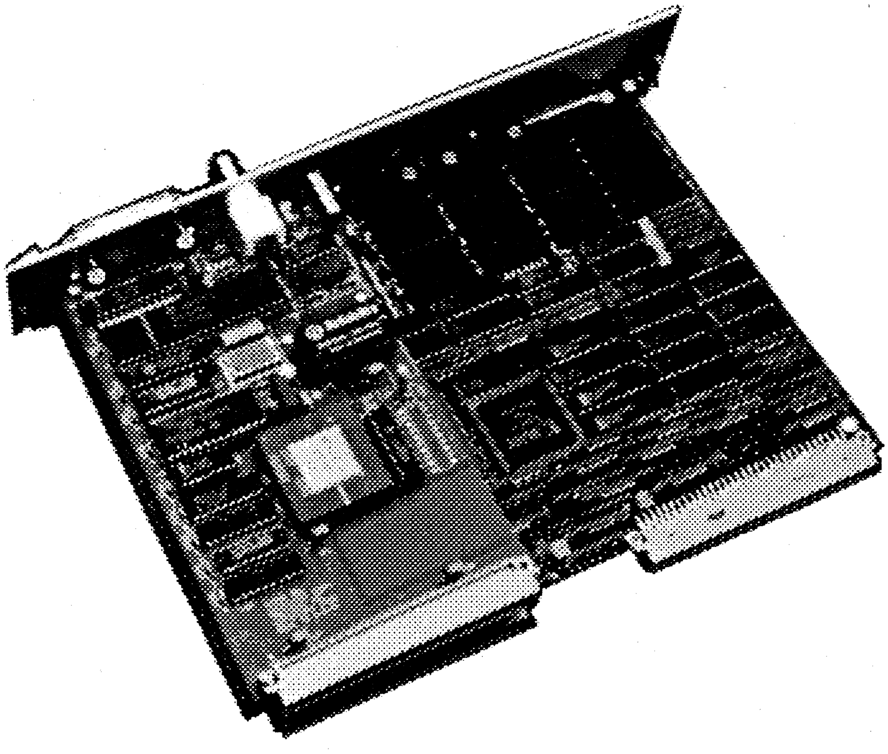
HEURIKON CORP.

HK68/V30/HK68/V30XE User's Manual



Heurikon Corporation
8310 Excelsior Drive
Madison, WI 53717

HK68/V30 / HK68/V30XE USER'S MANUAL

VERSION D
AUGUST 1989

PREFACE

The purpose of this manual is to document the features of the Heurikon HK68/V30 and HK68/V30XE microcomputer boards.

This manual covers the unique features of the HK68/V30 and HK68/V30XE boards. Although general information, such as MPU, SCSI, CIO, and SCC programming is discussed, more detailed information is available directly from the chip manufacturers.

NOTE: The body of this manual is written for board revision levels 2 and later. Appendix A notes the differences which apply to board revision levels P and 1.

Feel free to contact Heurikon Corporation's Customer Support Department if you have questions. We are prepared to answer general questions as well as providing help with specific applications.

The information in the manual has been checked and is believed to be accurate and reliable. HOWEVER, NO RESPONSIBILITY IS ASSUMED BY HEURIKON FOR ITS USE OR FOR ANY INACCURACIES. Specifications are subject to change without notice. HEURIKON DOES NOT ASSUME ANY LIABILITY ARISING OUT OF USE OR OTHER APPLICATION OF ANY PRODUCT, CIRCUIT OR PROGRAM DESCRIBED HEREIN. This document does not convey any license under Heurikon's patents or the rights of others.

HK68, HK68/V30, HK68/V30XE and Hbug are trademarks of Heurikon Corporation

UNIX is a trademark of AT&T Bell Laboratories.

VMEbus is a trademark of Motorola.

Ethernet is a trademark of Xerox Corporation.

Copyright 1989 Heurikon Corporation. All rights reserved.

TABLE OF CONTENTS

Chapter 11 SERIAL I/O

Chapter 12 SCSI PORT

Chapter 13 CENTRONICS PORT

Chapter 14 REAL-TIME CLOCK (RTC) - OPTIONAL FEATURE

Chapter 15 ETHERNET SUPPORT - THE XE MODULE (OPTIONAL)

Chapter 16 VMEBUS INTERFACE

Chapter 17 SUMMARY INFORMATION

Appendix A HK68/V30 BOARD REVISION DIFFERENCES

Tables

Figures

Chapter 1

AN OVERVIEW OF THE HK68/V30

1.1 FEATURES SUMMARY

CPU

Motorola 68030 microprocessor chip.
20 MHz., 25 and 33 MHz. option.
32-bit internal architecture.
32-bit address and data paths.
4-gigabyte addressing range.
On-Chip Memory Management Unit.
256-byte Data Cache and 256-byte Instruction Cache.
(See Chapter 3 of this manual)

FPP

Motorola 68881 or 68882
IEEE-P754 Binary Floating Point Standard
(See Chapter 5 of this manual)

DMAC

On-card WE32104/204 DMA controller.
10 MHz. 18 and 24 MHz. option.
Supports high-speed SCSI and SCC data transfers.
Four channels.
(See Chapter 4 of this manual)

RAM

4 or 16 megabyte capacity.
One parity bit per byte.
Uses 1024K x 1 or 4096K x 1 DRAMs.
Hardware refresh.
(See Chapter 7 of this manual)

EpROM

One ROM socket.
1 Mbyte total capacity.
Page Addressable ROM and EEpROM capability.
(See Chapter 7 of this manual)

VMEbus

24-bit (standard), 32-bit (extended) and 16-bit (short) addressing.
32-bit data bus, compatible with 16 and 8-bit boards.
7-level interrupt handler.
7-level interrupter.
4-level bus arbiter, priority or round robin.
(See Chapter 8 of this manual)

Serial I/O

Two serial I/O ports (Zilog Z8530 Serial Communication Controller).
Separate baud rate generators for each port.
Asynchronous and synchronous modes.
RS-232C interface, RS-422 option.
(See Chapter 11 of this manual)

SCSI

ANSI X3T9.2 compatible SCSI controller.
Supports up to 7 disk drive controllers or other devices.
Synchronous protocol support.
(See Chapter 12 of this manual)

Centronics

Control I/F and eight-bit output port for Centronics-type printer.
(See Chapter 13 of this manual)

LEDs

Four user LEDs under software control.
Seven MPU/DMAC/BUS status LEDs
(See Chapter 9 of this manual)

CIO

Zilog Z8536 Counter/Timer and Parallel I/O Unit.
Three 16-bit counter/timers.
Three parallel ports for on-card control functions.
(See Chapter 10 of this manual)

NV-RAM

Nonvolatile Static RAM; 256 x 4 configuration.
Internal EEpROM.
100 year retention; 10,000 store cycle lifetime.
For user definable functions.
(See Chapter 7.8 of this manual)

Mailbox

Allows remote control of the HK68/V30 via specified VMEbus addresses.
MPU halt, reset, interrupt, and on-card bus lock functions.
(See Chapter 8.9 of this manual)

RTC

Optional Real-Time Clock module for time-of-day maintenance. With battery backup.
(See Chapter 14 of this manual)

Ethernet

Optional XE plugover module provides Ethernet/Thin-Ethernet support
Single-height VME card (3U) implementation.
32K byte static RAM for packet buffering.
32-byte ROM for Ethernet Physical Node address.
National Semiconductor 8390 Network Interface Controller (NIC).
National Semiconductor 8391 Serial Network Interface (SNI).
National Semiconductor 8392 Coaxial Transceiver Interface (CTI).
(See Chapter 15 of this manual)

1.2 Block Diagram

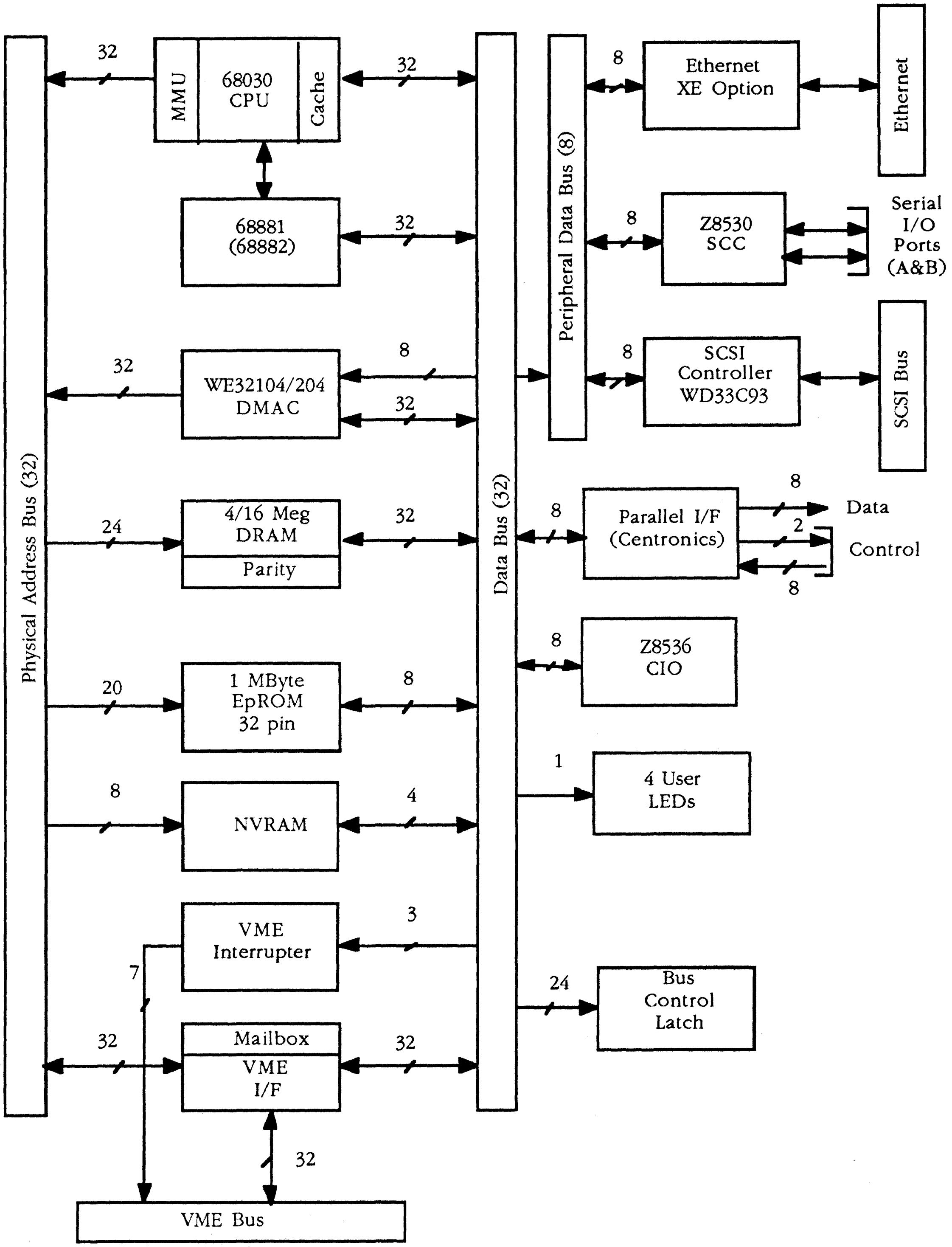


Fig. 1-1. HK68/V30 Block Diagram

Chapter 2

GETTING GOING

2.1 INSTALLATION STEPS

Here is what you need to get the Heurikon HK68/V30 "on-the-air":

1. Heurikon HK68/V30 Microcomputer board

2. Card cage and power supply

3. Serial I/F cable (RS-232)

4. CRT Terminal

5. Heurikon Hbug monitor and bootstrap EPROM

CAUTION: All semiconductors should be handled with care. Static discharges can easily damage the components on the HK68/V30. Keep the board in an anti-static bag whenever it is out of the system chassis and do not handle the board unless absolutely necessary. Ground your body before touching the HK68/V30 board.

CAUTION: High operating temperatures will cause unpredictable operation. Because of the high chip density, fan cooling is required for most configurations, even when cards are placed on extenders.

All products are fully tested before they are shipped from the factory. When you receive your HK68/V30, follow these steps to assure yourself that the system is operational:

1. Visually inspect the board(s) for loose components which could be the result of shipping vibrations. Visually inspect the chassis and all cables. Be sure all boards are seated properly in the card cage. Be sure all cables are securely in place.

2. Connect a CRT terminal to Serial Port B, via connector P3. If you are making your own cables, refer to section 11. Set the terminal as follows:

- 9600 baud, full duplex.
- Eight data bits (no parity).
- Two stop bits for transmit data.
- One stop bit for receive data.
- If your terminal does not have separate controls for transmit and receive stop bits, select one stop bit for both transmit and receive.

3. Connect AC power and turn the system on.

4. Push the system RESET button. A sign-on message and prompt from the monitor should appear on the screen. If not, check your power supply voltages and CRT cabling.

5. Now is the time to read the monitor manual and the operating system literature. Short course: To boot the operating system, insert a diskette and enter 'bf' (for boot floppy) or 'bw' (to boot from Winchester.)

6. Reconfigure the jumpers, etc, as necessary for your application. See section 16 for a summary of I/O device addresses and configuration jumpers.

2.2 TROUBLESHOOTING AND SERVICE INFORMATION

In case of difficulty, use this checklist:

1. Be sure the system is not overheating.

2. Inspect the power cables and connectors. If the HK68/V30 board has power, the large chips should feel warm to your touch.

3. If the Hbug monitor program is executing, run the diagnostics, via command 'uc' or 'um'.

4. Check your power supply for proper DC voltages. If possible, look for excessive power supply ripple or noise using an oscilloscope. Note that the use of P2 is required to meet the power specifications.

5. Check the chips to be sure they are firmly in place. Look for chips with bent or broken pins. In particular, you should check the EpROM (Section 7.2 of this manual describes the correct EpROM placement).

