, and should only access RCB when RBE = 0.

ICE Support

The 8044 In-Circuit Emulator (ICE-44) allows the user to exercise the 8044 application system and monitor the execution of instructions in real time.

The emulator operates with Intel's Intellec™ development system. The development system interfaces with the user's 8044 system through an in-cable buffer box. The cable terminates in a 8044 pin-compatible plug, which fits into the 8044 socket in the user's system. With the emulator plug in place, the user can excercise his system in real time while collecting up to 255 instruction cycles of real-time data. In addition, he can single-step the program.

Static RAM is available (in the in-cable buffer box) to emulate the 8044 internal and external program memory and external data memory. The designer can display and alter the contents of the replacement memory in the buffer box, the internal data memory, and the internal 8044 registers, including the SFR's.

SIUST: SIU State Counter (byte-addressable)

The SIU State Counter (Address D9H) reflects the state of the internal logic which is under SIU control. Therefore, care must be taken not to write into this register. This register provides a useful means for debugging 8044 receiver problem.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature.....−65°C to −150°C
Voltage on $\overline{EA}$, VPP Pin to VSS...−0.5V to −21.5V
Voltage on Any Other Pin to VSS....−0.5V to −7V
Power Dissipation 2W

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, VCC = 5V ± 10%, VSS = 0V

Symbol	Parameter		Min	Max	Unit	Test Conditions
VIL	Input Low Voltage (Except $\overline{EA}$ Pin of 8744H)		−0.5	0.8	V	
VIL1	Input Low Voltage to $\overline{EA}$ Pin of 8744H		0	0.8	V	
VIH	Input High Voltage (Except XTAL2, RST)		2.0	VCC + 0.5	V	
VIH1	Input High Voltage to XTAL2, RST		2.5	VCC + 0.5	V	XTAL1 = VSS
VOL	Output Low Voltage (Ports 1, 2, 3)*			0.45	V	IOL = 1.6mA
VOL1	Output Low Voltage (Port 0,ALE,$\overline{PSEN}$)*					
		8744H		0.60 0.45	V V	IOL = 3.2 mA IOL = 2.4 mA
		8044AH/8344AH		0.45	V	IOL = 3.2 mA
VOH	Output High Voltage (Ports 1, 2, 3)		2.4		V	IOH = −80 μA
VOH1	Output High Voltage (Port 0 in External Bus Mode, ALE, $\overline{PSEN}$)		2.4		V	IOH = −400 μA
IIL	Logical 0 Input Current (Ports 1, 2, 3)			−500	μA	Vin = 0.45V
IIL1	Logical 0 Input Current to $\overline{EA}$ Pin of 8744H only			−15	mA	
IIL2	Logical 0 Input Current (XTAL2)			−3.6	mA	Vin = 0.45V
ILI	Input Leakage Current (Port 0) 8744H 8044AH/8344AH			 ±100 ±10	 μA μA	 0.45 < Vin < VCC 0.45 < Vin < VCC
IIH	Logical 1 Input Current to $\overline{EA}$ Pin of 8744H			500	μA	
IIH1	Input Current to RST to Activate Reset			500	μA	Vin < (VCC − 1.5V)
ICC	Power Supply Current: 8744H 8044AH/8344AH			 285 170	 mA mA	All Outputs Disconnected: $\overline{EA}$ = VCC
CIO	Pin Capacitance			10	pF	Test Freq. = 1MHz(1)

***NOTES:**
1. Sampled not 100% tested. T_A = 25°C.
2. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VOLs of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pin when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.

A.C. CHARACTERISTICS

T_A = 0°C to +70°C, VCC = 5V ± 10%, VSS = 0V, Load Capacitance for Port 0, ALE, and PSEN = 100 pF, Load Capacitance for All Other Outputs = 80 pF

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

Symbol	Parameter	12 MHz Osc		Variable Clock 1/TCLCL = 3.5 MHz to 12 MHz		Unit
		Min	Max	Min	Max	
TLHLL	ALE Pulse Width	127		2TCLCL-40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL-40		ns
TLLAX[1]	Address Hold After ALE Low	48		TCLCL-35		ns
TLLIV	ALE Low to Valid Instr in 8744H 8044AH/8344AH		 183 233		 4TCLCL-150 4TCLCL-100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	58		TCLCL-25		ns
TPLPH	$\overline{PSEN}$ Pulse Width 8744H 8044AH/8344AH	 190 215		 3TCLCL-60 3TCLCL-35		 ns ns
TPLIV	$\overline{PSEN}$ Low to Valid Instr in 8744H 8044AH/8344AH		 100 125		 3TCLCL-150 3TCLCL-125	 ns ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		0		ns
TPXIZ[2]	Input Instr Float After $\overline{PSEN}$		63		TCLCL-20	ns
TPXAV[2]	$\overline{PSEN}$ to Address Valid	75		TCLCL-8		ns
TAVIV	Address to Valid Instr in 8744H 8044AH/8344AH		 267 302		 5TCLCL-150 5TCLCL-115	 ns ns
TAZPL	Address Float to $\overline{PSEN}$	−25		−25		ns

NOTES:
1. TLLAX for access to program memory is different from TLLAX for data memory.
2. Interfacing RUPI-44 devices with float times up to 75ns is permissible. This limited bus contention will not cause any damage to Port 0 drivers.

EXTERNAL DATA MEMORY CHARACTERISTICS

Symbol	Parameter	12 MHz Osc		Variable Clock 1/TCLCL = 3.5 MHz to 12 MHz		Unit
		Min	Max	Min	Max	
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL-100		ns
TLLAX	Address Hold after ALE	48		TCLCL-35		ns
TRLDV	$\overline{RD}$ Low to Valid Data in		252		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		2TCLCL-70	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL-150	ns
TAVDV	Address to Valid Data In		585		9TCLCL-165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL-50	3TLCLCL+50	ns
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition					
	8744H	13		TCLCL-70		ns
	8044AH/8344AH	23		TCLCL-60		ns
TQVWH	Data Setup Before $\overline{WR}$ High	433		7TCLCL-150		ns
TWHQX	Data Held After $\overline{WR}$	33		TCLCL-50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		25		25	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High					
	8744H	33	133	TCLCL-50	TCLCL+50	ns
	8044AH/8344AH	43	123	TCLCL-40	TCLCL+50	ns

NOTE:
1. TLLAX for access to program memory is different from TLLAX for access data memory.

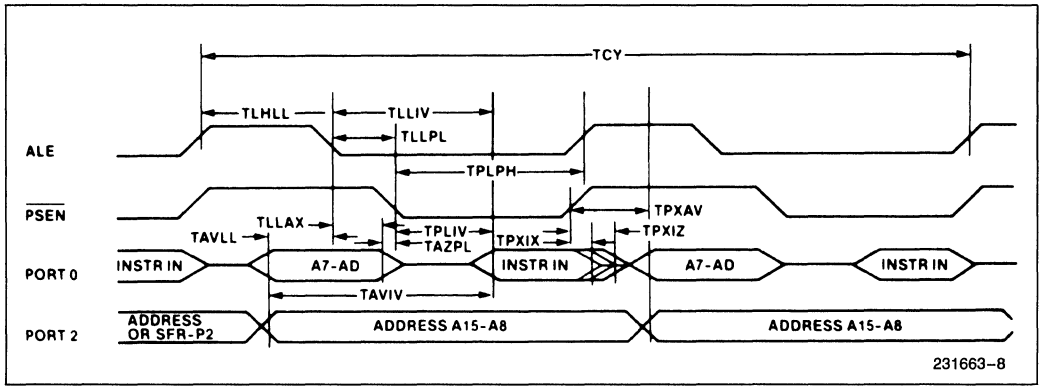
Serial Interface Characteristics

Symbol	Parameter	Min	Max	Unit
TDCY	Data Clock	420		ns
TDCL	Data Clock Low	180		ns
TDCH	Data Clock High	100		ns
tTD	Transmit Data Delay		140	ns
tDSS	Data Setup Time	40		ns
tDHS	Data Hold Time	40		ns