6. Check your terminal switches and cables. Be sure the P3 connector is on properly. The cable stripe (wire #1) should be toward the edge of the HK68/V30 board. The port B portion

of the cable is on the pin #34 side. If you have made your own cables, pay particular attention to the cable drawings in section 11.

7. Check the jumpers to be sure your board is configured properly. All jumpers should be in the "standard configuration" positions listed in section 17.3.

8. After you have checked all of the above items, call our Customer Service Department for help. Please have the following information handy:

- The monitor program revision level (part of sign-on message).
- The HK68/V30 p.c.b. serial number (scribed along card edge).
- The complete HK68/V30 model number, including option codes.
- The serial number of the Operating System.

If you plan to return the board to Heurikon for service, contact our Customer Service Department to obtain a *Return Merchandise Authorization (RMA)* number. Be prepared to provide the items listed above, plus your Purchase Order number and billing information if your HK68/V30 is out of warranty. If you return the board, be sure to enclose it in the anti-static bag, such as the one in which it was originally shipped. Send it prepaid to:

Heurikon Corporation
Factory Service Department
3201 Latham Drive
Madison, WI 53713

Please put the RMA number on the package so we can handle your problem most efficiently.

2.3 Monitor Summary

The HK68/V30 monitor and bootstrap program, Hbug, is contained in one EPROM. It is intended to provide a fundamental ability to check the memory and I/O devices, to manually enter a program and to down-line load or bootstrap a larger program into memory. Advanced features and utilities may be loaded from media or via an operating system.

Refer to the Hbug manual for details on the commands and command formats.

Chapter 3

MPU SUMMARY INFORMATION

3.1 INTRODUCTION

This section details some of the important features of the 68030 MPU chip and, in particular, those items which are specific to the implementation on the Heurikon HK68/V30.

3.2 MPU INTERRUPTS

The MPU can internally set an interrupt priority level in such a way that interrupts of a lower priority will not be honored. Interrupt level seven, however, cannot be masked off.

Each interrupt source may also be individually disabled, with the exception of Parity Error. The VMEbus interrupts and SCSI interrupt are masked on and off via the CIO (Refer to sections 8.7 and 10.2). The CIO, DMAC, SCC, Mailbox and Centronics interfaces each have their own enable/disable logic. ACFAIL may be disabled by removing jumper J29.

Level	Interrupt (bus)	Interrupt (on-card)
7	IRQ7	Parity error, ACFAIL (highest priority) non-maskable, autovectored
6	IRQ6	CIO (vectored) (sub-priority: timer 3, port A, timer 2, port B, timer 1)
5	IRQ5	DMAC (vectored) Ethernet (autovectored) (XE option only)
4	IRQ4	SCSI (autovectored)
3	IRQ3	SCC (autovectored)
2	IRQ2	Mailbox (autovectored) Interrupt
1	IRQ1	Centronics (autovectored) Interrupt
0		Idle, no interrupt

Table 3-1. MPU Interrupt Levels

When an interrupt is recognized by the MPU, the current instruction is completed and an interrupt acknowledge sequence is initiated, whose purpose is to acquire an interrupt vector from the interrupting device. The vector number is used to select one of 256 exception vectors located in reserved memory locations (see section 3.3 for a listing.) The exception vector specifies the address of the interrupt service routine.

In case there are two interrupts pending at the same level, the on-card device is serviced before the bus interrupt.

The CIO and DMAC devices on the HK68/V30 are capable of generating more than one vector, depending on the particular condition which caused the interrupt. This significantly reduces the time required to service the interrupt because the program does not have to rigorously test for the interrupt cause. The SCC is autovectored due to DMAC limitations. Section 8.7 has more information on the HK68/V30 interrupt logic. The VMEbus interrupts are vectored; the vector is automatically read from the interrupting device.

The HK68/V30 interrupt logic interprets a spurious interrupt as a VMEbus interrupt and attempts to read a vector from the bus. A spurious interrupt is one which is long enough to be recognized as an interrupt, but not long enough to be acknowledged.

3.3 MPU Exception Vectors

Exception vectors are memory locations from which the MPU fetches the address of a routine to handle an exception (interrupt). All exception vectors are two words long (four bytes), except for the reset vector which is four words. The listing below shows the vector space as it appears to the Heurikon HK68/V30 MPU. It is more specific than the 68030 MPU manual listing due to particular implementations on the HK68/V30 board. Refer to the MPU documentation for more details. The vector table normally occupies the first 1024 bytes of RAM, but may be moved to other locations under software control. Unused vector positions may be used for other purposes (e.g., code or data) or point to an error routine.

Vector	Address Offset	Assignment
0	000	Reset: Initial SSP (Supervisor Stack Pointer)
1	004	Reset: Initial PC (Supr Program Counter)
2	008	Bus Error (Watchdog Timer)
3	00C	Address Error
4	010	Illegal Instruction
5	014	Divide by Zero
6	018	CHK Instruction (register bounds)
7	01C	TRAPV Instruction (overflow)
8	020	Privilege Violation (STOP, RESET, RTE, etc)
9	024	Trace (Program development tool)
10	028	Instruction Group 1010 Emulator
11	02C	FPP Coprocessor not present
12	030	(reserved)
13	034	FPP Coprocessor Protocol Violation
14	038	Format Error
15	03C	Uninitialized Interrupt
16-23	040-05F	(reserved-8)
24	060	Spurious Interrupt, not used
25	064	Level 1 autovector, Centronics
26	068	Level 2 autovector, VME mailbox
27	06C	Level 3 autovector, SCC
28	070	Level 4 autovector, SCSI Interrupt
29	074	Level 5 autovector, Ethernet (XE option)
30	078	Level 6 autovector, not used
31	07C	Level 7 autovector, parity error, ACFAIL
32-47	080-0BF	TRAP Instruction Vectors (16)
48-54	0C0-0DB	FPP Exceptions (8)
55	0DC	(reserved)
56	0E0	MMU Configuration Error
57-63	0E4-0FF	(reserved-6)
64-255	100-3FF	User Interrupt Vectors (192)

Table 3-2. MPU Exception Vectors

Autovectoring is used for SCC, Centronics, VMEbus mailbox, AC-FAIL, parity error and SCSI. Interrupts from the CIO and DMAC can be programmed to provide a vector number (which would likely point into the "User Interrupt Vector" area, above). VMEbus interrupts (IRQ1 - IRQ7) are vectored; the vector is supplied by the interrupting device over the VMEbus.

The following table gives suggested interrupt vectors for each of the possible (on-card) device interrupts which could occur. Note that the listing is in order of interrupt priority, highest priority first.

Level	Vector	Device	Condition
7	31		Parity error or ACFAIL autovectored interrupt
6	96	CIO	Timer 3
	98	CIO	Timer 2
	78	CIO	External Interrupt (P5-11)
	76		SCSI Reset
	74		SCC Port A Ring Indicator
	72		SCC Port B Ring Indicator
	70		VME SYSFAIL
	68		VME Interrupt in Progress
	66		VME Logic Init Complete
	64		(CIO Port B bit 0)
	100	CIO	Timer 1
	102	CIO	Timer, error
5	112	DMAC	DMAC Chnl 0 (SCSI) error
	113		DMAC Chnl 0 (SCSI) normal
	114		DMAC Chnl 1 (Serial) error
	115		DMAC Chnl 1 (Serial) normal
	116		DMAC Chnl 2 (Ethernet) error (XE option)
	117		DMAC Chnl 2 (Ethernet) normal (XE option)
	118		DMAC Chnl 3 (Serial) error
	119		DMAC Chnl 3 (Serial) normal
	29	Ethernet	Ethernet autovectored interrupt (XE option)
4	28	SCSI	SCSI autovectored interrupt
3	27	SCC	SCC autovectored interrupt
2	26	VME	VMEbus mailbox autovectored interrupt
1	25	Cent	Centronics autovectored interrupt

Table 3-3. Suggested Interrupt Vectors

The suggested interrupt vectors for the CIO and DMAC devices take into account that some of the bits of the vectors are shared, e.g., all CIO Port B vectors have five bits which are the same for all interrupt causes.

Each on-card device has interrupt enable and control bits which allow the actual interrupt priority levels to be modified under program control by temporarily disabling certain devices.

Of course, fewer vectors may be used if the devices are programmed not to use modified vectors or if interrupts from some devices are not enabled.

If you want to use the suggested vector numbers in the above table, the proper values to load into the device vector registers are:

Device	Hex Value	Decimal Value
CIO, Port B:	0x40	64
CIO, C/T vector:	0x60	96
DMAC 0	0x70	112
DMAC 1	0x72	114
DMAC 2	0x74	116
DMAC 3	0x76	118

Table 3-4. Device Interrupt Vector Values (Suggested)

Making your way through the Zilog CIO and SCC manuals in search of details on the interrupt logic is quite an experience. We suggest you start with these recommended readings from the CIO and SCC technical manuals:

CIO Z8536 Technical Manual
Vector register: section 2.10.1
Bit priorities: section 3.3.2

SCC Z8530 Technical Manual
Port priorities: section 3.2.2, table 3-5
Vector register: section 4.1.3
Vectors: section 4.1.10, table 4-3

3.4 STATUS LEDs

There are seven status LEDs which continuously show the state of the board as follows:

LED Label	Name	Meaning
F	Fail	The SYSFAIL line is being driven active by this board.
M	Master	The HK68/V30 is the master on the VMEbus.
B	Slave (Bus)	See description below
D	DMA	See description below
H	Halt	See description below
U	User	See description below
S	Supr	See description below

Table 3-5. Status LEDs

3.5 CONTROL PANEL INTERFACE/MPU STATUS

There are five status outputs which allow remote monitoring of the HK68/V30 processor. Connections are made through a 14-pin connector, P5.

P5 pin	Name	Meaning
2	Supr	The MPU is in the supervisor state.
4	User	The MPU is in the user state.
6	DMA	The DMAC has control of the local bus.
8	Halt	The MPU has halted. (Double bus fault, odd stack address or the system reset line or mailbox halt line is active.)
10	Slave	The HK68/V30 is being accessed as a slave on the VMEbus.
1,3,5,7,9	Vcc	Vcc (+5) volts

Table 3-6. Control Panel Interface (P5)

The output signals are low when true. Each is suitable for connection to a LED cathode. An external resistor must be provided for each output to limit current to 15 milliamps.

If both the Supr and the User LEDs are on, the MPU is accessing CPU space (i.e. the floating point coprocessor or interrupt acknowledge).

Two input signals are also provided on P5 for interrupt and reset:

P5 pin	Name	Function
P5-11	INTR*	Connected to CIO bit B7, and pull-up (Refer to section 10.3)
P5-12	Gnd	
P5-13	RESET*	When low, causes a local reset
P5-14	Gnd	

Table 3-7. Control Panel Interface (P5)

A recommended mating connector for P5 is Molex P/N 15-29-8148.

3.6 MPU Cache Control

The 68030 cache may be controlled as follows:

Address	Function (write-only)

00C3,0008	MPU Cache Control
	D0 = 0, cache disabled (default)
	D0 = 1, cache enabled

Table 3-8. MPU Cache Control

The cache control register in the MPU itself must also be set properly to enable the MPU cache and cache burst accesses. See section 6.3 of the Motorola MC68030 User's Manual.

If caching is enabled, data and instructions from on-card ROM and DRAM will be cached. Date and instructions from the VMEbus will also be cached if VMEbus caching is enabled. See section 8.4.3 for further information on VMEbus caching.

3.7 Coprocessors

The HK68/V30 supports the FPP coprocessor, which is described in more detail in section 5.

3.8 Watchdog Timer

The HK68/V30 has two timers which monitor board activity. One is used to monitor on-card activity; the other is for the VMEbus.

If the on-card watchdog timer is enabled and if the on-card address strobe stays on longer than 200 microseconds, the timer will expire. This will cause the current memory cycle to be terminated. The watchdog timer is *disabled* by writing a one to address 00C4,000C. The timer is *enabled* by writing a zero to address 00C4,000C; this is the power-on default state.

Address	Function (write-only)

00C4,000C	Watchdog Timer
	D0 = 0, timer enabled
	D0 = 1, timer disabled

Table 3-9. Watchdog Timer.

See section 6.2 for more details on the watchdog timer.

Chapter 4

DIRECT MEMORY ACCESS CONTROLLER (DMAC)

4.1 INTRODUCTION

The WE32104 or WE32204 DMA Controllers provide a high performance interface between certain 8-bit on-card peripherals and the system address and data buses. The WE32104 and WE32204 have four separate channels, with each channel having its own data buffer (FIFO) to optimize the use of the system and peripheral buses. The net result is better utilization of the system bus which translates to higher system performance. For complete details on the WE32104 and the WE32204, refer to the AT&T WE32104 manual and application notes.

4.2 DMAC REGISTER SUMMARY

The DMA registers are accessed starting at location 0080,0200 according to the following tables:

Channel	Function	Base Address
0	SCSI	0080,0200
1	SCC to/from Memory	0080,0280
2	Ethernet (XE option)	0080,0300
3	SCC to Memory*	0080,0380

*Note: An AT&T Technical Bulletin recommends against using channel 3 of a WE32104 for memory to peripheral transfers.

Table 4-1. DMAC Channel Base Addresses

SCSI DMA transfers must use Channel 0. Likewise, SCC DMA transfers must use channels 1 and/or 3, and Ethernet DMA transfers (XE option) must use channel 2. Any channel may be used for memory to memory or memory fill transfers. For DMA transfers, the VMEbus may be a source, a destination, or both.