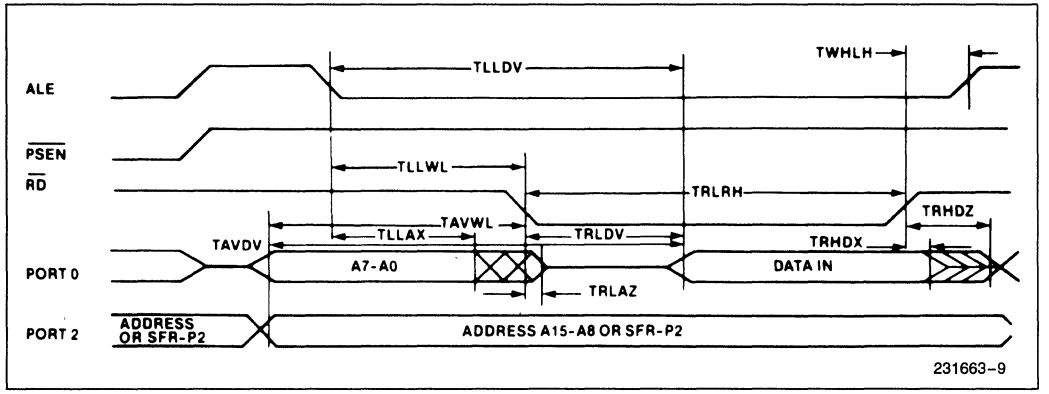
WAVEFORMS

Memory Access

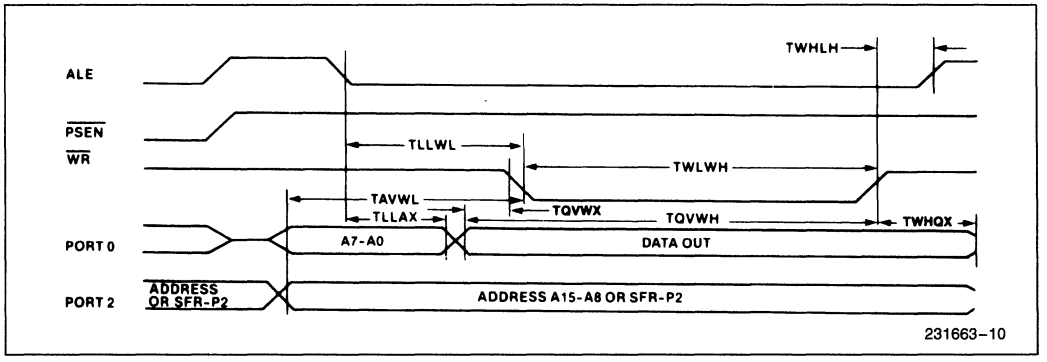
PROGRAM MEMORY READ CYCLE



DATA MEMORY READ CYCLE

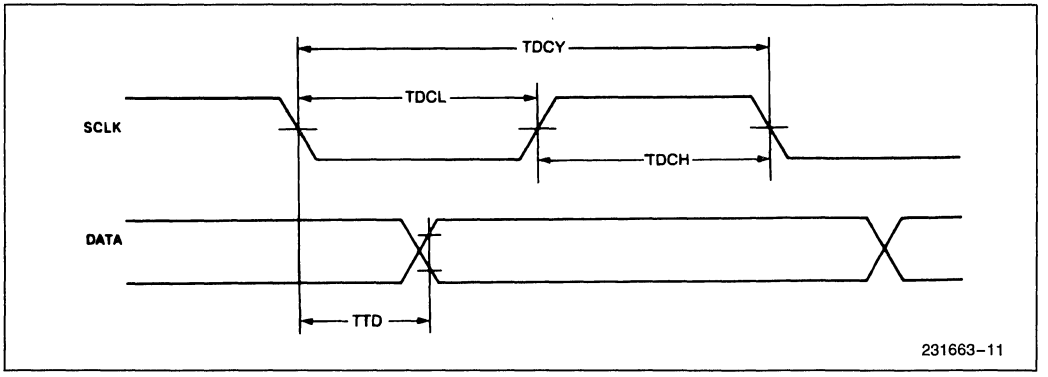


DATA MEMORY WRITE CYCLE

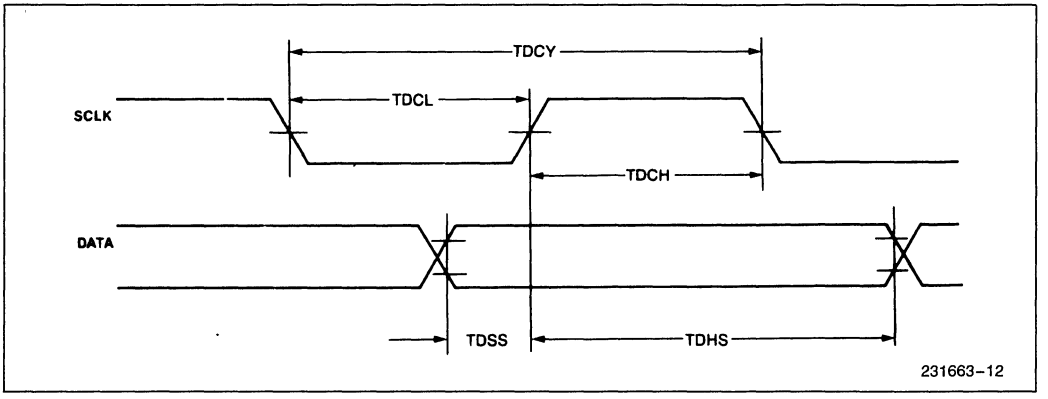


SERIAL I/O WAVEFORMS

SYNCHRONOUS DATA TRANSMISSION



SYNCHRONOUS DATA RECEPTION



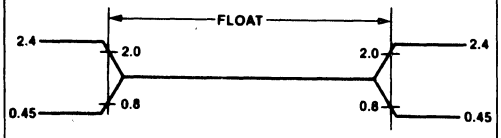
AC TESTING INPUT, OUTPUT, FLOAT WAVEFORMS

INPUT/OUTPUT

231663-13

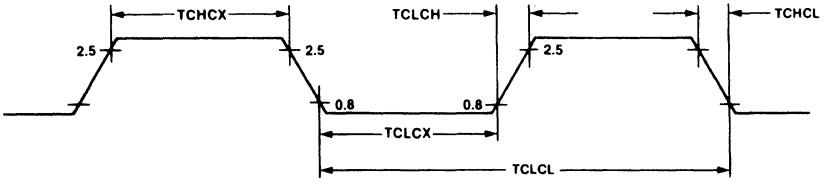
AC testing inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0" Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

FLOAT



231663-14

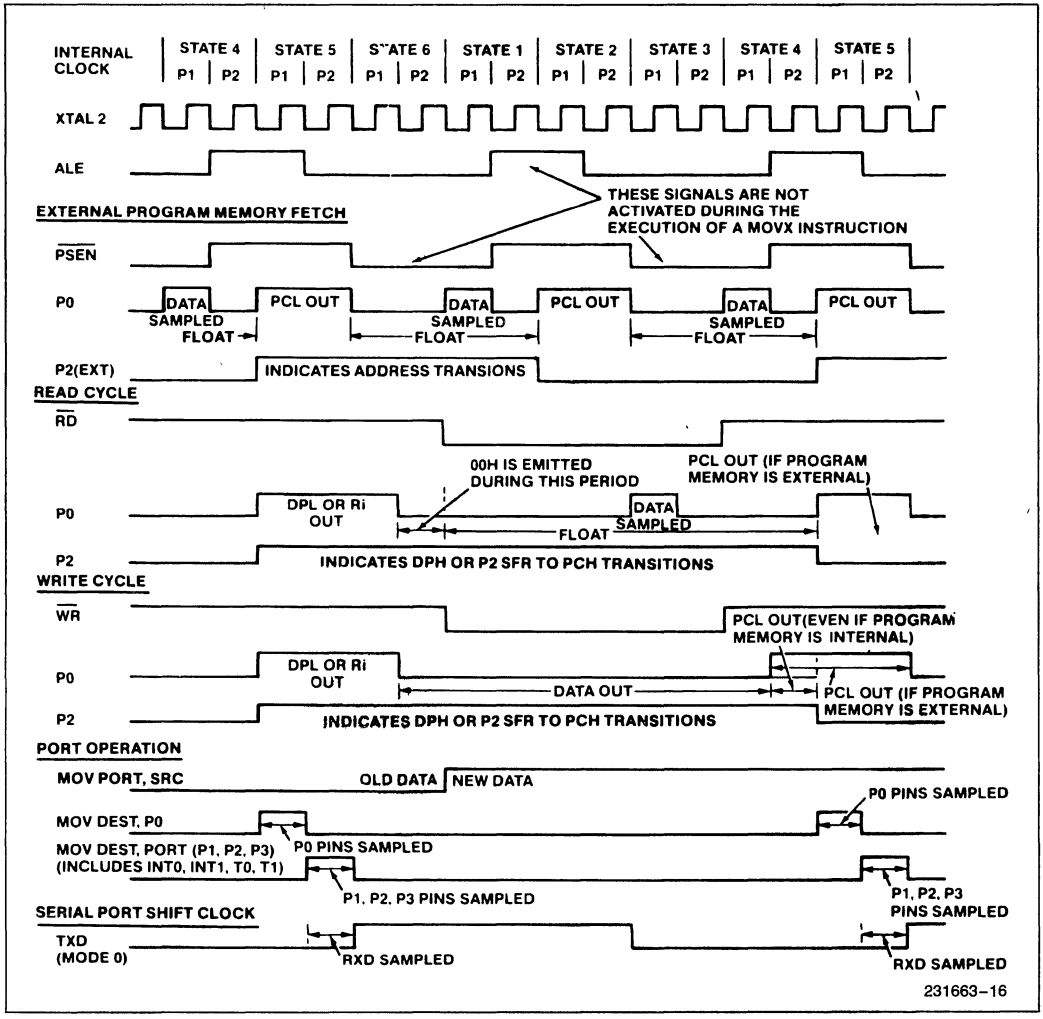
EXTERNAL CLOCK DRIVE XTAL2



231663-15

Symbol	Parameter	Variable Clock Freq = 3.5 MHz to 12 MHz		Unit
		Min	Max	
TCLCL	Oscillator Period	83.3	285.7	ns
TCHCX	High Time	20	TCLCL-TCLCX	ns
TCLCX	Low Time	20	TCLCL-TCHCX	ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component to component. Typically though, (T_A = 25°C, fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

8744H EPROM CHARACTERISTICS

Erasure Characteristics

Erasure of the 8744H Program Memory begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 Ångstroms. Since sunlight and fluorescent lighting have wavelengths in this range, constant exposure to these light sources over an extended period of time (about 1 week in sunlight, or 3 years in room-level fluorescent lighting) could cause unintentional erasure. If an application subjects the 8744H to this type of exposure, it is suggested that an opaque label be placed over the window.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 Ångstroms) to an integrated dose of at least 15 W-sec/cm^2 rating for 20 to 30 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1s state.

Programming the EPROM

To be programmed, the 8744H must be running with a 4 to 6 MHz oscillator. (The reason the oscillator needs to be running is that the internal bus is being used to transfer address and program data to appropriate registers.) The address of an EPROM location to be programmed is applied to Port 1 and pins P2.0-P2.3 of Port 2, while the data byte is applied to Port 0. Pins P2.4-P2.6 and $\overline{PSEN}$ should be held low, and P2.7 and RST high. (These are all TTL levels except RST, which requires 2.5V for high.) $\overline{EA}$/VPP is held normally high, and is pulsed to +21V. While EA/VPP is at 21V, the ALE/$\overline{PROG}$ pin, which is normally being held high, is pulsed low for 50 msec. Then EA/VPP is returned to high. This is illustrated in Figure 8. Detailed timing specifications are provided in the EPROM Programming and Verification Characteristics section of this data sheet.

Program Memory Security

The program memory security feature is developed around a "security bit" in the 8744H EPROM array. Once this "hidden bit" is programmed, electrical access to the contents of the entire program memory array becomes impossible. Activation of this feature is accomplished by programming the 8744H as described in "Programming the EPROM" with the exception that P2.6 is held at a TTL high rather than a TTL low. In addition, Port 1 and P2.0-P2.3 may be in any state. Figure 9 illustrates the security bit programming configuration. Deactivating the security feature, which again allows programmability of the EPROM, is accomplished by exposing the EPROM to ultraviolet light. This exposure, as described in "Erasure Characteristics," erases the entire EPROM array. Therefore, attempted retrieval of "protected code" results in its destruction.

Program Verification

Program Memory may be read only when the "security feature" has not been activated. Refer to Figure 10 for Program Verification setup. To read the Program Memory, the following procedure can be used. The unit must be running with a 4 to 6 MHz oscillator. The address of a Program Memory location to be read is applied to Port 1 and pins P2.0-P2.3 of Port 2. Pins P2.4-P2.6 and $\overline{PSEN}$ are held at TTL low, while the ALE/$\overline{PROG}$, RST, and $\overline{EA}$/VPP pins are held at TTL high. (These are all TTL levels except RST, which requires 2.5V for high.) Port 0 will be the data output lines. P2.7 can be used as a read strobe. While P2.7 is held high, the Port 0 pins float. When P2.7 is strobed low, the contents of the addressed location will appear at Port 0. External pullups (e.g., 10K) are required on Port 0 during program verification.