Hex Offset	Register	Size (bytes)
000	Source Address	4
004	Destination Address	4
00C	Base Address	4
012	Transfer Count	2
01A	Interrupt Vector	2
022	Status Register	2
026	Mode	2
02A	Device Control	2
400	Memory Fill Data/Data Buffer Register 0	4
404-41C	Data Buffer Registers 1-7	4
213	Mask (Global)	1

Table 4-2. DMA Register Offset Summary

4.3 DEVICE CONTROL REGISTER

The Device Control Registers (DCRs) must be properly programmed for the DMAC device to communicate with SCSI controller, SCC or Ethernet interfaces **even if DMA transfers are not used.** Use the following table:

Channel	Value
0	0x0010
1	0x4000 0x400C (for SCC port A) 0x4008 (for SCC port B)
2	0x8060
3	0x4000 0x400C (for SCC port A) 0x4008 (for SCC port B)

Table 4-3. DMAC Device Control Register (DCR)

DCR 0 must be initialized before the first access of the SCSI chip. DRC 1 and DCR 3 must be initialized before the first access of the SCC chip. DCR2 must be initialized before the first access of the Ethernet interface (when using the XE option).

4.4 DMAC Buffer Flushing

In some circumstances, a DMAC channel's internal buffer will need to be flushed to memory. Although the DMAC itself does not provide a means to do this, the HK68/V30 provides a mechanism for DMAC buffer flushing. Section 12.3 describes the problem and its solution in the context of SCSI data transfers. The DMAC buffer flush addresses are shown in the following table:

Channel	Flush Address	Reference
0	0080,000B	12.3
1	00C3,000A	11.8.2
2	none	
3	00C3,000C	11.8.2

Table 4-4 DMAC Buffer Flush Addresses

4.5 Relevant Jumpers - DMAC

Jumper block J8 connects data transfer request signals from the SCC to the DMAC. See section 11.8.1 for further information.

Chapter 5

FLOATING POINT COPROCESSOR (FPP)

5.1 INTRODUCTION

The HK68/V30 allows the use of an optional MC68881 or MC68882 floating point processor chip. It runs as a coprocessor with the MPU.

Jumper J9 selects the FPP clock source; it is factory set.

5.2 FPP FEATURE SUMMARY

- Allows fully concurrent instruction execution with the main processor.

- Eight general-purpose floating-point data registers, each supporting a full 80-bit extended-precision real data format (a 64-bit mantissa plus a sign bit, and a 15-bit biased exponent).

- A 67-bit ALU to allow very fast calculations, with intermediate precision greater than the extended-precision format.

- A 67-bit barrel shifter for high-speed shifting operations (for normalizing, etc.)

- 46 instruction types, including 35 arithmetic operations.

- Fully conforms to the IEEE P754 standard, including all requirements and suggestions. Also supports functions not defined by the IEEE standard, including a full set of trigonometric and logarithmic functions.

- Supports seven data types: byte, word, and long integers; single, double, and extended-precision real numbers; and packed binary coded decimal string real numbers.

- Efficient mechanisms for procedure calls, context switches, and interrupt handling.

FPP programming details are available in the 68881/68882 technical manual.

5.3 FPP Bypass

The HK68/V30 will operate without the FPP chip. Simply unplug the FPP if it is not required. No wires or jumpers are needed.

If the Watchdog Timer is enabled (via the System Control Latch), the software can determine if the FPP chip is installed. An attempt to access a non-existent FPP will result in a Watchdog time-out and a Bus Error, forcing a Line 1111 MPU Exception, vector number 11.

Chapter 6

SYSTEM ERROR HANDLING

6.1 INTRODUCTION

Numerous events could cause an error to occur. The responses to these events are carefully controlled.

6.2 ERROR CONDITIONS

The following error conditions may arise during MPU or DMAC cycles:

RAM Parity Incorrect parity was detected during a read cycle (by the MPU, DMAC or VMEbus) from on-card RAM memory. This may be due to a true parity error (RAM data changed,) or because the memory location was not initialized prior to the read and it contained garbage. Parity errors generate a *level 7 autovector interrupt.*

A pointer to the parity error handling routine should be loaded at Vector Base Register offset 00007C. Parity checking cannot be disabled.

Watchdog Timeout During an access, usually to the bus, no acknowledge was received within a fixed time interval. This is usually the result of no device being assigned to the specified address.

The timeout interval for an on-card access is 200 microseconds. For a bus access, the interval is approximately 100 microseconds after the access begins if the HK68/V30's VMEbus watchdog timer is used.

For a timeout on an MPU access *to* the bus, the memory cycle is terminated, the *bus error* exception is taken by the MPU and execution resumes at the location specified by the exception vector. A timeout during a DMA cycle will set the DMAC error condition. For either MPU or DMAC bus accesses, the timeout interval is set by the VMEbus watchdog timer, if any.

For an access to the HK68/V30 *from* the bus, a timeout will return a bus error to the bus master. No local *bus error* exception occurs. The timeout interval is set by the VMEbus watchdog timer , and a timeout will occur only if a VMEbus watchdog timer is enabled.

For an MPU or DMAC on-card access, the timeout interval is set by the local bus watchdog timer. As for bus accesses, a timeout will cause a *bus error* exception for an MPU access and set an error condition for a DMAC access.

If an on-card access persists for 1.6 milliseconds and the local bus watchdog timer is enabled, the HK68/V30 will reset. Such an access may be due either to the MPU or DMAC attempting to access an on-card resource, or to an access to the HK68/V30 from the VMEbus (It is possible that the system will also reset. See section 8.3.2 for more information.).

Double Bus Fault

Another bus error occurred during the processing of a previous bus error, address error or reset exception. This error is the result of a major software bug or a hardware malfunction. A typical software bug which could cause this error would be an improperly initialized stack pointer, which points to an invalid address.

A double bus fault forces the MPU to enter the *HALT* state. Processing stops. The HALT status LED will come on. The only way out of this condition is to issue a hardware reset.

MMU Fault

The on-chip, 68030 MMU has detected a write violation or an undefined segment address. The memory cycle is terminated and the *bus error* exception is taken.

Divide by Zero

The value of the divisor for a divide instruction is zero. The instruction is aborted and *vector 5* is used to transfer to an error routine.

Privileged Violation

A program executing in the user state attempted to execute a privileged instruction. The instruction is not executed. Exception *vector 8* is used to transfer control.

Address Error An odd address has been specified for an instruction. The bus cycle is aborted and *vector 3* is used to transfer control.

Illegal Instruction The bit pattern for the fetched instruction is not legal or is unimplemented. The instruction is not executed. Exception *vector 4, 10 or 11* is used to transfer control.

Format Error The format of the stack frame is not correct for an RTE instruction. The instruction is aborted and exception *vector 14* is used to transfer control.

Line 1111 Emulator The FPP Coprocessor is not present and a coprocessor instruction was fetched. The instruction is not executed. Exception *vector 11* will be taken.

FPP Exceptions The FPP Coprocessor has detected a data processing error, such as an overflow or a divide by zero. The FPP causes the MPU to take one of eight exceptions in the range of *48* to *54*.

As the above list indicates, there are two causes for a bus error exception. In order to determine the cause of a bus error exception, test the fault status bits in the MMU. If the MMU indicates no fault then the bus error was caused by the watchdog timer.

Chapter 7

ON-CARD MEMORY CONFIGURATION

7.1 INTRODUCTION

The Heurikon HK68/V30 microcomputer will accommodate a variety of RAM and ROM configurations. The board has 36 ZIP DRAMs, a non-volatile RAM, and one ROM socket which can accommodate EpROM, paged EpROM or EEpROM. Off-card memory may be accessed via the VMEbus.

7.2 ROM

The ROM occupies a fixed one megabyte physical address space. At power-on, the MPU fetches the reset vector from the first eight locations of the ROM (chip position U68). The reset vector specifies the initial program counter and status register values.

Base Address	ROM	Chip
0000,0000	0	U68

Table 7-1. ROM Address Summary

Four jumpers must be set according to the ROM type being used:

EPROM Type	ROM Capacity	----Jumper Positions---- J11	J12	J13	J14
27256	32 Kbytes	B	A	B	A
27512	64 Kbytes	B	A	A	A
27010	128 Kbytes	A	A	A	B
27020	256 Kbytes	A	A	A	B
27040	512 Kbytes	A	A	A	B
27080	1 Mbyte	A	A	A	A
27513 paged	64 Kbytes	B	B	D	A
27011 paged	128 Kbytes	B	B	D	A
1 Mbit paged	256 Kbytes	B	B	D	A
2 Mbit paged	512 Kbytes	B	B	D	A
2864 EEpROM	8 Kbytes	B	B	none	A
28256 EEpROM	32 Kbytes	B	B	C	A

Table 7-2. ROM Capacity and Jumper Positions

See section 17.3 and Fig. 17-1 for help in locating and configuring the jumpers.

The ROM socket is 32 pins. When using a 28-pin device, justify it so socket pins 1, 2, 31 and 32 are empty. Figure 17-1 shows the location of ROM socket pin 1. Twenty four-pin devices are not supported.

The ROM access time must be 250 nanoseconds or less.

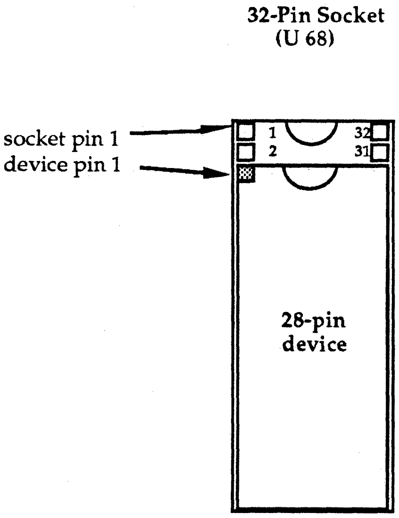


Fig. 7-1. ROM Positioning Diagram.

Electrically Erasable or paged EpROMs may be used. An EEpROM allows specific addresses to be changed by writing to the ROM. When writing to the EEpROM, a delay must be provided *by the software* between write operations. For the 2864, this delay is 10 milliseconds.

Paged EpROMs allow future growth of ROM capacity, without adding address pins. A single device can contain multiple 16K byte pages. A specific page is selected by ***writing*** the page value to the EpROM. For example, to select page three of a 27513, write 0x03 to address 0000,0000. If a paged EpROM has a reset pin, a system reset will select page zero.

7.3 On-Card RAM

The HK68/V30 uses 36 ZIP DRAM packages, each one bit wide. There is one parity bit per byte. Standard memory configurations are four or 16 megabytes. On-card RAM occupies physical addresses starting at 0300,0000.

RAM Type	Quantity	Capacity
1Meg x 1 ZIP	36	4 Megabytes
4 Meg x 1 ZIP	36	16 Megabytes

Table 7-3. On-card RAM Capacity

7.4 On-card Memory Sizing

The following algorithm can be used to determine the amount of on-card RAM memory installed. This procedure takes advantage of "mirrors" which exist in higher addresses when the on-card physical memory size is less than the logical memory space.

1. Clear 16 megabytes of memory starting at location 0300,0000.
2. Write 5555 (hex) to location 0300,0000.
3. Read a word from 0340,0000.

If the value read is 5555, the board has four megabytes of memory installed. If the value is zero, the board has 16 megabytes of memory.

7.5 Bus Memory

See section 8 for details concerning the bus interface.

7.6 Physical Memory Map

See section 17.3 for an I/O device address summary.

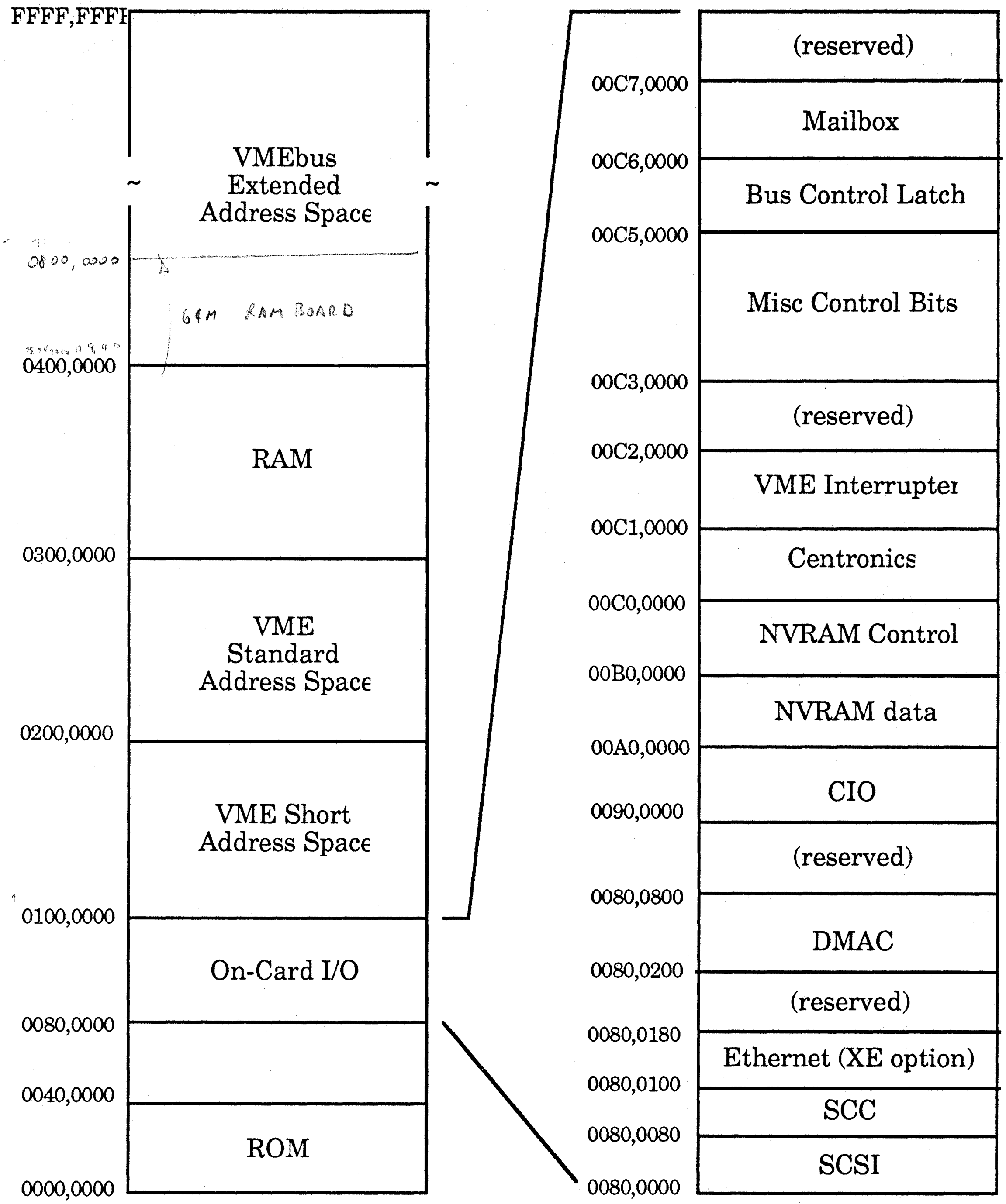


Fig. 7-2. Physical Memory Map

7.7 Memory Timing

Please consult the factory if you need details on memory timing.

The HK68/V30 uses hardware logic to control refreshing of the dynamic memory. The refresh clock runs at is 76,800 Hz. Thus, one row of the RAM array is refreshed every 13 microseconds. Worst case conditions result in a speed penalty of about 1.5% to accommodate the refresh cycles.

Memory timing is controlled by jumper J10, which selects the proper delays for DRAM address multiplexing and RAS/CAS timing. This jumper is factory set; please don't fiddle with it.

7.8 Non-Volatile RAM

A distinctive feature of the HK68 family of products is its non-volatile RAM (NV-RAM), which allows precious data or system configuration information to be stored and recovered across power cycles. The RAM is configured as 256, four-bit words (low half of a byte). When the MPU reads a byte of data from the NV-RAM, the upper four bits of the value it receives are indeterminate. The NV-RAM is accessible as shown below.

Address	Mode	Function
00A0,00xx	R/W	Read/Write RAM contents (4 bits).
00B0,0000	Read	Recall RAM contents from Non-volatile memory.
00B0,0000	Write	Store RAM contents in Non-volatile memory. The 68030 "tas" (test and set) instruction must be used for this operation.

Table 7-4. Non-Volatile RAM Addresses

Physically, the NV-RAM (a Xicor X2212 or equivalent) consists of a static RAM overlaid bit-for-bit with a non-volatile EEpROM. The store operation takes 10 milliseconds to complete. Recall time is approximately one microsecond. Allowances for those delays should be made in *software,* since the memory hardware does not stop the MPU during the store or recall cycles. The chip is rated for 10,000 store cycles, minimum. During a store operation, only those bits which have been changed are "cycled". The use of a "tas" instruction helps prevent an unintentional store operation by an errant program or a power failure glitch.

At power-up, the shadow RAM contents are indeterminate. Do a recall operation before accessing the NV-RAM for the first time. Recall cycles do not affect the device lifetime.

The HK68/V30 monitor (Hbug) and certain system programs use the NV-RAM. The exact amount reserved for Heurikon usage depends on the system. A major portion of the RAM, however, is available for customer use. Heurikon usage is summarized below (details are available separately).

Function
Magic Number
Checksum
Accumulated number of writes
Board type, serial number and revision level
Hardware configuration information
Software configuration information
System configuration information

Table 7-5. NV-RAM Contents (partial)

Chapter 8

VMEbus Control

8.1 Introduction

The control logic for the VMEbus allows numerous bus masters to share the resources on the bus. Up to 21 boards may be used on the VMEbus.

The VMEbus interface uses 32 address lines for a total of 4 gigabytes of VMEbus address space, as well as 32 data lines to support 8, 16, 24 or 32-bit data transfers. The interface also supports "standard" and "short" addressing, using 24 and 16 address lines.

The HK68/V30 has an INTERRUPTER MODULE as well as an INTERRUPT HANDLER, both of which are capable of utilizing any or all of the seven VMEbus interrupt lines.

8.2 Bus Control Signals

8.2.1 VMEbus P1 Descriptions

The following signals on connector P1 and P2 are used for the VMEbus interface. For a complete listing of the pinouts, refer to section 17.

A01-A15 ADDRESS bus (bits 1-15). Three-state driven address lines that are used to broadcast a short address.

A16-A23 ADDRESS bus (bits 16-23). Three-state driven address lines that are used in conjunction with A01-A15 to broadcast a standard address.

A24-A31 ADDRESS bus (bits 24-31). Three-state driven address lines that are used in conjunction with A01-A23 to broadcast an extended address.

ACFAIL* AC FAILURE. An open-collector driven signal which indicates that the AC input to the power supply is no longer being provided or that the required AC input voltage levels are not being met. If jumper J29 is installed, this signal is connected to CIO port C, bit D0, and also MPU interrupt level 7.

AM0-AM5 ADDRESS MODIFIER (bits 0-5). Three-state driven lines that are used to broadcast information such as address size and cycle type. These lines are very similar in usage to the function lines on the MPU.

AS* ADDRESS STROBE. A three-state driven signal that indicates when a valid address has been placed on the address bus.

BBSY* BUS BUSY. An open-collector driven signal low by the current MASTER to indicate that it is using the bus. When the MASTER releases this line, the resultant rising edge causes the ARBITER to sample the bus request lines and grant the bus to the highest priority requester. The HK68/V30 supports early release mode.

BCLR* BUS CLEAR. A totem-pole driven signal, generated by an ARBITER to indicate when there is a higher priority request for the bus. This signal requests the current MASTER to release the bus. This signal is an input and an output of the HK68/V30, associated with J23.

BERR* BUS ERROR. An open-collector driven signal generated by a SLAVE or BUS TIMER. This signal indicates to the MASTER that the data transfer was not completed. This signal can be generated by the HK68/V30 due to a parity error on the HK68/V30 or by the HK68/V30's Bus Watchdog timer, associated with J28.

If BERR* is due to a parity error, the parity error occurred either during the slave access or during a recent MPU or DMAC access.

BG0IN*-BG3IN* BUS GRANT (0-3) IN. Totem-pole driven signals generated by the ARBITER and REQUESTERS. "Bus grant in" and "bus grant out" signals form bus grant daisy chains. The "bus grant in" signal indicates, to the board receiving it, that it may use the bus if it wishes to.

BG0OUT*-BG3OUT* BUS GRANT (0-3) OUT. Totem-pole driven signals generated by REQUESTERS. The bus grant out signal indicates to the next board in the daisy-chain that it may use the bus.

BR0*-BR3* BUS REQUEST (0-3). Open-collector driven signals generated by REQUESTERS. A low level on one of these lines indicates that some MASTER needs to use the bus.

D00-D31 DATA BUS. Three-state driven bidirectional data lines used to transfer data between MASTERS and SLAVES.

DS0*, DS1* DATA STROBE ZERO, ONE. A three-state driven signal used in conjunction with LWORD* and A01 to indicate how many data bytes are being transferred (1, 2, 3, or 4). During a write cycle, the falling edge of the first data strobe indicates that valid data is available on the data bus.

DTACK* DATA TRANSFER ACKNOWLEDGE. An open-collector driven signal generated by a SLAVE. The falling edge of this signal indicates that valid data is available on the data bus during a read cycle, or that data has been accepted from the data bus during a write cycle. The rising edge indicates when the SLAVE has released the data bus at the end of a READ CYCLE.

IACK* INTERRUPT ACKNOWLEDGE. An open-collector or three-state driven signal used by an INTERRUPT HANDLER acknowledging an interrupt request. It is routed, via a backplane signal trace, to the IACKIN* pin of slot one, where it forms the beginning of the IACKIN*, IACKOUT* daisy-chain.

IACKIN* INTERRUPT ACKNOWLEDGE IN. A totem-pole driven signal. The IACKIN* signal indicates to the VMEbus board receiving it that it is allowed to respond to the INTERRUPT ACKNOWLEDGE CYCLE that is in progress if it so wishes.

IACKOUT* INTERRUPT ACKNOWLEDGE OUT. A totem-pole driven signal. The IACKIN* and IACKOUT* signals form a daisy-chain. The IACKOUT* signal is sent by a board to indicate to the next board in the daisy-chain that it is allowed to respond to the INTERRUPT ACKNOWLEDGE CYCLE that is in progress.

IRQ1*-IRQ7* INTERRUPT REQUEST (1-7). Open-collector driven signals, generated by an INTERRUPTER, which carry interrupt requests. When several lines are monitored by a single INTERRUPT HANDLER the highest numbered line is given the highest priority.

LWORD* LONGWORD. A three-state driven signal used in conjunction with DS0*, DS1*, and A01 to select which byte location(s) within the 4 byte group are accessed during the data transfer.

RESERVED RESERVED. A signal line reserved for future VMEbus enhancements. This line is not used.

SERCLK SERIAL CLOCK. A totem-pole driven signal which is used to synchronize the data transmission on the VMEbus. Not implemented on the HK68/V30.

SERDAT* SERIAL DATA. An open-collector driven signal which is used for VMEbus data transmission. Not implemented on the HK68/V30.

SYSCLK SYSTEM CLOCK. A totem-pole driven signal which provides a constant 16 MHz. clock signal that is independent of any other bus timing. This signal is associated with J24.

SYSFAIL* SYSTEM FAIL. An open-collector driven signal that indicates that a failure has occurred in the system. Also used at power-on to indicate that at least one VMEbus board is still in its power-on initialization phase. This signal may be generated by any board on the VMEbus. The state of this signal may be read on CIO Port B, bit D3. A HK68/V30 board reset drives this line low. It is released by writing a one to address 00C4,000E.

SYSRESET* SYSTEM RESET. An open-collector driven signal which, when low, causes the system to be reset. This signal may be either a board input or board output and is associated with jumper J25.

WRITE* WRITE. A three-state driven signal generated by the MASTER to indicate whether the data transfer cycle is a read or a write. A high level indicates a read operation; a low level indicates a write operation.

+5V STDBY +5 Vdc STANDBY. This line supplies +5 Vdc to devices requiring battery backup. Not used on the HK68/V30.

8.3 System Controller Functions

Normally, a VMEbus system requires a System Controller card to provide the system clock, a bus watchdog timer, and a bus arbiter. However, a separate System Controller is *not* needed with an HK68/V30. The HK68/V30 provides the following System Controller functions:

- A 4-level VMEbus arbiter, using either prioritized or round robin arbitration

- A SYSCLK driver

- A SYSRESET driver
- A VMEbus watchdog timer

When the HK68/V30 is acting as a System Controller, it should be installed in VME slot 1.

The following table details the jumper settings which enable or disable the HK68/V30's System Controller functions:

Function	HK68/V30 **is** System Controller	HK68/V30 is **not** System Controller
System Clock (SYSCLK*)	J24-B	J24-A
System Reset (SYSRESET*)	J25-B	J25-A
Bus Clear (BCLR*)	J23 installed	J23 removed
Bus Watchdog	J28 installed	J28 removed
HK68/V30 Bus Req. Level	J20, J21, J22, J26 and J27 see section 8.3.1 and Fig. 17-1.	

Table 8-1. System Controller Functions

8.3.1 Bus Arbitration

There are four separate bus request lines on the VMEbus. Each bus request line has an associated bus grant daisy chain. These lines may be prioritized, or they may be treated with equal importance (round robin), depending on the arbitration mode selected by the System Controller board. The HK68/V30 may be configured for either of these modes as a manufacturing option.

The bus request line that the HK68/V30 uses to request the bus is jumper-selected. Jumpers J21, J22, J26, and J27 must be installed in one of the eight possible configurations shown in Fig. 17-1, depending on which bus request line the HK68/V30 is to use and whether or not it is to be the System Controller. The configurations shown in Fig. 17-1 are the *only* valid arrangements of these jumpers. Be sure that the setting of J20 matches the level selected by jumpers J21, J22, J26, and J27.

Jumper J23 allows the HK68/V30 arbiter to drive BCLR* on the VMEbus. For priority arbitrations, the arbiter will drive BCLR* to indicate when there is a higher priority request for the bus. For round robin arbitration, the arbiter will drive BCLR* when there is any other request for the bus, since any request may be considered to have a higher priority.

8.3.2 System Reset

As shown in Table 8-2, the HK68/V30 may be configured to either drive SYSRESET* or to be reset by SYSRESET*, depending on the setting of jumper J25 (see Fig. 17-1).

SYSRESET* direction	J25 Setting	Result
board input	J25-A	SYSRESET* causes on-card reset, HK68/V30 cannot drive SYSRESET*
board output	J25-B	SYSRESET* will not cause on-card reset. HK68/V30 drives SYSRESET* during any on-card reset.

Table 8-2 SYSRESET Configuration*

8.3.3 VMEbus Watchdog Timer

The HK68/V30 has two timers which monitor board activity. One is used to monitor on-card activity; the other is for the VMEbus (see sections 3.8 and 6.2)

The VMEbus timer is associated only with activity on the VMEbus. It is enabled by installing jumper J28. The timer will expire during a long bus access (either bus Data Strobe asserted for greater than 100 microseconds) by *any* bus master and generate a VMEbus error (BERR). This is normally a VMEbus System Controller function, but it may be enabled whether or not the HK68/V30 is the System Controller.

8.4 Accesses TO the VMEbus (Master Mode)

8.4.1 Bus Addressing

The HK68/V30 supports three VMEbus address modes, "short", "standard" and "extended". Short addresses use the lower 16 VMEbus address lines to specify the target address. Standard addresses use 24 address lines, and extended addresses use all 32 address lines. Table 8-3 details the relationship between the on-card physical address and the corresponding VMEbus regions.

On-Card addresses	VMEbus Region
01xx,0000 thru 01xx,FFFF	VMEbus Short Address (0000 thru FFFF)
0200,0000 thru 02FF,FFFF	VMEbus Standard (00,0000 thru FF,FFFF)
0400,0000 thru FFFF,FFFF	VMEbus Extended (0400,0000 and up)

Table 8-3. VMEbus Regions

Extended VME addresses from 0000,0000 through 03FF,FFFF are not accessible.

The HK68/V30 is prevented from addressing itself on the VMEbus (except for the mailbox).