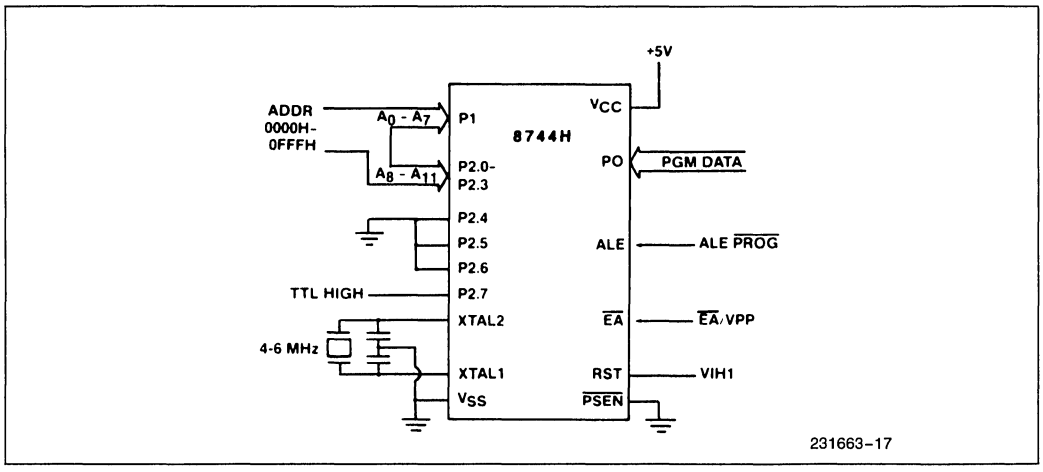


Figure 8. Programming Configuration

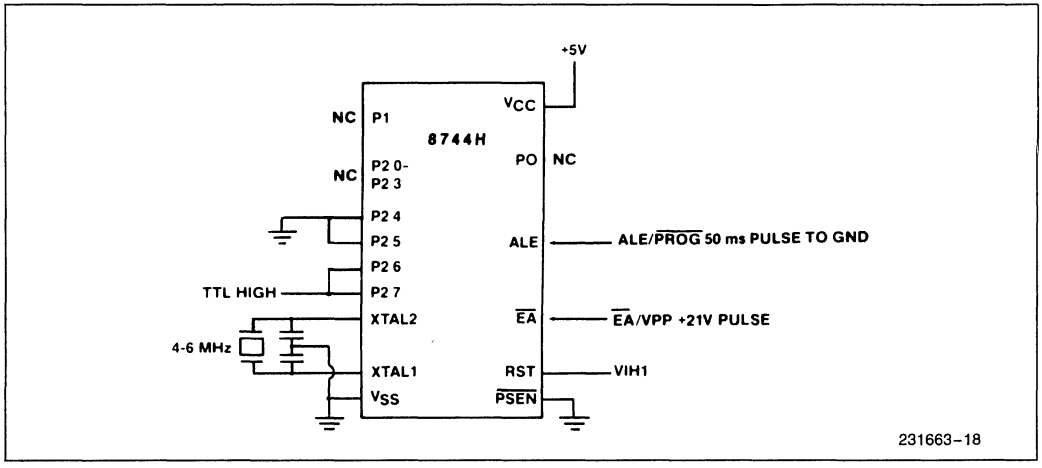


Figure 9. Security Bit Programming Configuration

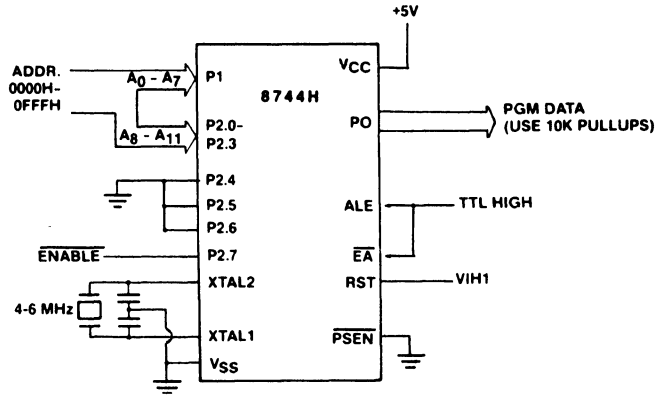


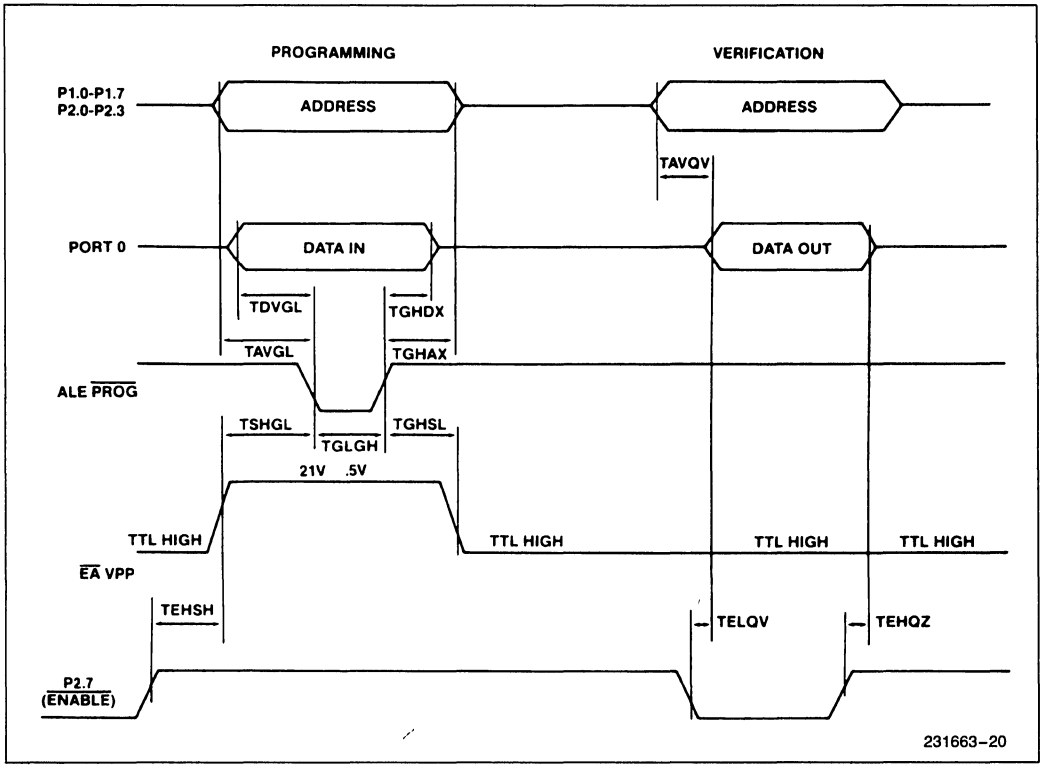
Figure 10. Program Verification Configuration

EPROM PROGRAMMING, SECURITY BIT PROGRAMMING AND VERIFICATION CHARACTERISTICS

TA = 21°C to 27°C, V_{CC} = 4.5V to 5.5V, V_{SS} = 0V

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	20.5	21.5	V
IPP	Programming Current		30	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Setup to $\overline{PROG}$	48TCLCL		
TGHAX	Address Hold after $\overline{PROG}$	48TCLCL		
TDVGL	Data Setup to $\overline{PROG}$	48TCLCL		
TGHDX	Data Hold after $\overline{PROG}$	48TCLCL		
TEHSH	$\overline{ENABLE}$ High to Vpp	48TCLCL		
TSHGL	V_{PP} Setup to $\overline{PROG}$	10		µsec
TGHSL	V_{PP} Hold after $\overline{PROG}$	10		µsec
TGLGH	$\overline{PROG}$ Width	45	55	msec
TAVQV	Address to Data Valid		48TCLCL	
TELQV	$\overline{ENABLE}$ to Data Valid		48TCLCL	
TEHQZ	Data Float after $\overline{ENABLE}$	0	48TCLCL	

EPROM PROGRAMMING, SECURITY BIT PROGRAMMING AND VERIFICATION WAVEFORMS



Service and Support 18

intel®

iRUG DESCRIPTION

iRUG is the Intel iRMX® User's Group. It is an incorporation chartered to establish a forum for users of the iRMX operating system and to promote and encourage development of iRMX-based software.

iRUG membership is for any person, firm, or corporation who has purchased or is authorized to use licensed iRMX software products from Intel Corporation or an OEM. Benefits of membership include: access to the user's library of iRMX software tools and utilities; membership in local and international chapters; access to the group bulletin board; receipt of quarterly international magazines; opportunity to present papers and conduct workshops; invitations to seminars devoted to the use of Intel products.

The user's library, maintained by iRUG, contains software programs written and submitted by members and Intel employees. Programs available range from file or directory manipulation commands and terminal attribute selection utilities to dynamic logon, background job facilities and basic communication utilities.

Local and international iRUG chapters provide a forum for members to meet other iRMX operating system users in an informal setting. At local meetings and the annual international seminar, members can discuss their ideas, share their experiences and techniques, and give feedback to Intel for future improvements and features of the iRMX operating system. The meetings also showcase new products offered by Intel and other developments in iRMX-based software supplied by other companies.

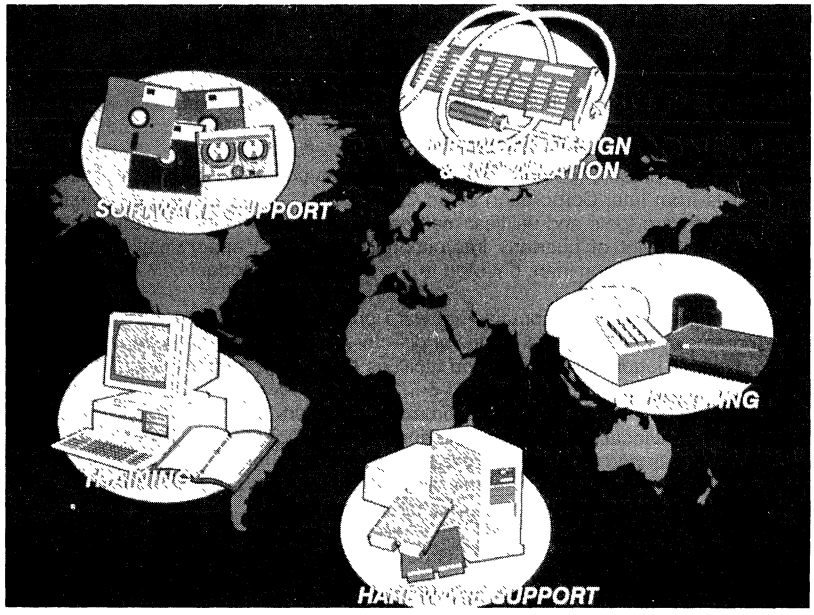
The "Intel Forum" sponsors iRUG Special Interest Group (SIG) on the CompuServe Information Service. The message facility (bulletin board) allows members to leave and receive messages from other members. These might include problems and solutions regarding the iRMX operating system or new techniques to be shared.

"The Human Interface" is iRUG's quarterly international magazine. It serves as a supplement to chapter meetings by providing: library listings, information on the latest releases of products running on the iRMX operating system; officer messages; member SPRs; vendor ads; release and update plans for the iRMX operating system; and technical articles from the members.

For Information Contact:

Janet Huston
iRUG Coordinator
5200 N.E. Elam Young Parkway
Mailstop HF3-23
Hillsboro, OR 97124
(503) 696-5078

280678-1



REAL-TIME SUPPORT FROM REAL-TIME EXPERTS

Intel's real-time service and support offerings include everything you need to keep your Intel equipment—and yourself—in peak performance at all times. Intel's hands-on training gets you up to speed quickly to reduce your development time. Our consulting services, software support, and networking assistance help you solve your design and networking problems efficiently. Your equipment will be up and running night and day with one of Intel's flexible repair programs. Hardware, software, tools, boards, or systems . If Intel's name is on it, Intel stands behind it.

WORLDWIDE CUSTOMER SERVICES

- Over one hundred Intel service and training centers worldwide
- Hardware maintenance programs with prompt, worldwide response
- Software support that includes updates, telephone hotline support, troubleshooting guides and *;COMMENTS*, a technical newsletter
- Consulting by Intel System Engineering Services for both hardware and software design solutions
- Hands-on training workshops in Intel systems hardware, operating systems, languages, tools and networking products
- Network design, installation and maintenance

September, 1989
Order Number 4610178 001 Rev C

CONVENIENT, FLEXIBLE HARDWARE MAINTENANCE OPTIONS

Intel offers a wide variety of hardware support programs, from 24-hour on-site protection to economical carry-in repair. Intel can help keep your systems running with the greatest economy and convenience. Consider all the options:

- **On-site coverage.** An Intel customer engineer will repair your equipment at your site during normal business hours.
- **Extended coverage.** Companies with critical applications choose this program, which offers on-site repair up to 24 hours a day, seven days a week.
- **Carry-in service.** An economical option for small system users that saves you transportation costs.
- **Mail-in service.** If time is not at issue, you can save even more by mailing in your part to an Intel Repair Center.
- **Per-call service.** If your service needs are infrequent, this option gives you access to Intel expertise on a time and materials basis.

COMPREHENSIVE SOFTWARE SUPPORT

Everything you need to stay up to date on the latest improvements to your Intel software is available through Intel's software support services: Technical Information Phone Service (TIPS), software updates, Troubleshooting Guides, and *;COMMENTS*, a technical newsletter

- **Technical Information Phone Service (TIPS).** TIPS puts you in touch instantly with Intel's software engineering staff to solve critical problems quickly.
- **Software updates.** As Intel enhances its software, it keeps you up to date with the latest updates and documentation.
- **Troubleshooting Guides.** Intel Troubleshooting Guides provide technical information on problem workarounds and solutions.
- ***;COMMENTS.*** We also publish *;COMMENTS*, a monthly newsletter full of programming techniques, technical product information, and application solutions

SYSTEMS ENGINEERING CONSULTING SERVICES

Intel's Systems Engineers have the experience and engineering expertise with Intel boards, systems, software, and tools, so they can save you valuable development time. They can work at your site or from a local Intel office, helping you get your application to market sooner, boost your productivity and ensure compatibility with future Intel projects.