8.4.2 Bus Requesting and Release

When the MPU or DMAC attempts to access the VMEbus, a VMEbus request is generated. If necessary, the board waits until the bus is available. When the bus is available, the HK68/V30 is granted use of the bus and the access proceeds.

Status LED "M" (master) on the front panel will be lit when the HK68/V30 is the bus master.

When the HK68/V30 is the bus master and the requested bus operation is completed, the bus will be released according to the state of two control bus control signals, BC1 and BC0. They are bits of the Bus Control Latch described below (see Section 8.6.).

BC1	BC0	Bus Release Status
0	0	(Release-On-Request) Release the bus if any other board has a request for the bus (or if BCLR is true). Default at power-up.
0	1	(Release-On-Priority) Release the bus only if BCLR is true. This means release only if a higher priority request is pending.
1	0	(No-Release) Never release the bus, once acquired. This state can be used to capture the bus.
1	1	(Release-When-Done) Release bus after every operation.

Table 8-4. Bus Control Bits

The HK68/V30 supports early release. In early release mode, the board releases BBSY* during a bus cycle, so that bus arbitration may occur while the bus cycle completes.

The VMEbus has four bus request lines. The HK68/V30 must be configured to request the bus on one of these four lines. The bus request line is selected by jumpers J21, J22, J26, J27. These jumpers must be installed in one of eight configurations shown in Fig. 17-1, depending on which bus request line the HK68/V30 is to use and whether or not the HK68/V30 is configured as a System Controller. The configurations of these jumpers shown in Fig. 17-1 are the *only* valid arrangements. Jumper J20 must also be set to match the selected bus request line.

8.4.3 CACHING VMEBUS OPERANDS

The HK68/V30 provides the ability to cache instructions and data read over the VMEbus. To cache operands from the bus:

- Processor caching must be enabled (see section 3.6)
- Bus caching must also be enabled. This is done by setting bit D23 of the Bus Control Latch to 1 (see section 8.6). The default state of this bit is 0.

With bus caching enabled, a cachable access to the bus forces the bus to do a long word read regardless of the size of the operand requested. This requirement is imposed by the 68030. This means that accesses to 8 and 16-bit ports cannot be cached.

An access to the bus is cachable under the following conditions:

1. Bus caching is enabled
2. The access is to standard or extended address space
3. The MMU does not indicate caching of the access should be inhibited.

Short space accesses or those marked as noncacheable by the MMU page tables (i.e., page descriptors with their CI bit bit set) will not be cached by the processor and will not force long word reads on the bus.

If VMEbus standard or extended spaces contain I/O device registers (which should not be cached) or 8-bit or 16-bit ports (which cannot be cached), either bus caching must be disabled, or the MMU must be used to inhibit caching of those portions of the address space.

Bus caching has no effect on DMAC accesses to the VMEbus.

8.5 ACCESSES FROM THE VMEBUS (SLAVE MODE)

A VMEbus slave access to the HK68/V30 is recognized when the most significant VMEbus address lines match the "Slave Compare Address" and the VMEbus address modifier lines match the

"Slave Address Modifier" code. The Slave Compare Address and Slave Address Modifier code are fields in the Bus Control Latch (see section 8.6). Address lines A24-A31 enter the address comparison only for "extended" address modifiers. Table 8-5 lists which address bits participate in address comparison.

The "Slave Compare Address" sets the VMEbus base address of a window into on-card RAM (On the HK68/V30, only RAM is accessible from the bus.). The size of the window is set by jumpers J16 through J19, as described in Table 8-5. (Refer to Fig. 17-1 for jumper locations.) The "Replacement Address" field of the Bus Control Latch specifies the on-card base address of the window visible on the bus. For example, if the window size is set at 1 megabyte, the "Replacement Address" sets which 1 megabyte of on-card RAM is accessible from the bus. The "Slave Compare Address" and "Replacement Address" may each be set at a multiple of the window size.

Slave Window Size	J16	J17	J18	J19	Address Comparison*	RAM Address
1 megabyte	B	B	B	B	(A23-A20)=(CA23-CA20) (std.) (A31-A20)=(CA31-CA20) (ext.)	(RA23-RA20) • (A19-A0)
2 megabytes	B	B	A	B	(A23-A21)=(CA23-CA21) (std.) (A31-A21)=(CA31-CA21) (ext.)	(RA23-RA21) • (A20-A0)
4 megabyte	A	B	A	B	(A23-A22)=(CA23-CA22) (std.) (A31-A22)=(CA31-CA22) (ext.)	(RA23-RA22) • (A21-A0)
8 megabyte	A	B	A	A	A23=CA23 (std.) (A31-A23)=(CA31-CA23) (ext.)	(RA23) • (A22-A0)
16 megabyte	A	A	A	A	true (std.) (A31-A24)=(CA31-CA24) (ext.)	(A23-A0)

*Ax = VMEbus Address Line
CAx = Slave Compare Address bit from Bus Control Latch
RAx = Replacement Address bit from Bus Control Latch

Table 8-5. Slave Mode Address Matching

It may be helpful to understand mapping from a VMEbus address to an on-card RAM address in the following way:

1. Bus address lines A19-A0 become RAM address lines A19-A0.

2. Jumpers J16 through J19 set the window size.

3. The window size determines which of bus address lines A23-A20 participate in the address match.

4. If one of bus address bits A23-A20 does not participate in the address match, then it becomes the corresponding RAM address line.

5. If one of bus address bits A23-A20 does participate in the address match, then it must match the corresponding CAx (Compare Address) bit, and that bus address bit in the RAM address is replaced by the corresponding RAx (Replacement Address) bit.

6. On a 4-megabyte HK68/V30, the 4-megabyte RAM is addressable at each 4-megabyte boundary in the 16-megabyte space.

Once a valid bus request has been detected, an on-card bus request is generated to the MPU and DMAC. When the current MPU or DMAC cycle is completed, the MPU or DMAC will release the on-card bus and the slave access will proceed.

After a slave access, control of the on-card bus will not be returned to the MPU or DMAC for approximately 1.3 microseconds. This mode, called "Release with Hold", improves performance when a series of rapid requests from the VMEbus is expected. However, if the "Release Without Hold" bit in the Bus Control Latch (see section 8.6) is set to 1, the on-card bus will be returned immediately to the MPU or DMAC following a slave access. This mode maximizes performance of the MPU and DMAC at the expense of having more overhead on each slave access.

NOTE: For a slave Read/Modify/Write access of the HK68/V30, the read and write portions are indivisible on-card *only* in "Release with Hold" mode or if the mailbox VMEbus lock is on.

Lengthy memory cycles (tens of microseconds), originating from the bus, should be avoided. RAM refreshing is suspended during an access by the bus of on-card RAM. When DTACK* is received from the HK68/V30 board, the bus master must terminate the bus cycle. If the on-card Watchdog Timer is enabled (see sections 3.8 and 6.2), any access longer than 1.6 milliseconds from the bus will automatically reset the HK68/V30 (and the system if the HK68/V30 drives SYSRESET*).

Status LED L3 will be on when a slave access is in progress.

8.6 Bus Control Latch

A 24-bit Bus Control Latch (at 00C5,0000) is used to specify various parameters concerning the operation of the VMEbus. This is a write-only register. The default state at power-up is all zeros.

Attention Hbug users: do not use the 'sl' command; use 'fl'.

Bit	Function
D23	Bus Caching Enable
D22	VME RMW Cycle Control (RMWCC)
D21	Bus Control BC 1
D20	Bus Control BC 0
D19	VME Slave Release Without Hold
D18	Slave Address Modifier 2
D17	Slave Address Modifier 1
D16	Slave Address Modifier 0
D15	Replacement Address 23
D14	Replacement Address 22
D13	Replacement Address 21
D12	Replacement Address 20
D11	Slave Compare Address 31
D10	Slave Compare Address 30
D9	Slave Compare Address 29
D8	Slave Compare Address 28
D7	Slave Compare Address 27
D6	Slave Compare Address 26
D5	Slave Compare Address 25
D4	Slave Compare Address 24
D3	Slave Compare Address 23
D2	Slave Compare Address 22
D1	Slave Compare Address 21
D0	Slave Compare Address 20

Table 8-6. Bus Control Latch (VME slave logic)

The Slave Compare Address, Replacement Address, and Slave Release Without Hold bits are described in section 8.5.

The Bus Control bits (D21-D20) are described in section 8.4.2.

The Bus Caching Enable bit (D23) is described in section 8.4.3.

The 68030 Read-Modify-Write Cycle control bit (D22) should normally be set to zero. This causes each processor RMW cycle to generate two separate VMEbus cycles without releasing the bus, which is required for proper operation of off-card MMU page tables. (The 68030 asserts its RMC signal on MMU page table accesses.) If true VMEbus RMW cycles are required by your application (i.e., you need to use the 68030 TAS or CAS instructions with target addresses on the VMEbus), then D22 should be set to one and all MMU page tables must be on-card. The VMEbus limits such a RMW cycle to a single bus address. Please contact Customer Service if you need additional information.

The Slave Address Modifier field selects which address modifier codes are allowable for slave access cycles (see section 8.5) according to the following chart:

SAM2	SAM1	SAM0	Slave Address Space
0	0	0	No slave access allowed (slave interface disabled)
0	0	1	Standard Supervisor Data
0	1	0	Standard Data
0	1	1	Standard (all)
1	0	0	Extended Supervisor Data
1	0	1	Extended Data
1	1	0	Extended Supervisor
1	1	1	Extended (all)

Table 8-7. Slave Address Modifiers

8.7 VME Bus Interrupts

The seven VMEbus interrupts are monitored and controlled by the MPU and CIO. A vectored interrupt to the MPU can be generated when a desired bus interrupt signal is on. There are two functions described below. The *Interrupter* generates bus interrupts; the *Interrupt Handler* receives interrupts from the bus.

8.7.1 Interrupter Module Operation

To *generate* a VMEbus interrupt, follow these steps:

1. Decide which of the seven VMEbus interrupt lines you wish to activate. IRQ7* has the highest priority.

2. Disable that level via the CIO so that the INTERRUPT HANDLER does not respond to the interrupt line you are about to use. If you fail to do this, you could interrupt yourself.

3. Write an eight-bit value to the appropriate VME Status/ID latch, as described below. This value is usually treated as a simple interrupt vector, but it could represent other information as well. This value is provided to the board that acknowledges the interrupt when it executes an INTERRUPT ACKNOWLEDGE cycle on the VMEbus with *your* priority level encoded on address lines 1 to 3 (see the Interrupt Handler description which follows.)

   The very act of writing to the Status/ID latch activates the INTERRUPTER circuitry, and the interrupt is generated.

Address	Vector Size	Function (write-only)
00C1,0004	8	Interrupt level 1
00C1,0008	8	Interrupt level 2
00C1,000C	8	Interrupt level 3
00C1,0010	8	Interrupt level 4
00C1,0014	8	Interrupt level 5
00C1,0018	8	Interrupt level 6
00C1,001C	8	Interrupt level 7

Table 8-8. VMEbus Interrupter Addresses

Only one (outgoing) interrupt may be pending at a time. The state of the on-card interrupt logic can be tested by the CIO. The Interrupt Active bit will be true if an interrupt in still pending from this board.

8.7.2 Interrupt Handler Operation

Each bus interrupt generates an interrupt to the MPU at a specific MPU interrupt priority level, as detailed in section 3.2. When an interrupt is recognized, the MPU will execute an interrupt acknowledge cycle on the VMEbus to read the vector from the interrupting board. This vector is used as an index into the MPU vector table.

Any bus interrupt may be disabled via CIO Port A.

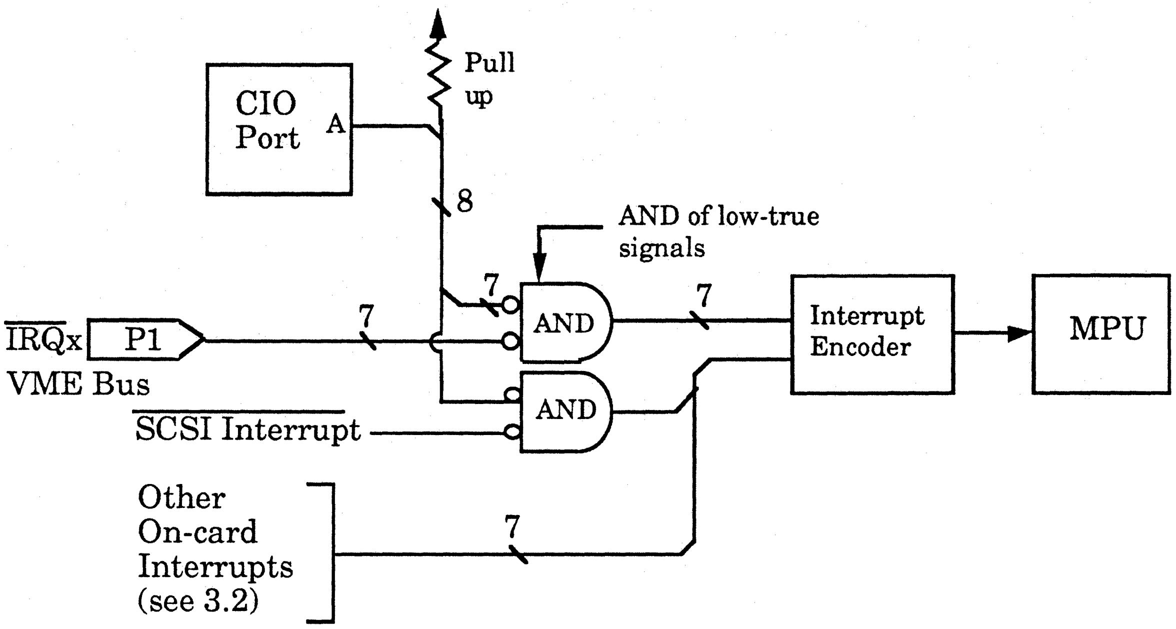


Fig. 8-1. Interrupt Signal Routing

8.8 SYSFAIL Control

The SYSFAIL* line is driven low by the HK68/V30 by a board reset. The SYSFAIL* line will remain low on the VMEbus until all boards release this line after completing their initialization and self test sequences. The SYSFAIL* line also signifies a system failure, and can generate an interrupt via the CIO if it should go on during normal system operation. The current state of this signal may be read via the CIO, see section 10.3.