Intel Systems Engineers can work with you to:

- Guide your development effort
- Provide solutions to your problems
- Help you utilize new products more effectively
- Create your new applications
- Provide you with personalized training
- Customize an Intel product to fit your needs

With an Intel Systems Engineer on your team you receive service on a flexible basis—available when and where you need it!

HANDS-ON TRAINING THAT REDUCES DEVELOPMENT TIME

Intel offers in-depth training workshops on all its hardware and software products, from microcontrollers to operating systems. Customers tell us over and over that one to ten days spent with Intel saves weeks of self-study—that's weeks off your development schedule.

We routinely deliver workshops at Intel training centers all over the world, or we can come to your site. We'll even customize our standard courses to better fit your unique application and design needs.

Intel workshops offered continuously around the world include:

- **Operating systems, languages, and tools.** Learn the fundamentals and advanced points of programming with iRMX®, iRMK™, XENIX®, PL/M, and UNIX® System V Operating Systems, or developing applications for Intel microprocessors using Intel development tools.
- **Local area networking.** Technical workshops on Intel and non-Intel networking products.
- **End-user workshops.** Introductory and advanced instruction on office automation, system administration and network administration.
- **Microprocessor and microcontroller workshops.** Learn to program the 8080, 8086, 386™, i486™, microprocessors and 80860 and 80960 microcontrollers for your next real-time application.

TOTAL COVERAGE OF YOUR NETWORK NEEDS

Intel has the answer to all your networking needs. We provide assistance in network design (both physical and logical), component selection, installation, system administrator and user training, on-site network management, and maintenance.

We specialize in networking multiple departments, buildings, and sites. We can help you understand the various network topologies and protocols, establish mainframe connections, automate your factory, and implement portable networks.

INTEL CUSTOMER SERVICES: WORLDWIDE, FLEXIBLE, AND ACCESSIBLE

Intel has structured its systems support, service, and training programs in such a way that we're flexible and accessible enough to match your unique needs. Wherever in the world you are. And we mean wherever you are. Intel has more than one hundred service and training centers in thirty countries on six continents. Wherever you are, Intel speaks your language: service.

Let Intel build a support program specifically suited for your needs. Call or write us today for more information: In the USA call: 1-800-INTEL-4-U. Worldwide: refer to the sales office list on the back of this document.

DOMESTIC SALES OFFICES

ALABAMA

†Intel Corp.
5015 Bradford Dr., #2
Huntsville 35805
Tel: (205) 830-4010
FAX: (205) 837-2640

ARIZONA

†Intel Corp.
11225 N. 28th Dr.
Suite D-214
Phoenix 85029
Tel: (602) 869-4980
FAX: (602) 869-4294

Intel Corp.
1161 N. El Dorado Place
Suite 301
Tucson 85715
Tel: (602) 299-6815
FAX: (602) 296-8234

CALIFORNIA

†Intel Corp.
21515 Vanowen Street
Suite 116
Canoga Park 91303
Tel: (818) 704-8500
FAX: (818) 340-1144

†Intel Corp.
2250 E. Imperial Highway
Suite 218
El Segundo 90245
Tel: (213) 640-6040
FAX: (213) 640-7133

Intel Corp
1510 Arden Way
Suite 101
Sacramento 95815
Tel: (916) 920-8096
FAX: (916) 920-8253

†Intel Corp
9665 Chesapeake Dr.
Suite 325
San Diego 95123
Tel: (619) 292-8086
FAX: (619) 292-0628

†Intel Corp.*
400 N. Tustin Avenue
Suite 450
Santa Ana 92705
Tel: (714) 835-9642
TWX: 910-595-1114
FAX: (714) 541-9157

†Intel Corp *
San Tomas 4
2700 San Tomas Expressway
2nd Floor
Santa Clara 95051
Tel: (408) 986-8086
TWX: 910-338-0255
FAX: (408) 727-2620

COLORADO

Intel Corp.
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel: (719) 594-6622
FAX: (303) 594-0720

†Intel Corp.*
650 S Cherry St.
Suite 915
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289
FAX. (303) 322-8670

CONNECTICUT

†Intel Corp.
301 Lee Farm Corporate Park
83 Wooster Heights Rd
Danbury 06810
Tel: (203) 748-3130
FAX: (203) 794-0339

FLORIDA

†Intel Corp.
6363 N.W 6th Way
Suite 100
Ft Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407
FAX. (305) 772-8193

†Intel Corp.
5850 T.G. Lee Blvd.
Suite 340
Orlando 32822
Tel: (407) 240-8000
FAX (407) 240-8097

Intel Corp
11300 4th Street North
Suite 170
St. Petersburg 33716
Tel: (813) 577-2413
FAX: (813) 578-1607

GEORGIA

Intel Corp.
20 Technology Parkway, N.W
Suite 150
Norcross 30092
Tel. (404) 449-0541
FAX. (404) 605-9762

ILLINOIS

†Intel Corp *
300 N. Martingale Road
Suite 400
Schaumburg 60173
Tel: (312) 605-8031
FAX. (312) 706-9762

INDIANA

†Intel Corp.
8777 Purdue Road
Suite 125
Indianapolis 46268
Tel: (317) 875-0623
FAX: (317) 875-8938

IOWA

Intel Corp.
1930 St. Andrews Drive N.E.
2nd Floor
Cedar Rapids 52402
Tel: (319) 393-1294

KANSAS

†Intel Corp
10985 Cody St.
Suite 140, Bldg. D
Overland Park 66210
Tel: (913) 345-2727
FAX: (913) 345-2076

MARYLAND

†Intel Corp.*
10010 Junction Dr.
Suite 200
Annapolis Junction 20701
Tel: (301) 206-2860
FAX: (301) 206-3677
(301) 206-3678

MASSACHUSETTS

†Intel Corp.*
Westford Corp. Center
3 Carlisle Road
2nd Floor
Westford 01886
Tel: (508) 692-3222
TWX: 710-343-6333
FAX: (508) 692-7867

MICHIGAN

†Intel Corp.
7071 Orchard Lake Road
Suite 100
West Bloomfield 48322
Tel: (313) 851-8096
FAX: (313) 851-8770

MINNESOTA

†Intel Corp.
3500 W. 80th St.
Suite 360
Bloomington 55431
Tel: (612) 835-6722
TWX: 910-576-2867
FAX: (612) 831-6497

MISSOURI

†Intel Corp.
4203 Earth City Expressway
Suite 131
Earth City 63045
Tel: (314) 291-1990
FAX: (314) 291-4341

NEW JERSEY

†Intel Corp.*
Parkway 109 Office Center
328 Newman Springs Road
Red Bank 07701
Tel: (201) 747-2233
FAX: (201) 747-0983

†Intel Corp
280 Corporate Center
75 Livingston Avenue
First Floor
Roseland 07068
Tel: (201) 740-0111
FAX. (201) 740-0626

NEW YORK

Intel Corp.*
850 Cross Keys Office Park
Fairport 14450
Tel: (716) 425-2750
TWX 510-253-7391
FAX: (716) 223-2561

†Intel Corp.*
2950 Expressway Dr., South
Suite 130
Islandia 11722
Tel. (516) 231-3300
TWX: 510-227-6236
FAX: (516) 348-7939

†Intel Corp.
Westage Business Center
Bldg. 300, Route 9
Fishkill 12524
Tel: (914) 897-3860
FAX: (914) 897-3125

NORTH CAROLINA

†Intel Corp.
5800 Executive Center Dr.
Suite 105
Charlotte 28212
Tel: (704) 568-8966
FAX: (704) 535-2236

Intel Corp.
5540 Centerview Dr.
Suite 215
Raleigh 27606
Tel: (919) 851-9537
FAX: (919) 851-8974

OHIO

†Intel Corp.*
3401 Park Center Drive
Suite 220
Dayton 45414
Tel: (513) 890-5350
TWX: 810-450-2528
FAX: (513) 890-8658

†Intel Corp.*
25700 Science Park Dr
Suite 100
Beachwood 44122
Tel: (216) 464-2736
TWX: 810-427-9298
FAX: (804) 282-0673

OKLAHOMA

Intel Corp.
6801 N. Broadway
Suite 115
Oklahoma City 73162
Tel: (405) 848-8086
FAX: (405) 840-9819

OREGON

†Intel Corp.
15254 N.W. Greenbrier Parkway
Building B
Beaverton 97005
Tel: (503) 645-8051
TWX: 910-467-8741
FAX: (503) 645-8181

PENNSYLVANIA

†Intel Corp.*
455 Pennsylvania Avenue
Suite 230
Fort Washington 19034
Tel: (215) 641-1000
TWX: 510-661-2077
FAX (215) 641-0785

†Intel Corp.*
400 Penn Center Blvd.
Suite 610
Pittsburgh 15235
Tel: (412) 823-4970
FAX: (412) 829-7578

PUERTO RICO

†Intel Corp.
South Industrial Park
P.O. Box 910
Las Piedras 00671
Tel: (809) 733-8616

TEXAS

Intel Corp.
8911 Capital of Texas Hwy
Austin 78759
Tel: (512) 794-8086
FAX: (512) 338-9335

†Intel Corp.*
12000 Ford Road
Suite 400
Dallas 75234
Tel. (214) 241-8087
FAX. (214) 484-1180

†Intel Corp.*
7322 S W Freeway
Suite 1490
Houston 77074
Tel: (713) 988-8086
TWX: 910-881-2490
FAX: (713) 988-3660

UTAH

†Intel Corp.
428 East 6400 South
Suite 104
Murray 84107
Tel: (801) 263-8051
FAX. (801) 268-1457

VIRGINIA

†Intel Corp.
1504 Santa Rosa Road
Suite 108
Richmond 23288
Tel: (804) 282-5668
FAX: (216) 464-2270

WASHINGTON

†Intel Corp.
155 108th Avenue N.E
Suite 386
Bellevue 98004
Tel: (206) 453-8086
TWX. 910-443-3002
FAX: (206) 451-9556

Intel Corp.
408 N. Mullan Road
Suite 102
Spokane 99206
Tel (509) 928-8086
FAX: (509) 928-9467

WISCONSIN

Intel Corp.
330 S. Executive Dr.
Suite 102
Brookfield 53005
Tel (414) 784-8087
FAX: (414) 796-2115

CANADA

BRITISH COLUMBIA

Intel Semiconductor of
Canada, Ltd.
4585 Canada Way
Suite 202
Burnaby V5G 4L6
Tel: (604) 298-0387
FAX: (604) 298-8234

ONTARIO

†Intel Semiconductor of
Canada, Ltd.
2650 Queensview Drive
Suite 250
Ottawa K2B 8H6
Tel: (613) 829-9714
FAX (613) 820-5936

†Intel Semiconductor of
Canada, Ltd.
190 Attwell Drive
Suite 500
Rexdale M9W 6H8
Tel: (416) 675-2105
FAX: (416) 675-2438

QUEBEC

Intel Semiconductor of
Canada, Ltd.
620 St. Jean Boulevard
Pointe Claire H9R 3K2
Tel. (514) 694-9130
FAX: 514-694-0064

†Sales and Service Office
*Field Application Location

DOMESTIC DISTRIBUTORS

ALABAMA

Arrow Electronics, Inc.
1015 Henderson Road
Huntsville 35805
Tel: (205) 837-6955

†Hamilton/Avnet Electronics
4940 Research Drive
Huntsville 35805
Tel: (205) 837-7210
TWX: 810-726-2162

Pioneer/Technologies Group, Inc.
4825 University Square
Huntsville 35805
Tel: (205) 837-9300
TWX. 810-726-2197

ARIZONA

†Hamilton/Avnet Electronics
505 S. Madison Drive
Tempe 85281
Tel: (602) 231-5140
TWX: 910-950-0077