On the HK68/V30, SYSFAIL* must be released under software control by writing a one to address 00C4,000E. SYSFAIL* may be turned on by writing a zero to that address.

8.9 Mailbox Interface

Certain on-card functions can be controlled via special addresses in the VMEbus *Supervisor Short Address Space,* that is, when the address modifier lines (AM5* to AM0*) are 0x2D. The HK68/V30 will respond (as a slave) to a short address which matches the Mailbox select lines, in the following table.

Address	Function (Slave Mode)
Mbase + 0	MPU Autovectored Interrupt 2 (see 3.2) (Mailbox Interrupt)
Mbase + 2	HK68/V30 Reset
Mbase + 4	VMEbus Lock On
Mbase + 5	VMEbus Lock Off
Mbase + 6	MPU Halt On
Mbase + 7	MPU Halt Off

Table 8-9. Mailbox Functions

The "Mbase" value is specified by 13 Mailbox Base Address bits in the Mailbox Address Latch at address 00C6,0000 (16-bits, write-only). Bus address lines A15 through A3 must match the corresponding data bits in the Mailbox Address Latch. The lower three bits of the latch are not used.

The mailbox logic must be enabled by setting the control bit at address 00C4,0004.

Address	Function (write-only)

00C4,0004	Mailbox Control
	D0 = 0, Disable (default)
	D0 = 1, Enable

00C4,0006	Mailbox Interrupt Reset
	D0 = 0, Reset ON (default)
	D0 = 1, Reset OFF

Table 8-10. Mailbox Control

The Interrupt Reset line must be pulsed by the interrupt service routine (software) to clear the interrupt. Both control bits must be set (1) to enable the mailbox interrupt.

The Lock function, when ON, will deny the MPU or DMAC use of the on-card bus after the *next* access from the bus. The Lock function must be cleared before the MPU or DMAC will be allowed to resume operation. This feature can be used to reduce arbitration time during a block data transfer from the VMEbus.

8.10 Relevant Jumpers - Bus Control

Jumper	Function	Position
J16	VME Slave Window Size	See section 8.5
J17	VME Slave Window Size	See section 8.5
J18	VME Slave Window Size	See section 8.5
J19	VME Slave Window Size	See section 8.5
J20	Bus Request Select	See section 8.3.1 & 8.4.2
J21	BG Daisy Chain 2	See section 8.3.1 & 8.4.2
J22	BG Daisy Chain 0	See section 8.3.1 & 8.4.2
J23	BCLR*	See section 8.3.1
J24	SYSCLK*	See section 8.3
J25	SYSRESET*	See section 8.3
J26	BG Daisy Chain 3	See section 8.3.1 & 8.4.2
J27	BG Daisy Chain 1	See section 8.3.1 & 8.4.2
J28	VME Timeout (BERR)	See section 8.3.3
J29	ACFAIL* Interrupt	See sections 3.2 & 8.2.1

Table 8-11. Bus Control Jumpers

Chapter 9

MISCELLANEOUS DEVICES

9.1 USER LEDs

There are four LEDs (located near the P5 connector and close to the corner of the board) whose meanings may be defined by the program.

LED Number	Address (write-only)
4	00C3,0006
3	00C3,0004
2	00C3,0002
1	00C3,0000

Table 9-1. User LEDs - Addresses

Writing a zero turns the chosen LED on; writing a one will turn it off. At power-on or after a system reset, the LEDs will be ON.

Chapter 10

CIO Usage

10.1 Introduction

The on-card CIO device performs a variety of functions. In addition to the three 16-bit timers that may be used to generate interrupts or count events, the CIO has numerous parallel I/O bits.

The CIO has two independent 8-bit, bidirectional I/O ports (ports A and B) and a 4-bit special-purpose I/O port (port C). Data path polarity (whether bits are inverting or noninverting), data direction (whether bits are input or output), port configuration (bit port or handshake port), ones catchers, and open-drain outputs are programmable for all ports. The configuration and functions of the ports are programmed by means of the port specification registers for each port, which are described fully in the CIO technical manual.

10.2 Port A Bit Definition

Port A handles eight MPU interrupt enable signals. All bits are outputs.

Bit	Function	Port A Data Path Polarity	Interface	Reference
D7	VMEbus IRQ7 enable	Negative True	P1	3.2 & 8.7
D6	VMEbus IRQ6 enable	Negative True	P1	3.2 & 8.7
D5	VMEbus IRQ5 enable	Negative True	P1	3.2 & 8.7
D4	VMEbus IRQ4 enable	Negative True	P1	3.2 & 8.7
D3	VMEbus IRQ3 enable	Negative True	P1	3.2 & 8.7
D2	VMEbus IRQ2 enable	Negative True	P1	3.2 & 8.7
D1	VMEbus IRQ1 enable	Negative True	P1	3.2 & 8.7
D0	SCSI Bus Interrupt enable	Negative True		12

Table 10-1. CIO Port A Bit Definitions

After power-up or reset, the CIO port A pins float and are pulled high, thus placing these interrupt control pins in the "disable" state. Writing a zero to these bits with bit inversion (or a one without inversion) will disable the corresponding interrupt.

10.3 Port B Bit Definition

Port B handles various control signals. All bits are inputs.

Bit	Function	Port B Data Path Polarity	Interface	Reference
D7	External Interrupt	Negative True	P5-11	3.5
D6	SCSI Reset	Negative True	P2-A20	12.2
D5	Serial Ring A	Negative True	P3-14,17(J5)	11.10
D4	Serial Ring B	Negative True	P3-31,34(J6)	11.10
D3	VME SYSFAIL	Negative True	P1-C10	8.8
D2	VME Interrupt in Progress	Negative True		8.7
D1	VME Logic Init Complete	Positive True		
D0	Centronics I/F ACK status	Negative True	P4-19	13.4

Table 10-2. CIO Port B Bit Definitions

Bit D2 may be used to test if there is a pending VMEbus interrupt still active from *this* board.

The "VME Logic Init Complete" bit should be tested prior to the first access of the VMEbus or the Mailbox logic after power-up or reset. This signal will be false for 15 to 40 milliseconds following a power-on reset.

Bit D6 should be used as an input only. Section 12.2 describes driving the SCSI Reset line.

Internal priorities of the CIO place D7 as highest (D0 as lowest) for simultaneous interrupts from this port.

10.4 Port C Bit Definition

Port C on the CIO chip is used to read three on-card status signals.

Bit	Function	Port C Data Path Polarity
D3	VME mailbox lock	Negative true
D2	(not used)	
D1	VMEbus ACFAIL*	Negative true
D0	RAM Parity Error	Negative true

Table 10-3. CIO Port C Bit Definitions

Bits D1 and D0 may be used distinguish between a RAM parity error and an ACFAIL signal, since both signals cause an autovectored level 7 interrupt. Use a one's-catcher for the RAM parity error bit. The interrupt acknowledge will remove the error condition before a service routine could read it directly.

10.5 Counter/Timers

The CIO contains three independent, 16-bit counter/timers. For long delays, timers 1 and 2 may be internally linked together to form a 32-bit counter chain. When programmed as timers, the following equation may be used to determine the time constant value for a particular interrupt rate.

TC = 2,457,600 / interrupt rate (in Hz.)

When the timer is clocked internally, the count rate is 2.4576 MHz. The HK68/V30 board uses a 19.6608 MHz clock oscillator as the system time base. The frequency tolerance specification is +/- 0.01%. If you are using the 19.6608 MHz clock as the CIO time base, the maximum accumulative timing error will be about 9 seconds per day, although the typical error is less than one second per day. Better long-term accuracy may be achieved via a power line (60 Hz.) interrupt, using a bus interrupt or the Real-Time Clock (RTC) option (see section 14).

10.6 Register Address Summary (CIO)

Register	Address	Function
Port C, Data	0090,0000	Miscellaneous Status Bits
Port B, Data	0090,0002	Miscellaneous Status Bits
Port A, Data	0090,0004	Interrupt Enable Bits
Control Regs	0090,0006	CIO Configuration & Control

Table 10-4. CIO Register Addresses

All registers are eight bits wide.

10.7 CIO Initialization

The following figure shows a typical initialization sequence for the CIO. The first byte of each data pair in "ciotable" specifies an internal CIO register; the second byte is the control data. Some control data values need to be changed in the table, based on your application. An active low signal can be inverted (so that a "1" is read from the data port when the signal is true) by initializing the port to invert that particular bit. Refer to section 3.3 for information concerning CIO interrupt vectors.

```
char ciotable[] = {
0x00, 0x01, 0x00,      /* reset, set chip ptr to reg zero  */

                       /* port A initialization            */
0x20, 0x00,            /* bit port, no pattern match       */
0x22, 0xff,            /* invert negative true bits        */
0x23, 0x00,            /* all bits are outputs             */
0x24, 0x00,            /* normal (not open drain)          */
0x25, 0x00,            /* pattern polarity register        */
0x26, 0x00,            /* all levels                       */
0x27, 0x00,            /* pattern mask (no ints)           */

                       /* port B initialization            */
0x28, 0x06,            /* bit port priority encoded vector */
0x2a, 0xfd,            /* invert all but D1                */
0x2b, 0xff,            /* all bits are inputs              */
0x2c, 0x00,            /* normal input (no one's catchers) */
0x2d, 0xff,            /* bit interrupt on a one           */
0x2e, 0x00,            /* no transition, levels only       */
0x2f, 0x00,            /* no interrupts enabled            */
0x03, 0x40,            /* set interrupt vector             */
0x09, 0xc0,            /* set int enable, no int on err    */

                       /* port C initialization            */
0x05, 0x0f,            /* port C bits inverting            */
0x06, 0x0f,            /* port C bits inputs               */
0x07, 0x0f,            /* one's catchers                   */

                       /* timer 3 initialization           */
0x1e, 0x80,            /* set mode to auto reload          */
0x1a, 0xa0,            /* high byte delay constant         */
0x1b, 0x00,            /* low byte delay constant          */
0x04, 0x60,            /* interrupt vector                 */

                       /* enable ports, timer & interrupts */
0x09, 0x20,            /* clear any port B ints            */
0x08, 0x20,            /* clear any port A ints            */
0x01, 0x94,            /* enable timer 3, port a & port b  */
0x0c, 0xc6,            /* set interrupt enable and         */
                       /* gate command bit & trig. cmd bit */
0x00, 0x8e             /* master int enable & vector includes
                        * status for timer 3 and port B    */
};

struct cdevice {       /* CIO register structure           */

    char cdata;        /* port C                           */
    char dummy1;       /* placeholder                      */
    char bdata;        /* port B                           */
    char dummy2;       /* placeholder                      */
    char adata;        /* port A                           */
    char dummy3;       /* placeholder                      */
    char ctrl;         /* control port                     */
};
```

```
#define CIO ((struct cdevice *)0x00900000)
cioinit()
{
    int i, t3intr();

                       /* Don't forget to set all CIO
                        * interrupt vectors. For example:  */

    *(int(*))(0x60*4) = (int)t3intr  /* Timer 3 interrupt */
    i = CIO->ctrl;              /* assure register sync    */
    CIO->ctrl = ciotable[0];    /* avoid a clr instruction */
    i = CIO->ctrl;              /* assure register sync    */
    for (i = 0; i < sizeof(ciotable); i++)
     CIO->ctrl = ciotable[i]; /* send ciotable to CIO chip*/
}

/* note: Aintr() is not used on the V30 */

Aintr()                /* clear Port A interrupt           */
                       /* one of 8 routines                */
{                      /* process port A interrupts here   */
    CIO->ctrl = 0x08; CIO->ctrl = 0x20;
}

Bintr()                /* clear Port B interrupt           */
                       /* one of 8 routines                */
{                      /* process port B interrupts here   */
    CIO->ctrl = 0x09; CIO->ctrl = 0x20;
}

timer3()      /* clear Timer 3 interrupt, arrive via t3intr */

{   /* process timer interrupt here */
    CIO->ctrl = 0x0c; CIO->ctrl = 0x24;
}
```

Fig. 10-1. CIO Program Example (C Portion)

```
           .globl   t3intr%, timer3

# the vector at 0x60*4 points to this routine

t3intr%:   movm.l   &0xFFFF,-(%sp) #save registers
           jsr      timer3         # to C portion
           movm.l   (%sp)+,&0xFFFF # restore registers
           rte
```

Fig. 10-2. CIO Program Example (Assembly Code Portion)

10.8 CIO Programming Hints

1. To maintain compatibility with 68010 programs, do not use the 68030 "clr.b" instruction to set a CIO register to zero. On the 68000 and 68010, that instruction does a "phantom" read of the port before it does the zero write. The read operation will upset the CIO internal register selection sequencer. Similarly, when using a high level language, do not set a CIO register value to the constant "0" because the compiler may use a "clr.b". Use a variable which is set to zero, or output the values from a lookup table.

   For example:

   ```
   zero = 0;
   *CIOcntrl = 0x20;
   *CIOcntrl = zero;
   ```

2. The one's catchers in a CIO port will be cleared whenever any bits are changed in the pattern mask register. Avoid changing the mask register if you are using a one's catcher. If this is not possible, a program that writes to the pattern mask register should first OR the CIO data register into a memory variable. Later, that memory value can be ORed with the CIO data register to find out what the data register would have been if the CIO had not cleared it. Routines which respond to a one's catcher interrupt must clear the corresponding bits in the memory value and the CIO data register. There will still be a critical period where a fast input pulse could be missed, even when using this scheme.

3. If you get an unexpected interrupt from bit D0 of a CIO port, it may be because another enabled CIO input signal went false before the MPU initiated the interrupt acknowledge cycle. The use of a ones catcher may be appropriate to latch the input line.