Hamilton/Avnet Electronics
30 South McKiemy
Chandler 85226
Tel: (602) 961-6669
TWX. 910-950-0077

Arrow Electronics, Inc
4134 E. Wood Street
Phoenix 85040
Tel. (602) 437-0750
TWX: 910-951-1550

Wyle Distribution Group
17855 N Black Canyon Hwy.
Phoenix 85023
Tel: (602) 249-2232
TWX: 910-951-4282

CALIFORNIA

Arrow Electronics, Inc.
10824 Hope Street
Cypress 90630
Tel: (714) 220-6300

Arrow Electronics, Inc.
19748 Dearborn Street
Chatsworth 91311
Tel: (213) 701-7500
TWX 910-493-2086

†Arow Electronics, Inc
521 Weddell Drive
Sunnyvale 94086
Tel: (408) 745-6600
TWX: 910-339-9371

Arrow Electronics, Inc
9511 Ridgehaven Court
San Diego 92123
Tel (619) 565-4800
TWX: 888-064

†Arrow Electronics, Inc.
2961 Dow Avenue
Tustin 92680
Tel: (714) 838-5422
TWX. 910-595-2860

†Avnet Electronics
350 McCormick Avenue
Costa Mesa 92626
Tel (714) 754-6071
TWX 910-595-1928

†Hamilton/Avnet Electronics
1175 Bordeaux Drive
Sunnyvale 94086
Tel: (408) 743-3300
TWX· 910-339-9332

†Hamilton/Avnet Electronics
4545 Ridgeview Avenue
San Diego 92123
Tel· (619) 571-7500
TWX 910-595-2638

†Hamilton/Avnet Electronics
9650 Desoto Avenue
Chatsworth 91311
Tel: (818) 700-1161

†Hamilton Electro Sales
10950 W. Washington Blvd.
Culver City 20230
Tel: (213) 558-2458
TWX: 910-340-6364

Hamilton Electro Sales
1361B West 190th Street
Gardena 90248
Tel. (213) 217-6700

†Hamilton/Avnet Electronics
3002 'G' Street
Ontario 91761
Tel. (714) 989-9411

†Avnet Electronics
20501 Plummer
Chatsworth 91351
Tel: (213) 700-6271
TWX. 910-494-2207

†Hamilton Electro Sales
3170 Pullman Street
Costa Mesa 92626
Tel: (714) 641-4150
TWX: 910-595-2638

†Hamilton/Avnet Electronics
4103 Northgate Blvd.
Sacramento 95834
Tel: (916) 920-3150

Wyle Distribution Group
124 Maryland Street
El Segundo 90254
Tel. (213) 322-8100

Wyle Distribution Group
7382 Lampson Ave.
Garden Grove 92641·
Tel· (714) 891-1717
TWX: 910-348-7140 or 7111

Wyle Distribution Group
11151 Sun Center Drive
Rancho Cordova 95670
Tel: (916) 638-5282

†Wyle Distribution Group
9525 Chesapeake Drive
San Diego 92123
Tel: (619) 565-9171
TWX: 910-335-1590

†Wyle Distribution Group
3000 Bowers Avenue
Santa Clara 95051
Tel: (408) 727-2500
TWX 910-338-0296

†Wyle Distribution Group
17872 Cowan Avenue
Irvine 92714
Tel: (714) 863-9953
TWX: 910-595-1572

Wyle Distribution Group
26677 W. Agoura Rd
Calabasas 91302
Tel: (818) 880-9000
TWX: 372-0232

COLORADO

Arrow Electronics, Inc
7060 South Tucson Way
Englewood 80112
Tel: (303) 790-4444

†Hamilton/Avnet Electronics
8765 E Orchard Road
Suite 708
Englewood 80111
Tel: (303) 740-1017
TWX. 910-935-0787

†Wyle Distribution Group
451 E 124th Avenue
Thornton 80241
Tel (303) 457-9953
TWX 910-936-0770

CONNECTICUT

†Arrow Electronics, Inc.
12 Beaumont Road
Wallingford 06492
Tel· (203) 265-7741
TWX: 710-476-0162

Hamilton/Avnet Electronics
Commerce Industrial Park
Commerce Drive
Danbury 06810
Tel: (203) 797-2800
TWX: 710-456-9974

†Pioneer Electronics
112 Main Street
Norwalk 06851
Tel: (203) 853-1515
TWX: 710-468-3373

FLORIDA

†Arrow Electronics, Inc.
400 Fairway Drive
Suite 102
Deerfield Beach 33441
Tel· (305) 429-8200
TWX: 510-955-9456

Arrow Electronics, Inc.
37 Skyline Drive
Suite 3101
Lake Marv 32746
Tel: (407) 323-0252
TWX: 510-959-6337

†Hamilton/Avnet Electronics
6801 N.W. 15th Way
Ft. Lauderdale 33309
Tel· (305) 971-2900
TWX: 510-956-3097

†Hamilton/Avnet Electronics
3197 Tech Drive North
St. Petersburg 33702
Tel: (813) 576-3930
TWX· 810-863-0374

†Hamilton/Avnet Electronics
6947 University Boulevard
Winter Park 32792
Tel. (305) 628-3888
TWX: 810-853-0322

†Pioneer/Technologies Group, Inc.
337 S Lake Blvd.
Alta Monte Springs 32701
Tel· (407) 834-9090
TWX. 810-853-0284

Pioneer/Technologies Group, Inc
674 S. Military Trail
Deerfield Beach 33442
Tel: (305) 428-8877
TWX· 510-955-9653

GEORGIA

†Arrow Electronics, Inc.
3155 Northwoods Parkway
Suite A
Norcross 30071
Tel (404) 449-8252
TWX. 810-766-0439

†Hamilton/Avnet Electronics
5825 D Peachtree Corners
Norcross 30092
Tel: (404) 447-7500
TWX 810-766-0432

Pioneer/Technologies Group, Inc
3100 F Northwoods Place
Norcross 30071
Tel. (404) 448-1711
TWX: 810-766-4515

ILLINOIS

Arrow Electronics, Inc.
1140 W. Thorndale
Itasca 60143
Tel: (312) 250-0500
TWX· 312-250-0916

†Hamilton/Avnet Electronics
1130 Thorndale Avenue
Bensenville 60106
Tel: (312) 860-7780
TWX· 910-227-0060

MTI Systems Sales
1100 W. Thorndale
Itasca 60143
Tel: (312) 773-2300

†Pioneer Electronics
1551 Carmen Drive
Elk Grove Village 60007
Tel: (312) 437-9680
TWX: 910-222-1834

INDIANA

†Arrow Electronics, Inc
2495 Directors Row, Suite H
Indianapolis 46241
Tel: (317) 243-9353
TWX. 810-341-3119

Hamilton/Avnet Electronics
485 Gradle Drive
Carmel 46032
Tel. (317) 844-9333
TWX: 810-260-3966

†Pioneer Electronics
6408 Castleplace Drive
Indianapolis 46250
Tel: (317) 849-7300
TWX: 810-260-1794

IOWA

Hamilton/Avnet Electronics
915 33rd Avenue, S.W.
Cedar Rapids 52404
Tel: (319) 362-4757

KANSAS

Arrow Electronics
8208 Melrose Dr., Suite 210
Lenexa 66214
Tel: (913) 541-9542

†Hamilton/Avnet Electronics
9219 Quivera Road
Overland Park 66215
Tel: (913) 888-8900
TWX· 910-743-0005

Pioneer/Tec Gr.
10551 Lockman Rd.
Lenexa 66215
Tel: (913) 492-0500

KENTUCKY

Hamilton/Avnet Electronics
1051 D. Newton Park
Lexington 40511
Tel. (606) 259-1475

MARYLAND

Arrow Electronics, Inc.
8300 Guilford Drive
Suite H, River Center
Columbia 21046
Tel: (301) 995-0003
TWX: 710-236-9005

Hamilton/Avnet Electronics
6822 Oak Hall Lane
Columbia 21045
Tel· (301) 995-3500·
TWX: 710-862-1861

†Mesa Technology Corp
9720 Patuxent Woods Dr.
Columbia 21046
Tel. (301) 290-8150
TWX· 710-828-9702

†Pioneer/Technologies Group, Inc.
9100 Gaither Road
Gaithersburg 20877
Tel: (301) 921-0660
TWX: 710-828-0545

Arrow Electronics, Inc.
7524 Standish Place
Rockville 20855
Tel: 301-424-0244

MASSACHUSETTS

Arrow Electronics, Inc.
25 Upton Dr.
Wilmington 01887
Tel: (617) 935-5134

†Hamilton/Avnet Electronics
10D Centennial Drive
Peabody 01960
Tel: (617) 531-7430
TWX· 710-393-0382

MTI Systems Sales
83 Cambridge St.
Burlington 01813

Pioneer Electronics
44 Hartwell Avenue
Lexington 02173
Tel: (617) 861-9200
TWX: 710-326-6617

MICHIGAN

Arrow Electronics, Inc.
755 Phoenix Drive
Ann Arbor 48104
Tel· (313) 971-8220
TWX 810-223-6020

Hamilton/Avnet Electronics
2215 29th Street S.E.
Space A5
Grand Rapids 49508
Tel. (616) 243-8805
TWX: 810-274-6921

Pioneer Electronics
4504 Broadmoor S.E.
Grand Rapids 49508
FAX: 616-698-1831

†Hamilton/Avnet Electronics
32487 Schoolcraft Road
Livonia 48150
Tel: (313) 522-4700
TWX: 810-282-8775

†Pioneer/Michigan
13485 Stamford
Livonia 48150
Tel. (313) 525-1800
TWX: 810-242-3271

MINNESOTA

†Arrow Electronics, Inc
5230 W 73rd Street
Edina 55435
Tel· (612) 830-1800
TWX. 910-576-3125

†Hamilton/Avnet Electronics
12400 Whitewater Drive
Minnetonka 55434
Tel: (612) 932-0600

†Pioneer Electronics
7625 Golden Triange Dr.
Suite G
Eden Prairi 55343
Tel: (612) 944-3355

MISSOURI

†Arrow Electronics, Inc.
2380 Schuetz
St. Louis 63141
Tel: (314) 567-6888
TWX. 910-764-0882

†Hamilton/Avnet Electronics
13743 Shoreline Court
Earth City 63045
Tel. (314) 344-1200
TWX: 910-762-0684

NEW HAMPSHIRE

†Arrow Electronics, Inc.
3 Perimeter Road
Manchester 03103
Tel· (603) 668-6968
TWX· 710-220-1684

†Hamilton/Avnet Electronics
444 E Industrial Drive
Manchester 03103
Tel (603) 624-9400

†Microcomputer System Technical Distributor Center

DOMESTIC DISTRIBUTORS (Contd.)

NEW JERSEY

†Arrow Electronics, Inc
Four East Stow Road
Unit 11
Marlton 08053
Tel (609) 596-6000
TWX: 710-897-0829

†Arrow Electronics
6 Century Drive
Parsipanny 07054
Tel (201) 538-0900

†Hamilton/Avnet Electronics
1 Keystone Ave., Bldg. 36
Cherry Hill 08003
Tel: (609) 424-0110
TWX: 710-940-0262

†Hamilton/Avnet Electronics
10 Industrial
Fairfield 07006
Tel: (201) 575-5300
TWX: 710-734-4388

†MTI Systems Sales
37 Kullick Rd.
Fairfield 07006
Tel. (201) 227-5552

†Pioneer Electronics
45 Route 46
Pinebrook 07058
Tel. (201) 575-3510
TWX: 710-734-4382

NEW MEXICO

Alliance Electronics Inc
11030 Cochiti S E
Albuquerque 87123
Tel. (505) 292-3360
TWX: 910-989-1151

Hamilton/Avnet Electronics
2524 Baylor Drive S E.
Albuquerque 87106
Tel: (505) 765-1500
TWX: 910-989-0614

NEW YORK

†Arrow Electronics, Inc.
3375 Brighton Henrietta
Townline Rd
Rochester 14623
Tel. (716) 275-0300
TWX 510-253-4766

Arrow Electronics, Inc
20 Oser Avenue
Hauppauge 11788
Tel (516) 231-1000
TWX. 510-227-6623

Hamilton/Avnet
933 Motor Parkway
Hauppauge 11788
Tel. (516) 231-9800
TWX: 510-224-6166

†Hamilton/Avnet Electronics
333 Metro Park
Rochester 14623
Tel: (716) 475-9130
TWX: 510-253-5470

†Hamilton/Avnet Electronics
103 Twin Oaks Drive
Syracuse 13206
Tel: (315) 437-0288
TWX: 710-541-1560

†MTI Systems Sales
38 Harbor Park Drive
Port Washington 11050
Tel: (516) 621-6200

†Pioneer Electronics
68 Corporate Drive
Binghamton 13904
Tel (607) 722-9300
TWX· 510-252-0893

Pioneer Electronics
40 Oser Avenue
Hauppauge 11787
Tel: (516) 231-9200

†Pioneer Electronics
60 Crossway Park West
Woodbury, Long Island 11797
Tel. (516) 921-8700
TWX 510-221-2184

†Pioneer Electronics
840 Fairport Park
Fairport 14450
Tel: (716) 381-7070
TWX· 510-253-7001

NORTH CAROLINA

†Arrow Electronics, Inc
5240 Greensdairy Road
Raleigh 27604
Tel (919) 876-3132
TWX· 510-928-1856

†Hamilton/Avnet Electronics
3510 Spring Forest Drive
Raleigh 27604
Tel· (919) 878-0819
TWX. 510-928-1836

Pioneer/Technologies Group, Inc
9801 A-Southern Pine Blvd
Charlotte 28210
Tel. (919) 527-8188
TWX. 810-621-0366

OHIO

Arrow Electronics, Inc
7620 McEwen Road
Centerville 45459
Tel (513) 435-5563
TWX· 810-459-1611

†Arrow Electronics, Inc
6238 Cochran Road
Solon 44139
Tel (216) 248-3990
TWX 810-427-9409

†Hamilton/Avnet Electronics
954 Senate Drive
Dayton 45459
Tel· (513) 439-6733
TWX 810-450-2531

Hamilton/Avnet Electronics
4588 Emery Industrial Pkwy
Warrensville Heights 44128
Tel (216) 349-5100
TWX. 810-427-9452

†Hamilton/Avnet Electronics
777 Brooksedge Blvd
Westerville 43081
Tel· (614) 882-7004

†Pioneer Electronics
4433 Interpoint Boulevard
Dayton 45424
Tel. (513) 236-9900
TWX: 810-459-1622

†Pioneer Electronics
4800 E. 131st Street
Cleveland 44105
Tel· (216) 587-3600
TWX 810-422-2211

OKLAHOMA

Arrow Electronics, Inc.
1211 E. 51st St , Suite 101
Tulsa 74146
Tel. (918) 252-7537

†Hamilton/Avnet Electronics
12121 E 51st St , Suite 102A
Tulsa 74146
Tel (918) 252-7297

OREGON

†Almac Electronics Corp
1885 N W 169th Place
Beaverton 97005
Tel· (503) 629-8090
TWX· 910-467-8746

†Hamilton/Avnet Electronics
6024 S W. Jean Road
Bldg C, Suite 10
Lake Oswego 97034
Tel: (503) 635-7848
TWX· 910-455-8179

Wyle Distribution Group
5250 N E Elam Young Parkway
Suite 600
Hillsboro 97124
Tel (503) 640-6000
TWX: 910-460-2203

PENNSYLVANIA

Arrow Electronics, Inc
650 Seco Road
Monroeville 15146
Tel (412) 856-7000

Hamilton/Avnet Electronics
2800 Liberty Ave
Pittsburgh 15238
Tel (412) 281-4150

Pioneer Electronics
259 Kappa Drive
Pittsburgh 15238
Tel (412) 782-2300
TWX 710-795-3122

†Pioneer/Technologies Group, Inc
Delaware Valley
261 Gibralter Road
Horsham 19044
Tel (215) 674-4000
TWX 510-665-6778

TEXAS

†Arrow Electronics, Inc
3220 Commander Drive
Carrollton 75006
Tel. (214) 380-6464
TWX· 910-860-5377

†Arrow Electronics, Inc
10899 Kinghurst
Suite 100
Houston 77099
Tel· (713) 530-4700
TWX 910-880-4439

†Arrow Electronics, Inc
2227 W Braker Lane
Austin 78758
Tel (512) 835-4180
TWX 910-874-1348

†Hamilton/Avnet Electronics
1807 W Braker Lane
Austin 78758
Tel (512) 837-8911
TWX 910-874-1319

†Hamilton/Avnet Electronics
2111 W. Walnut Hill Lane
Irving 75038
Tel: (214) 550-6111
TWX· 910-860-5929

†Hamilton/Avnet Electronics
4850 Wright Rd , Suite 190
Stafford 77477
Tel (713) 240-7733
TWX· 910-881-5523

†Pioneer Electronics
18260 Kramer
Austin 78758
Tel (512) 835-4000
TWX. 910-874-1323

†Pioneer Electronics
13710 Omega Road
Dallas 75234
Tel (214) 386-7300
TWX. 910-850-5563

†Pioneer Electronics
5853 Point West Drive
Houston 77036
Tel (713) 988-5555
TWX: 910-881-1606

Wyle Distribution Group
1810 Greenville Avenue
Richardson 75081
Tel: (214) 235-9953

UTAH

Arrow Electronics
1946 Parkway Blvd
Salt Lake City 84119
Tel (801) 973-6913

†Hamilton/Avnet Electronics
1585 West 2100 South
Salt Lake City 84119
Tel· (801) 972-2800
TWX: 910-925-4018

Wyle Distribution Group
1325 West 2200 South
Suite E
West Valley 84119
Tel (801) 974-9953

WASHINGTON

†Almac Electronics Corp
14360 S E Eastgate Way
Bellevue 98007
Tel (206) 643-9992
TWX. 910-444-2067

Arrow Electronics, Inc.
19540 68th Ave South
Kent 98032
Tel (206) 575-4420

†Hamilton/Avnet Electronics
14212 N E 21st Street
Bellevue 98005
Tel· (206) 643-3950
TWX 910-443-2469

Wyle Distribution Group
15385 N E. 90th Street
Redmond 98052
Tel: (206) 881-1150

WISCONSIN

Arrow Electronics, Inc
200 N Patrick Blvd , Ste. 100
Brookfield 53005
Tel (414) 767-6600
TWX. 910-262-1193

Hamilton/Avnet Electronics
2975 Moorland Road
New Berlin 53151
Tel: (414) 784-4510
TWX. 910-262-1182

CANADA

ALBERTA

Hamilton/Avnet Electronics
2816 21st Street N E
Calgary T2E 6Z3
Tel: (403) 230-3586
TWX· 03-827-642

Zentronics
Bay No 1
3300 14th Avenue N.E
Calgary T2A 6J4
Tel (403) 272-1021

BRITISH COLUMBIA

†Hamilton/Avnet Electronics
105-2550 Boundary
Burmalay V5M 3Z3
Tel (604) 437-6667

Zentronics
108-11400 Bridgeport Road
Richmond V6X 1T2
Tel (604) 273-5575
TWX: 04-5077-89

MANITOBA

Zentronics
60-1313 Border Unit 60
Winnipeg R3H 0X4
Tel (204) 694-1957

ONTARIO

Arrow Electronics, Inc
36 Antares Dr
Nepean K2E 7W5
Tel (613) 226-6903

Arrow Electronics, Inc
1093 Meyerside
Mississauga L5T 1M4
Tel (416) 673-7769
TWX 06-218213

†Hamilton/Avnet Electronics
6845 Rexwood Road
Units 3-4-5
Mississauga L4T 1R2
Tel (416) 677-7432
TWX 610-492-8867

Hamilton/Avnet Electronics
6845 Rexwood Rd , Unit 6
Mississauga L4T 1R2
Tel (416) 277-0484

†Hamilton/Avnet Electronics
190 Colonnade Road South
Nepean K2E 7L5
Tel (613) 226-1700
TWX 05-349-71

†Zentronics
8 Tilbury Court
Brampton L6T 3T4
Tel. (416) 451-9600
TWX 06-976-78

†Zentronics
155 Colonnade Road
Unit 17
Nepean K2E 7K1
Tel (613) 226-8840

Zentronics
60-1313 Border St
Winnipeg R3H 0l4
Tel· (204) 694-7957

QUEBEC

†Arrow Electronics Inc
4050 Jean Talon Quest
Montreal H4P 1W1
Tel: (514) 735-5511
TWX. 05-25590

Arrow Electronics, Inc.
500 Avenue St-Jean Baptiste
Suite 280
Quebec G2E 5R9
Tel: (418) 871-7500
FAX 418-871-6816

Hamilton/Avnet Electronics
2795 Halpern
St. Laurent H2E 7K1
Tel (514) 335-1000
TWX· 610-421-3731

Zentronics
817 McCaffrey
St Laurent H4T 1M3
Tel (514) 737-9700
TWX 05-827-535

†Microcomputer System Technical Distributor Center

EUROPEAN SALES OFFICES

DENMARK

Intel Denmark A/S
Glentevej 61, 3rd Floor
2400 Copenhagen NV
Tel (45) (31) 19 80 33
TLX: 19567

FINLAND

Intel Finland OY
Ruosilantie 2
00390 Helsinki
Tel: (358) 0 544 644
TLX: 123332

FRANCE

Intel Corporation S A.R.L
1, Rue Edison-BP 303
78054 St Quentin-en-Yvelines
Cedex
Tel: (33) (1) 30 57 70 00
TLX: 699016

WEST GERMANY

Intel Semiconductor GmbH*
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen
Tel: (49) 089/90992-0
TLX: 5-23177

Intel Semiconductor GmbH
Hohenzollern Strasse 5
3000 Hannover 1
Tel: (49) 0511/344081
TLX: 9-23625

Intel Semiconductor GmbH
Abraham Lincoln Strasse 16-18
6200 Wiesbaden
Tel: (49) 06121/7605-0
TLX 4-186183

Intel Semiconductor GmbH
Zettachring 10A
7000 Stuttgart 80
Tel: (49) 0711/7287-280
TLX 7-254826

ISRAEL

Intel Semiconductor Ltd.*
Atidim Industrial Park-Neve Sharet
P.O. Box 43202
Tel-Aviv 61430
Tel: (972) 03-498080
TLX: 371215

ITALY

Intel Corporation Italia S.p.A.*
Milanofiori Palazzo E
20090 Assago
Milano
Tel: (39) (02) 89200950
TLX: 341286

NETHERLANDS

Intel Semiconductor B.V.*
Postbus 84130
3099 CC Rotterdam
Tel: (31) 10.407.11 11
TLX: 22283

NORWAY

Intel Norway A/S
Hvamvelen 4-PO Box 92
2013 Skjetten
Tel: (47) (6) 842 420
TLX: 78018

SPAIN

Intel Iberia S.A.
Zurbaran, 28
28010 Madrid
Tel: (34) (1) 308 25 52
TLX: 46880

SWEDEN

Intel Sweden A.B.*
Dalvagen 24
171 36 Solna
Tel. (46) 8 734 01 00
TLX: 12261

SWITZERLAND

Intel Semiconductor A.G.
Zuerichstrasse
8185 Winkel-Rueti bei Zuerich
Tel: (41) 01/860 62 62
TLX: 825977

UNITED KINGDOM

Intel Corporation (U.K.) Ltd.*
Pipers Way
Swindon, Wiltshire SN3 1RJ
Tel: (44) (0793) 696000
TLX: 444447/8