4. If you turn on a bit in the pattern mask register, that bit will generate an interrupt (if the port is enabled) even if the input signal is false. To prevent this, disable the port while adjusting the pattern mask register.

5. Experience at Heurikon indicates that the CIO outputs glitch when the CIO is reset; all pins momentarily become outputs. On the HK68/V30, this is troublesome because the CIO can momentarily drive the parity error and/or ACFAIL* status lines connected to CIO port C, which are the same signal lines that cause the level 7 parity and ACFAIL* interrupts. The result is typically a spurious interrupt. To minimize or eliminate this problem, disable the CIO ports (CIO register 1) and CIO interrupts (CIO register 0) before resetting the CIO (This is not shown in the example initialization code).

Refer to the Z8536 technical manual for more details on programming the CIO. Some people find the CIO technical manual difficult to understand. We encourage you to read all of it twice, before you pass judgement. Section 3.3 has a list of suggested readings from the CIO manual. Contact us (or Zilog) to obtain application notes.

Chapter 11

SERIAL I/O

11.1 INTRODUCTION

The HK68/V30 board has two RS-232C serial I/O ports. Each port may optionally be configured for RS-422 operation with a special interface cable, as detailed in section 11.9. Each port has a separate baud rate generator and can operate in asynchronous or synchronous modes.

Both serial ports may be wired to the DMAC chip. Refer to sections 4 and 11.8 for more information.

11.2 RS-232 PINOUTS

Data transmission conventions are with respect to the external serial device. The HK68/V30 board is wired as a "Data Set." The connector pinouts are as follows:

Pin	"D" conn. pin	RS-232 Function	SCC Signal	Direction
x	1	Protective ground		n/c
P3- 1	14	x		
P3- 2	2	Transmit Data	RxDA	(from device)
P3- 3	15	Transmit Clock	RTxCA	(from device)
P3- 4	3	Receive Data	TxDA	(to device)
P3- 5	16	x		
P3- 6	4	Request To Send *	DCDA	(from device)
P3- 7	17	Receive Clock	TRxCA	(from device)
P3- 8	5	Clear To Send	DTRA †	(to device)
P3- 9	18	x		
P3-10	6	Data Set Ready	RTSA	(to device)
P3-11	19	x		
P3-12	7	Signal Ground		
P3-13	20	Data Terminal Ready *	CTSA	(from device)
P3-14	8	Data Carrier Detect (J5)	‡	(from device)
P3-15	21	x		
P3-16	9	x		
P3-17	22	Ring Indicator (J5)	‡	(from device)
x	10-13	x		
x	23-25	x		

* Signals have default pullup resistors, controlled by J2.

† Special control bit, not SCC pin.

‡ Ring status bit (CIO port B).

Table 11-1. Serial Port A Pinouts

Pin	"D" Conn. Pin	RS-232 Function	SCC Signal	Direction
x	1	Protective ground		n/c
P3-18	14	x		
P3-19	2	Transmit Data	RxDB	(from device)
P3-20	15	Transmit Clock	RTxCB	(from device)
P3-21	3	Receive Data	TxDB	(to device)
P3-22	16	x		
P3-23	4	Request To Send *	DCDB	(from device)
P3-24	17	Receive Clock	TRxCB	(from device)
P3-25	5	Clear To Send	DTRB †	(to device)
P3-26	18	(+5 via J4, Fuse F1)		
P3-27	6	Data Set Ready	RTSB	(to device)
P3-28	19	x		
P3-29	7	Signal Ground		
P3-30	20	Data Terminal Ready *	CTSB	(from device)
P3-31	8	Data Carrier Detect (J6)	‡	(from device)
P3-32	21	(+12 via J3, Fuse F2)		
P3-33	9	(-12 via J1, Fuse F3)		
P3-34	22	Ring Indicator (J6)	‡	(from device)
x	10-13	x		
x	23-25	x		

* Signals have default pullup resistors, controlled by J2.
† Special control bit, not SCC pin.
‡ Ring status bit (CIO port B).

Table 11-2. Serial Port B Pinouts

Note that the interconnect cable from P3 is arranged in such a manner that the "D" connector pinouts are correct for RS-232C conventions. Not all pins on the "D" connectors are used. Recommended mating connectors are Ansley P/N 609-3401CE and Molex P/N 15-29-8348.

Signals indicated with "*" have default pullup resistors, controlled by J2.

NOTE: The serial ports may *appear* to be inoperative if J2 is set to default "FALSE" and if the device connected to the port does not drive the DTR and RTS pins TRUE. The Hbug monitor software, for example, initializes the SCC channels to respect the state of DTR and RTS. The DCD and RI signals are routed to the CIO by J5 and J6. See sections 10.3, 11.10 and 17.3.

NOTE: The SCC DTR output pins are used as DMA data transfer request signals. Because of this, the real DTR functions for both ports (which appear as "CTS" on P3-8 and P3-25) are controlled by special control bits. See section 11.7.

NOTE: The maximum usable *frequency* of the standard RS-232 drivers and receivers is 160 KHz. Higher frequencies may be accommodated using the RS-422 interface option or by using different RS-232 drivers and receivers through special manufacturing provisions.

11.3 Signal Naming Conventions (RS-232)

Since the RS-232 ports are configured as "data sets," the naming convention for the interface signals may be confusing. The interface signal names are with respect to the terminal device attached to the port while the SCC pins are with respect to the SCC as if it, too, is a terminal device. Thus all signal pairs, e.g., "RTS" & "CTS," get switched between the I/F connector and the SCC chip. For example, "Transmit Data," P3-2, is the data transmitted from the device to the HK68/V30 board; the data appears at the SCC receiver as "Received Data." For the same reason, the "DTR" and "RTS" interface signals appear as the "CTS" and "DSR" bits in the SCC, respectively. If you weren't confused before, any normal person should be by now. Study the chart below and see if that helps.

SCC Signal	I/F Signal	Direction
Tx Data	Rcv Data	to device
Rcv Data	Tx Data	from device
Tx Clock	Rcv Clock	from device
Rcv Clock	Tx Clock	from device
RTS	DSR	to device
CTS	DTR	from device
DTR	CTS	to device
DCD	RTS	from device
	DCD	from device
	Ring Ind.	from device

Table 11-3. Signal Naming Conventions

The SCC was designed to look like a "data terminal" device. Using it as a "data set" creates this nomenclature problem. Of course, if you connect the HK68/V30 board to a modem ("data set"), then the SCC signal names are correct, however, a cable adapter is needed to properly connect to the modem. (Three pairs of signals must be reversed.)

SCC Signal	P3 Pin #s	"D" Pin # at HK68/V30	"D" Pin # at modem	RS-232 Signal
x	x	1	1	Prot Gnd
Rcv Data	2 (or 19)	2	3	Rcv Data
Tx Data	4 (or 21)	3	2	Tx Data
DCD	6 (or 23)	4	6	DSR
RTS	10 (or 27)	6	4	RTS
DTR	8 (or 25)	5	20	DTR
CTS	13 (or 30)	20	5	CTS
	14 (or 31)	8	8	DCD
(Ring Ind)	17 (or 34)	22	22	Ring Ind
(Sig Gnd)	12 (or 29)	7	7	Sig Gnd

Table 11-4. RS-232 Cable Reversal

Summary: The HK68/V30 may be directly connected to a data "terminal" device. A cable reversal is required for a connection to a modem.

11.4 Connector Conventions

Paragraph 3.1 of the EIA RS-232-C standard says the following concerning the mechanical interface between data communications equipment:

> *"The female connector shall be associated with...the data communications equipment... An extension cable with a male connector shall be provided with the data terminal equipment... When additional functions are provided in a separate unit inserted between the data terminal equipment and the data communications equipment, the female connector...shall be associated with the side of this unit which interfaces with the data terminal equipment while the extension cable with the male connector shall be provided on the side which interfaces with the data communications equipment."*

Substituting "modem" for "data communications equipment" and "terminal" for "data terminal equipment" leaves us with the impression that the modem should have a *female* connector and the terminal should have a *male*. The Heurikon HK68/V30 microcomputer interface cables are designed with female "D" connectors, because the serial I/O ports are configured as data sets (modems). Terminal manufacturers typically have a female connector also, despite the fact that they are terminals, not modems. Thus, the extension cable used to run between a terminal and the HK68/V30 (or a modem) will have male connectors at both ends.

If you do any work with RS-232 communications, you will end up with many types of cable adapters. Double males, double females, double males and females with reversal, cables with males and females at both ends, you name it! We will be happy to help make special cables to fit your needs.

11.5 SCC Initialization Sequence

The following table shows a typical initialization sequence for the SCC. This example is for port A. Port B is programmed in the same manner, substituting the correct control port address.

Data	Register Address	Function
00	0080,0087 (write)	Reset SCC register counter
09,C0	" "	Force reset (do for port A)
04,4C	" "	Async mode, x16 clock, 2 stop bits tx
05,EA	" "	Tx: RTS, Enable, 8 data bits
03,E1	" "	Rcv: Enable, 8 data bits
01,00	" "	No Interrupt, Update status
0B,50	" "	No Xtal, Tx & Rcv clk internal
0C,baudL	" "	Set Low half of baud rate constant
0D,baudH	" "	Set high half of baud rate constant
0E,03	" "	Null, BR enable

Table 11-5. SCC Initialization Sequence

Note: the notation "09,C0" (etc.) means the values 09 (hex) and C0 should be sent to the specified SCC port. The first byte selects the internal SCC register; the second byte is the control data. The above sequence only initializes the ports for standard asynchronous I/O without interrupts. The 'baudL' and 'baudH' values refer to the low and high halves of the baud rate constant which may be determined from the Baud Rate Constants section below.

For information concerning SCC interrupt vectors, refer to section 3. Refer to the Z8530 technical manual for more details on SCC programming. Contact us (or Zilog) to obtain application notes.

To maintain compatibility with 68010 programs, do not use the 68030 "clr.b" instruction to set a SCC register to zero. On the 68000 and 68010, that instruction does a "phantom" read of the port before it does the zero write. The read operation will upset the SCC internal register selection sequencer. Similarly, when using a high level language, do not set a SCC register value to the constant "0" because the compiler may use a "clr.b". Use a variable which is set to zero, or output the values from a lookup table. For example, this is correct:

```
zero = 0;
*SCCcntrl = 0x20;
*SCCcntrl = zero;
```

Note: The DMAC device control registers must be initialized *before* the SCC chip is accessed, even if the the DMAC is not used for SCC data transfers.

11.6 Baud Rate Constants

If the internal SCC baud rate generator logic has been selected, the actual baud rate must be specified during the SCC initialization sequence by loading a 16-bit time constant value into each generator. The following table gives the values to use for some common baud rates. Other rates may be generated by applying the formula given in Table 11-6.

Baud Rate	x1 clock rate	x16 clock rate
110	22,340	1,394
300	8,190	510
1200	2,046	126
2400	1,022	62
4800	510	30
9600	254	14
19,200	126	6
38,400	62	2

Table 11-6. Baud Rate Constants

The time constant values listed above are computed as follows:

TC = 4,915,200/(2 * baud * factor) - 2

The x16 mode will obtain better results with asynchronous protocols because the receiver can search for the middle of the start bit. (In fact, the x1 mode will probably produce frequent receiver errors.)

The maximum SCC data speed is one megabit per second, using the x1 clock and synchronous mode. For asynchronous transmission, the maximum practical rate using the x16 clock is 51,200 baud.

11.7 Port Address Summary

Register	Port A	Port B
Control	0080,0087	0080,0083
Data	0080,008F	0080,008B

DTR	00C4,0000
	D0 = 0, DTR off
	D0 = 1, DTR on
		00C4,0002

Table 11-7. SCC Register Addresses

All ports are eight bits. The RS-232 RI (Ring Indicator) or DCD (Data Carrier Detect) signals are read using port B of the CIO; refer to section 10.3.

11.8 DMA Serial Transfer

DMAC channels 1 and 3 may be used for data transfer to and from the SCC. For each SCC port, DMA data transfers may be unidirectional (input or output), half duplex, or full duplex. There are two restrictions on using DMAC channels for data transfer:

1. Full duplex transmission requires two DMA channels. Only one SCC port may use full duplex DMA transfers.

2. AT&T recommends against using DMA channel 3 for memory-to-peripheral (i.e., output) transfers.

11.8.1 DMA Jumpers

Jumper Post	Function
J8-1	SCC Port A Receive Data Request (W/REQA pin)
J8-2	DMAC Channel 3 Request
J8-3	SCC Port B Receive Data Request (W/REQB pin)
J8-4	SCC Port B Transmit Data Request (DTRB pin)
J8-5	DMAC Channel 1 Request
J8-6	SCC Port A Transmit Data Request (DTRA pin)

Table 11-8. SCC DMA Request Jumpers

In order to use either SCC port for full duplex DMA transfer, it is necessary to provide two DMA transfer request signals - one for transmit data and one for receive data. Each SCC port has one dedicated DMA request signal (W/REQ), which is used for receive transfer requests. The port's DTR pin is used for transmit transfer requests (The SCC's DTR functions are performed by special control bits described in section 11.9). Both SCC port pins, W/REQ and DTR, must be programmed as DMA request signals. Sections 4.3.5 and 4.3.6 of the SCC Technical Manual apply to using the SCC for full duplex DMA transfers.

When configuring jumper block J8 for full duplex transfers, the W/REQ should be connected to the DMAC Channel 3 Request and DTR to the Channel 1 Request. This avoids using channel 3 for output transfers.

To use either SCC port for unidirectional or half duplex DMA transfer, it is sufficient to use only one DMA transfer request signal, W/REQ. W/REQ must be programmed as a DMA request signal, according to the type of transfer. Sections 4.3.4 and 4.3.6 of the SCC Technical Manual apply to unidirectional or half duplex transfers using a single DMAC channel.

When configuring jumper block J8 for unidirectional and half duplex transfers, avoid using channel 3 for any output transfers.

11.8.2 DMA Buffer Flushing

If a DMA transfer terminates before the DMAC transfer count reaches zero, the last few data bytes may remain in the DMAC's internal data buffer. When data transfer protocols are used whose packet lengths are not known ahead of time, this situation is quite likely to happen.