EUROPEAN DISTRIBUTORS/REPRESENTATIVES

AUSTRIA

Bacher Electronics G.m.b.H.
Rotenmuehlgasse 26
1120 Wien
Tel. (43) (0222) 83 56 46
TLX: 31532

BELGIUM

Inelco Belgium S A
Av. des Croix de Guerre 94
1120 Bruxelles
Oorlogskruisenlaan, 94
1120 Brussel
Tel: (32) (02) 216 01 60
TLX. 64475 or 22090

DENMARK

ITT-Multikomponent
Naverland 29
2600 Glostrup
Tel: (45) (0) 2 45 66 45
TLX. 33 355

FINLAND

OY Fintronic AB
Melkonkatu 24A
00210 Helsinki
Tel. (358) (0) 6926022
TLX 12422-1

FRANCE

Almex
Zone industrielle d'Antony
48, rue de l'Aubepine
BP 102
92164 Antony cedex
Tel. (33) (1) 46 66 21 12
TLX 250067

Jermyn-Generim
60, rue des Gemeaux
Silic 580
94653 Rungis cedex
Tel (33) (1) 49 78 49 78
TLX 261585

Metrologie
Tour d'Asnieres
4, av. Laurent-Cely
92606 Asnieres Cedex
Tel: (33) (1) 47 90 62 40
TLX 611448

Tekelec-Airtronic
Cite des Bruyeres
Rue Carle Vernet - BP 2
92310 Sevres
Tel: (33) (1) 45 34 75 35
TLX: 204552

WEST GERMANY

Electronic 2000 AG
Stahlgruberring 12
8000 Muenchen 82
Tel (49) 089/42001-0
TLX: 522561

ITT Multikomponent GmbH
Postfach 1265
Bahnhofstrasse 44
7141 Moeglingen
Tel: (49) 07141/4879
TLX: 7264472

Jermyn GmbH
Im Dachsstueck 9
6250 Limburg
Tel: (49) 06431/508-0
TLX: 415257-0

Metrologie GmbH
Meglingerstrasse 49
8000 Muenchen 71
Tel: (49) 089/78042-0
TLX: 5213189

Proelectron Vertriebs GmbH
Max Planck Strasse 1-3
6072 Dreieich
Tel: (49) 06103/30434-3
TLX: 417903

IRELAND

Micro Marketing Ltd.
Glenageary Office Park
Glenageary
Co. Dublin
Tel: (21) (353) (01) 85 63 25
TLX: 31584

ISRAEL

Eastronics Ltd.
11 Rozanis Street
P.O.B. 39300
Tel-Aviv 61392
Tel: (972) 03-475151
TLX: 33638

ITALY

Intesi
Divisione ITT Industries GmbH
Viale Milanofiori
Palazzo E/5
20090 Assago (MI)
Tel: (39) 02/824701
TLX: 311351

Lasi Elettronica S.p.A.
V. le Fulvio Testi, 126
20092 Cinisello Balsamo (MI)
Tel. (39) 02/2440012
TLX: 352040

Telcom S.r.l
Via M. Civitali 75
20148 Milano
Tel: (39) 02/4049046
TLX: 335654

ITT Multicomponents
Viale Milanofiori E/5
20090 Assago (MI)
Tel: (39) 02/824701
TLX: 311351

Silverstar
Via Dei Gracchi 20
20146 Milano
Tel: (39) 02/49961
TLX: 332189

NETHERLANDS

Koning en Hartman Elektrotechniek
B.V.
Energieweg 1
2627 AP Delft
Tel: (31) (0) 15/609906
TLX: 38250

NORWAY

Nordisk Elektronikk (Norge) A/S
Postboks 123
Smedsvingen 4
1364 Hvalstad
Tel: (47) (02) 84 62 10
TLX: 77546

PORTUGAL

ATD Portugal LDA
Rua Dos Lusiados, 5 Sala B
1300 Lisboa
Tel: (35) (1) 64 80 91
TLX: 61562

Ditram
Avenida Miguel Bombarda, 133
1000 Lisboa
Tel: (35) (1) 54 53 13
TLX: 14182

SPAIN

ATD Electronica, S A.
Plaza Ciudad de Viena, 6
28040 Madrid
Tel: (34) (1) 234 40 00
TLX: 42477

ITT-SESA
Calle Miguel Angel, 21-3
28010 Madrid
Tel. (34) (1) 419 09 57
TLX: 27461

Metrologia Iberica, S.A.
Ctra de Fuencarral, n.80
28100 Alcobendas (Madrid)
Tel (34) (1) 653 86 11

SWEDEN

Nordisk Elektronik AB
Torshamnsgatan 39
Box 36
164 93 Kista
Tel. (46) 08-03 46 30
TLX: 105 47

SWITZERLAND

Industrade A G
Hertistrasse 31
8304 Wallisellen
Tel: (41) (01) 8328111
TLX: 56788

TURKEY

EMPA Electronic
Lindwurmstrasse 95A
8000 Muenchen 2
Tel: (49) 089/53 80 570
TLX: 528573

UNITED KINGDOM

Accent Electronic Components Ltd.
Jubilee House, Jubilee Road
Letchworth, Herts SG6 1TL
Tel: (44) (0462) 686666
TLX: 826293

Bytech-Comway Systems
3 The Western Centre
Western Road
Bracknell RG12 1RW
Tel. (44) (0344) 55333
TLX. 847201

Jermyn
Vestry Estate
Otford Road
Sevenoaks
Kent TN14 5EU
Tel. (44) (0732) 450144
TLX: 95142

MMD
Unit 8 Southview Park
Caversham
Reading
Berkshire RG4 0AF
Tel. (44) (0734) 481666
TLX: 846669

Rapid Silicon
Rapid House
Denmark Street
High Wycombe
Buckinghamshire HP11 2ER
Tel: (44) (0494) 442266
TLX: 837931

Rapid Systems
Rapid House
Denmark Street
High Wycombe
Buckinghamshire HP11 2ER
Tel: (44) (0494) 450244
TLX: 837931

YUGOSLAVIA

H.R. Microelectronics Corp
2005 de la Cruz Blvd., Ste 223
Santa Clara, CA 95050
U.S.A
Tel: (1) (408) 988-0286
TLX 387452

Rapido Electronic Components
S.p.a.
Via C Beccaria, 8
34133 Trieste
Italia
Tel. (39) 040/360555
TLX: 460461

*Field Application Location

INTERNATIONAL SALES OFFICES

AUSTRALIA

Intel Australia Pty. Ltd *
Spectrum Building
200 Pacific Hwy., Level 6
Crows Nest, NSE, 2065
Tel 612-957-2744
FAX: 612-923-2632

BRAZIL

Intel Semicondutores do Brazil LTDA
Av Paulista, 1159-CJS 404/405
01311 - Sao Paulo - S.P.
Tel 55-11-287-5899
TLX: 3911153146 ISDB
FAX: 55-11-287-5119

CHINA/HONG KONG

Intel PRC Corporation
15/F, Office 1, Citic Bldg.
Jian Guo Men Wai Street
Beijing, PRC
Tel: (1) 500-4850
TLX: 22947 INTEL CN
FAX. (1) 500-2953

Intel Semiconductor Ltd.*
10/F East Tower
Bond Center
Queensway, Central
Hong Kong
Tel (5) 8444-555
TLX: 63869 ISHLHK HX
FAX (5) 8681-989

INDIA

Intel Asia Electronics, Inc
4/2, Samrah Plaza
St Mark's Road
Bangalore 560001
Tel 011-91-812-215065
TLX: 9538452875 DCBY
FAX: 091-812-215067

JAPAN

Intel Japan K K
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26
Tel: 0298-47-8511
TLX: 3656-160
FAX. 029747-8450

Intel Japan K.K *
Daiichi Mitsugi Bldg
1-8889 Fuchu-cho
Fuchu-shi, Tokyo 183
Tel. 0423-60-7871
FAX. 0423-60-0315

Intel Japan K K *
Bldg. Kumagaya
2-69 Hon-cho
Kumagaya-shi, Saitama 360
Tel: 0485-24-6871
FAX: 0485-24-7518

Intel Japan K K *
Mitsui-Seimei Musashi-kosugi Bldg
915 Shinmaruko, Nakahara-ku
Kawasaki-shi, Kanagawa 211
Tel: 044-733-7011
FAX 044-733-7010

Intel Japan K K
Nihon Seimei Atsugi Bldg.
1-2-1 Asahi-machi
Atsugi-shi, Kanagawa 243
Tel 0462-29-3731
FAX 0462-29-3781

Intel Japan K K.*
Ryokuchi-Eki Bldg.
2-4-1 Terauchi
Toyonaka-shi, Osaka 560
Tel 06-863-1091
FAX: 06-863-1084

Intel Japan K K.
Shinmaru Bldg.
1-5-1 Marunouchi
Chiyoda-ku, Tokyo 100
Tel: 03-201-3621
FAX: 03-201-6850

Intel Japan K K
Green Bldg
1-16-20 Nishiki
Naka-ku, Nagoya-shi
Aichi 450
Tel 052-204-1261
FAX. 052-204-1285

KOREA

Intel Technology Asia, Ltd
16th Floor, Life Bldg
61 Yoido-dong, Youngdeungpo-Ku
Seoul 150-010
Tel (2) 784-8186, 8286, 8386
TLX: K29312 INTELKO
FAX (2) 784-8096

SINGAPORE

Intel Singapore Technology, Ltd
101 Thomson Road #21-05/06
United Square
Singapore 1130
Tel: 250-7811
TLX 39921 INTEL
FAX. 250-9256

TAIWAN

Intel Technology Far East Ltd
8th Floor, No 205
Bank Tower Bldg
Tung Hua N. Road
Taipei
Tel 886-2-716-9660
FAX. 886-2-717-2455

INTERNATIONAL DISTRIBUTORS/REPRESENTATIVES

ARGENTINA

DAFSYS S R L
Chacabuco, 90-6 PISO
1069-Buenos Aires
Tel. 54-1-334-7726
FAX: 54-1-334-1871

AUSTRALIA

Email Electronics
15-17 Hume Street
Huntingdale, 3166
Tel 011-61-3-544-8244
TLX: AA 30895
FAX: 011-61-3-543-8179

NSD-Australia
205 Middleborough Rd.
Box Hill, Victoria 3128
Tel 03 8900970
FAX 03 8990819

BRAZIL

Elebra Microelectronica S A
Rua Geraldo Flausina Gomes, 78
10th Floor
04575 - Sao Paulo - S P.
Tel 55-11-534-9641
TLX. 55-11-54593/54591
FAX. 55-11-534-9424

CHILE

DIN Instruments
Suecia 2323
Casilla 6055, Correo 22
Santiago
Tel: 56-2-225-8139
TLX: 240.846 RUD

CHINA/HONG KONG

Novel Precision Machinery Co , Ltd.
Flat D, 20 Kingsford Ind Bldg.
Phase 1, 26 Kwai Hei Street
N.T., Kowloon
Hong Kong
Tel 852-0-4223222
TWX 39114 JINMI HX
FAX: 852-0-4261602

INDIA

Micronic Devices
Arun Complex
No 65 D V.G. Road
Basavanagudi
Bangalore 560 004
Tel: 011-91-812-600-631
011-91-812-611-365
TLX. 9538458332 MDBG

Micronic Devices
No. 516 5th Floor
Swastik Chambers
Sion, Trombay Road
Chembur
Bombay 400 071
TLX. 9531 171447 MDEV

Micronic Devices
25/8, 1st Floor
Bada Bazaar Marg
Old Rajinder Nagar
New Delhi 110 060
Tel: 011-91-11-5723509
011-91-11-589771
TLX: 031-63253 MDND IN

Micronic Devices
6-3-348/12A Dwarakapuri Colony
Hyderabad 500 482
Tel: 011-91-842-226748

S&S Corporation
1587 Kooser Road
San Jose, CA 95118
Tel (408) 978-6216
TLX: 820281
FAX. (408) 978-8635

JAPAN

Asahi Electronics Co. Ltd.
KMM Bldg 2-14-1 Asano
Kokurakita-ku
Kitakyushu-shi 802
Tel: 093-511-6471
FAX 093-551-7861

C. Itoh Techno-Science Co , Ltd
4-8-1 Dobashi, Miyamae-ku
Kawasaki-shi, Kanagawa 213
Tel: 044-852-5121
FAX. 044-877-4268

Dia Semicon Systems, Inc.
Flower Hill Shinmachi Higashi-kan
1-23-9 Shinmachi, Setagaya-ku
Tokyo 154
Tel 03-439-1600
FAX 03-439-1601

Okaya Koki
2-4-18 Sakae
Naka-ku, Nagoya-shi 460
Tel 052-204-2916
FAX: 052-204-2901

Ryoyo Electro Corp
Konwa Bldg
1-12-22 Tsukiji
Chuo-ku, Tokyo 104
Tel: 03-546-5011
FAX: 03-546-5044

KOREA

J-Tek Corporation
6th Floor, Government Pension Bldg
24-3 Yoido-dong
Youngdeungpo-ku
Seoul 150-010
Tel: 82-2-780-8039
TLX. 25299 KODIGIT
FAX: 82-2-784-8391

Samsung Electronics
150 Taepyungro-2 KA
Chungku, Seoul 100-102
Tel: 82-2-751-3985
TLX: 27970 KORSST
FAX: 82-2-753-0967

MEXICO

SSB Electronics, Inc
675 Palomar Street, Bldg. 4, Suite A
Chula Vista, CA 92011
Tel: (619) 585-3253
TLX. 287751 CBALL UR
FAX (619) 585-8322

Dicopel S A
Tochtli 368 Fracc Ind. San Antonio
Azcapotzalco
C P. 02760-Mexico, D F.
Tel 52-5-561-3211
TLX: 177 3790 Dicome
FAX: 52-5-561-1279

PSI de Mexico
Francisco Villas Esq Ajusto
Cuernavaca – Morelos – CEP 62130
Tel: 52-73-13-9412
FAX: 52-73-17-5333

NEW ZEALAND

Email Electronics
36 Olive Road
Penrose, Auckland
Tel 011-64-9-591-155
FAX 011-64-9-592-681

SINGAPORE

Electronic Resources Pte, Ltd
17 Harvey Road #04-01
Singapore 1336
Tel: 283-0888
TWX 56541 ERS
FAX: 2895327

SOUTH AFRICA

Electronic Building Elements
178 Erasmus Street (off Watermeyet Street)
Meyerspark, Pretoria, 0184
Tel. 011-2712-803-7680
FAX: 011-2712-803-8294

TAIWAN

Micro Electronics Corporation
5/F 587, Ming Shen East Rd.
Taipei, R O C
Tel 886-2-501-8231
FAX 886-2-505-6609

Sertek
15/F 135, Section 2
Chien Juo North Rd.
Taipei 10479, R O.C.
Tel: (02) 5010055
FAX: (02) 5012521
(02) 5058414

VENEZUELA

P Benavides S.A.
Avilanes a Rio
Residencia Kamarata
Locales 4 AL 7
La Candelaria, Caracas
Tel: 58-2-574-6338
TLX. 28450
FAX: 58-2-572-3321

*Field Application Location

DOMESTIC SERVICE OFFICES

ALABAMA

*Intel Corp
5015 Bradford Dr , Suite 2
Huntsville 35805
Tel: (205) 830-4010

ALASKA

Intel Corp.
c/o TransAlaska Data Systems
300 Old Steese Hwy.
Fairbanks 99701-3120
Tel: (907) 452-4401

Intel Corp.
c/o TransAlaska Data Systems
1551 Lore Road
Anchorage 99507
Tel: (907) 522-1776

ARIZONA

*Intel Corp.
11225 N 28th Dr.
Suite D-214
Phoenix 85029
Tel (602) 869-4980

*Intel Corp
500 E. Fry Blvd., Suite M-15
Sierra Vista 85635
Tel (602) 459-5010

CALIFORNIA

†Intel Corp.
21515 Vanowen St., Ste. 116
Canoga Park 91303
Tel: (818) 704-8500

*Intel Corp
2250 E Imperial Hwy., Ste. 218
El Segundo 90245
Tel (213) 640-6040

*Intel Corp
1900 Prairie City Rd.
Folsom 95630-9597
Tel. (916) 351-6143
1-800-468-3548

Intel Corp
9665 Cheasapeake Dr., Suite 325
San Diego 92123-1326
Tel. (619) 292-8086

**Intel Corp.
400 N. Tustin Avenue
Suite 450
Santa Ana 92705
Tel (714) 835-9642

**†Intel Corp
San Tomas 4
2700 San Tomas Exp., 2nd Floor
Santa Clara 95051
Tel (408) 986-8086

COLORADO

*Intel Corp.
650 S. Cherry St , Suite 915
Denver 80222
Tel: (303) 321-8086

CONNECTICUT

*Intel Corp
301 Lee Farm Corporate Park
83 Wooster Heights Rd.
Danbury 06810
Tel (203) 748-3130

FLORIDA

**Intel Corp.
6363 N W. 6th Way, Ste. 100
Ft. Lauderdale 33309
Tel (305) 771-0600

*Intel Corp
5850 T.G. Lee Blvd , Ste 340
Orlando 32822
Tel (407) 240-8000

GEORGIA

*Intel Corp.
3280 Pointe Pkwy., Ste 200
Norcross 30092
Tel. (404) 449-0541

HAWAII

*Intel Corp.
U S.I S.C Signal Batt
Building T-1521
Shafter Plats
Shafter 96856

ILLINOIS

**†Intel Corp.
300 N. Martingale Rd., Ste 400
Schaumburg 60173
Tel (312) 605-8031

INDIANA

*Intel Corp.
8777 Purdue Rd , Ste 125
Indianapolis 46268
Tel. (317) 875-0623

KANSAS

*Intel Corp.
10985 Cody, Suite 140
Overland Park 66210
Tel: (913) 345-2727

MARYLAND

**†Intel Corp.
10010 Junction Dr., Suite 200
Annapolis Junction 20701
Tel. (301) 206-2860
FAX 301-206-3677

MASSACHUSETTS

**†Intel Corp.
3 Carlisle Rd., 2nd Floor
Westford 01886
Tel (508) 692-1060

MICHIGAN

*†Intel Corp.
7071 Orchard Lake Rd , Ste 100
West Bloomfield 48322
Tel (313) 851-8905

MINNESOTA

*†Intel Corp
3500 W. 80th St., Suite 360
Bloomington 55431
Tel: (612) 835-6722

MISSOURI

*Intel Corp
4203 Earth City Exp., Ste 131
Earth City 63045
Tel: (314) 291-1990

NEW JERSEY

**Intel Corp.
300 Sylvan Avenue
Englewood Cliffs 07632
Tel. (201) 567-0821

*Intel Corp
Parkway 109 Office Center
328 Newman Springs Road
Red Bank 07701
Tel (201) 747-2233

*Intel Corp
280 Corporate Center
75 Livingston Ave., 1st Floor
Roseland 07068
Tel. (201) 740-0111

NEW YORK

*†Intel Corp
2950 Expressway Dr. South
Islandia 11722
Tel 516) 231-3300

*Intel Corp.
Westage Business Center
Bldg. 300, Route 9
Fishkill 12524
Tel. (914) 897-3860

NORTH CAROLINA

*Intel Corp.
5800 Executive Dr., Ste. 105
Charlotte 28212
Tel (704) 568-8966

**Intel Corp
2700 Wycliff Road
Suite 102
Raleigh 27607
Tel (919) 781-8022

OHIO

**†Intel Corp.
3401 Park Center Dr., Ste 220
Dayton 45414
Tel: (513) 890-5350

*†Intel Corp.
25700 Science Park Dr , Ste 100
Beachwood 44122
Tel: (216) 464-2736

OREGON

Intel Corp.
15254 N.W. Greenbrier Parkway
Building B
Beaverton 97005
Tel. (503) 645-8051

*Intel Corp.
5200 N.E. Elam Young Parkway
Hillsboro 97123
Tel: (503) 681-8080

PENNSYLVANIA

*†Intel Corp.
455 Pennsylvania Ave., Ste 230
Fort Washington 19034
Tel: (215) 641-1000

†Intel Corp
400 Penn Center Blvd , Ste. 610
Pittsburgh 15235
Tel: (412) 823-4970

Intel Corp.
1513 Cedar Cliff Dr.
Camp Hill 17011
Tel: (717) 761-0860

PUERTO RICO

Intel Corp.
South Industrial Park
P.O. Box 910
Las Piedras 00671
Tel: (809) 733-8616

TEXAS

Intel Corp
8815 Dyer St , Suite 225
El Paso 79904
Tel: (915) 751-0186

*Intel Corp.
313 E Anderson Lane, Suite 314
Austin 78752
Tel (512) 454-3628

**†Intel Corp.
12000 Ford Rd., Suite 401
Dallas 75234
Tel: (214) 241-8087

*Intel Corp
7322 S.W Freeway, Ste. 1490
Houston 77074
Tel (713) 988-8086

UTAH

Intel Corp
428 East 6400 South, Ste. 104
Murray 84107
Tel: (801) 263-8051

VIRGINIA

*Intel Corp.
1504 Santa Rosa Rd., Ste. 108
Richmond 23288
Tel (804) 282-5668

WASHINGTON

*Intel Corp.
155 108th Avenue N.E., Ste. 386
Bellevue 98004
Tel. (206) 453-8086

CANADA

ONTARIO

Intel Semiconductor of
Canada, Ltd.
2650 Queensview Dr., Ste. 250
Ottawa K2B 8H6
Tel. (613) 829-9714
FAX: 613-820-5936

Intel Semiconductor of
Canada, Ltd.
190 Attwell Dr., Ste. 102
Rexdale M9W 6H8
Tel: (416) 675-2105
FAX 416-675-2438

CUSTOMER TRAINING CENTERS

CALIFORNIA

2700 San Tomas Expressway
Santa Clara 95051
Tel (408) 970-1700
1-800-421-0386

ILLINOIS

300 N. Martingale Road
Suite 300
Schaumburg 60173
Tel: (708) 706-5700
1-800-421-0386

MASSACHUSETTS

3 Carlisle Road, First Floor
Westford 01886
Tel: (301) 220-3380
1-800-328-0386

MARYLAND

10010 Junction Dr.
Suite 200
Annapolis Junction 20701
Tel: (301) 206-2860
1-800-328-0386

SYSTEMS ENGINEERING MANAGERS OFFICES

MINNESOTA

3500 W 80th Street
Suite 360
Bloomington 55431
Tel: (612) 835-6722

NEW YORK

2950 Expressway Dr., South
Islandia 11722
Tel 506) 231-3300

†System Engineering locations
*Carry-in locations
**Carry-in/mail-in locations

intel®

Microcomputer Boards and Systems

Intel's complete line of OEM microcomputer systems takes advantage of Intel's technology to provide economical, self-contained, computer-based solutions for OEM applications. The extensive use of high integration VLSI produces high-performance, single board systems.

Intel has a broad range of hardware and software products for both software development and real-time applications. The Intel Custom Board and Systems Group offers a broad array of custom services for OEMs seeking unique solutions for their markets.

This handbook contains application notes, article reprints and data sheets for Intel OEM boards and systems including such key products as Multibus®, BITBUS™ and iRMX® operating systems. The data sheets include specifications and features; charts covering configuration and architecture; block diagrams; operational descriptions and functional descriptions. In addition, technical briefs provide helpful information about subsystems. An entire section of this handbook is devoted to explanations of the extensive service and support, engineering services, customer training and hardware maintenance services offered by Intel.

Printed in U.S.A. 0190 34K RRD BC
Microcomputer Systems

ISBN 1-55512-089-X