This causes no difficulty for output transfers. However, in the case of input transfers, it will be necessary for the DMAC's internal data buffer to be "flushed." A fuller explanation of this problem and its solution in the context of SCSI data transfers is found in section 12.3.

The HK68/V30's DMAC buffers for channels 1 or 3 may be flushed using the following sequence of operations:

1. Set the Halt bit of the affected DMAC channel Status and Control Register to 1.

2. Set the affected DMAC channel Transfer Count Register to 0x0001.

3. Set the Halt bit of the affected DMAC channel Status and Control Register to 0.

4. Write the 8-bit value 0x01 to the DMAC channel flush address of the affected channel (address 00C3,000A for channel 1 or address 00C3,000C for channel 3).

5. Wait until the affected DMAC channel completes (i.e., until the Status and Control Register indicates either a Normal Termination or until it is no longer Active).

6. Write the 8-bit value 00 to the affected DMAC channel flush address (address 00C3,000A for channel 1 or address 0x00C3,000C for channel 3).

This procedure results in one extraneous byte at the end of the data transferred to memory.

11.9 RS-422 Operation

As an option, one or both of the serial ports on the HK68/V30 may be configured for RS-422 operation. These ports are configured for RS-422 operation by removing the on-card RS-232 drivers and receivers and using a special adaptor cable that incorporates the RS-422 interface logic.

NOTE: The information in this section applies to adaptor cables 529A003, 529A004, and 529A005.

11.9.1 Signal Naming Conventions (RS-422)

Signal names for the RS-422 port(s) are consistent with the names of the SCC pins to which they connect. They *do not* follow the DCE naming conventions described in sections 11.3 and 11.4. For example, the signal Send Data connects to SCC pin TxD (Transmit Data) and is a board output.

11.9.2 RS-422 Connector Pinouts

Table 11-9 shows the pinout for each RS-422 connector. Tables 11-1 and 11-2 still apply to RS-232 connectors.

"D" Conn. Pin	Signal Name	SCC Signal	Direction
1	Shield		Signal Ground
14	Ground		Signal Ground
2	N/U		
15	Ground		Signal Ground
3	Ground		Signal Ground
16 4	Send Data+ Send Data-	Transmit Data	Board Output
17 5	Send Timing+ Send Timing-	Transmit Clock	Board Output
18 6	Receive Data+ Receive Data-	Receive Data	Board Input
19 7	RTS+ RTS-	RTS	Board Output
20 8	Receive Timing+ Receive Timing-	Receive Clock	Board Input
21 9	CTS+ CTS-	CTS	Board Input
22 10	N/U N/U		
23 11	N/U N/U		
24 12	DTR+ DTR-	Special control bit*	
25 13	Receive Ready+ Receive Ready-	DCD	Board Input

* Special control bit (not SCC signal). (See section 11.7.)

Table 11-9. RS-422 Connector Pinouts

NOTE: For RS-232 ports, both clock signals are board inputs. For RS-422 ports, Transmit Clock is a board output, while Receive Clock is a board input. The directions of these signals must be taken into account when configuring the SCC.

11.9.3 Jumper Settings

Jumper J4 must be installed to provide +5V to the RS-422 board. Fuse F1 must also be intact.

Jumper J2 does not apply to RS-422 ports. It does still apply to the RS-232 port if only one port is an RS-422 port. Unconnected RS-422 inputs default as follows:

Receive Data	false	(low at SCC)
Receive Timing	false	(high at SCC)
CTS	true	(low at SCC)
DCD	true	(low at SCC)

Table 11-10. RS-422 Input Defaults

The Ring status signal for an RS-422 port is not driven from the RS-422 connector. Instead, the state of the signal is set according to the following table:

	Port A	Port B
CIO port and bit	Port B, bit D5	Port B, bit D4
Jumper setting for signal false (=high)	J5-A	J6-A
Jumper setting for signal true (=low)	J5-B	J6-B

Table 11-11. RS-422 Ring Status Signal

The default jumper settings vary from those in Table 16-2 as follows:

- J2 : Not installed (Both ports A and B are RS-422 ports.)
: J2-B (Only one port is RS-422.)
- J4 : Installed.
- J5 : J5-B
- J6 : J6-B

11.10 Relevant Jumpers and Fuses (Serial I/O)

Jumper	Function	Position
J2	RS-232 Status Default	J2-B (True) J2-A (False)
J5	Port A RI, DCD Select	J5-A (DCD) J5-B (RI)
J6	Port B RI, DCD Select	J6-A (RI) J6-B (DCD)
J7	SCC Clock	factory set

Table 11-12. Signal Jumpers - Serial I/O

See sections 17.3 and Fig. 17-1 for jumper and fuse locations.

Jumper	Fuse	Function
J4	F1	+5v power to P3
J3	F2	+12 power to P3
J1	F3	-12 power to P3

Table 11-13. Power Jumpers and Fuses - Serial I/O

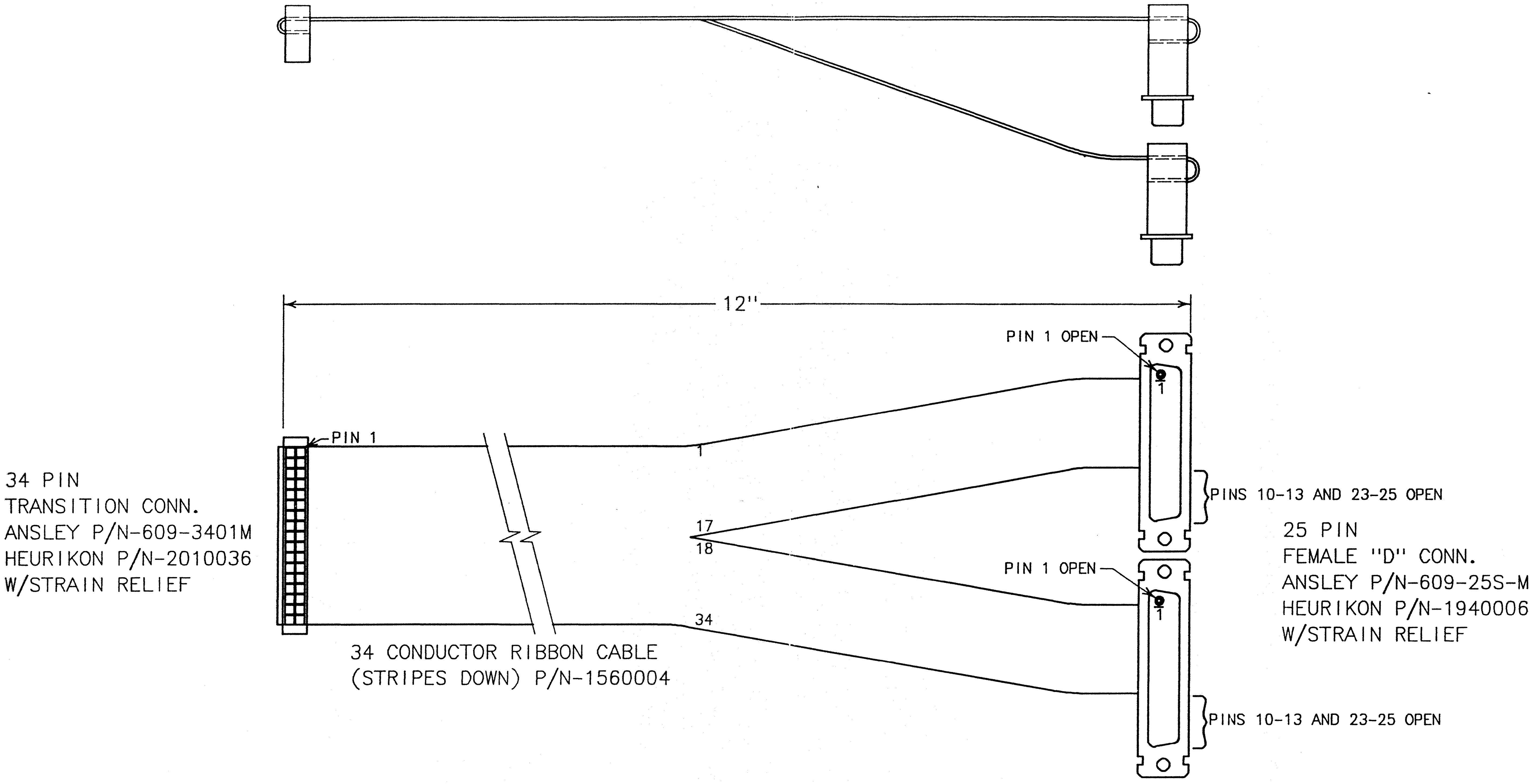


Fig. 11-1. Serial I/O Cable Drawing

DESCRIPTION	REV.	REV.BY	DATE

HEURIKON CORP.		
MADISON, WISCONSIN		
DWN: RJC	CKD:	TOL: ANGLE ±1° 2 PL. DEC.±( ) 3 PL. DEC.±( )
DATE: 7/19/88	SCALE: N.T.S.	
TITLE: V30 RS232 SERIAL INTERFACE CABLE		
SHEET: 1 OF 1	COPYRIGHT: 1988	DWG. NO:529D001

Chapter 12

SCSI Port

12.1 Introduction

The HK68/V30 uses the Western Digital WD33C93 chip to implement a Small Computer System Interface (SCSI) port. (Commonly called "Scuzzy".)

The SCSI port may be used to connect to a variety of peripheral devices. Most common are Winchester disks, floppy diskettes, and streamer tape drives.

Data transfers to and from the SCSI port may use DMA or programmed I/O (i.e. processor MOV instructions). SCSI transfer rates depend upon the characteristics of the attached controller and device being used and whether or not DMA is used.

Supported features and modes include:

Initiator role
Target role
Arbitration
Disconnect
Reconnect
DMA interface

12.2 SCSI Implementation Notes

The WD33C93 interrupt and DMA functions are fully supported on the HK68/V30.

The interrupt from the SCSI chip generates a level 4 autovector. See section 3.2 (MPU Interrupts) for details. The SCSI interrupt is enabled via the CIO port A.

The SCSI bus reset line signal is connected to the CIO (see section 10.3) to allow it to be monitored. When a SCSI reset (input) is detected, the software may reset the SCSI controller using the appropriate WD33C93 command. To generate a SCSI bus reset (output), the SCSI Reset control bit must be pulsed by the software (see below).

Data transfers to and from the SCSI port may use DMA or programmed I/O (i.e. processor MOV instructions).

12.3 SCSI Transfers Using DMA

DMAC channel 0 may be used for data transfers to and from the SCSI Controller. See section 4 for more information about the DMAC.

When using DMA for SCSI data transfers, the SCSI Controller Control Register should have the DMA bit (bit 7) set to a value of 1.

If a DMA transfer terminates before the DMAC transfer count reaches zero (e.g., after a SCSI disconnect), the last few data bytes may remain in the DMAC's internal data buffer.

For output transfers, this causes no difficulty. The DMAC's transfer count register will reflect the number of data bytes actually transferred to the SCSI Controller. A new transfer can start with the first byte not transferred to the SCSI Controller.

In the case of input transfers, the DMAC's transfer count register will reflect the number of bytes received from the SCSI Controller, which includes those bytes which the DMAC has transferred to memory plus those still in the DMAC's buffer. The DMAC itself provides no way to "flush" its own buffer (i.e., to transfer the data in its buffer to memory).

On the HK68/V30, the DMAC's buffer for channel 0 may be flushed in the following fashion:

1. Set the Halt bit of DMAC channel 0's Status and Control Register to 1.

2. Set the DMAC channel 0 Transfer Count Register to 0x0001.

3. Set the Halt bit of DMAC channel 0's Status and Control Register to 0.

4. Write any 8-bit value to the DMAC channel 0 flush address (0080,000B).

Steps 1 through 3 prepare the DMAC channel 0 to transfer one more byte. Step 4 generates a fake data byte transfer which appears to come from the WD33C93 SCSI controller. Following this fake data transfer, the DMAC's channel 0 Transfer Count Register decrements to zero, the transfer completes, and the DMAC writes all the data in its channel 0 data buffer to memory. The fake data transfer results in one extraneous byte at the end of the data transferred to memory.

12.4 Register Address Summary (SCSI)

Address	R/W	Bits	Function
0080,0003	W	8	Set Controller Address Register
0080,0003	R	8	Read Auxiliary Register
0080,0007	R/W	8	SCSI Controller Registers
0080,000B	W	8	DMAC Channel 0 Flush
00C4,0008	W	1	Burst Mode (1=on, 0=off)
00C4,000A	W	1	SCSI Bus Reset (1=reset, 0=release)

Table 12-1. SCSI Register Address Summary

Note: The DMAC device control register for channel 0 must be initialized *before* the SCSI chip is accessed, even if the DMA is not used for SCSI data transfers. Refer to section 4.3.

12.5 SCSI Port Pinouts

The HK68/V30 SCSI interface is via the VMEbus P2 connector. P2 is also used for the upper portion of the VME bus. The pinout of P2 is arranged so a ribbon cable which supports the SCSI bus can be easily attached. The tables below are for the SCSI interface cable, PC, which attaches to P2. The actual P2 pinout can be found in section 17.3.

Pin number	Name	Function	P2 pin
Odd pins except PC-25		Ground	
PC- 2	DB0/	Data bit 0	P2-A 1
PC- 4	DB1/	Data bit 1	P2-A 2
PC- 6	DB2/	Data bit 2	P2-A 3
PC- 8	DB3/	Data bit 3	P2-A 4
PC-10	DB4/	Data bit 4	P2-A 5
PC-12	DB5/	Data bit 5	P2-A 6
PC-14	DB6/	Data bit 6	P2-A 7
PC-16	DB7/	Data bit 7	P2-A 8
PC-18	DBP/	Data parity bit	P2-A 9
PC- 20, 22, 24		Ground	
PC-25		Spare	P2-C13
PC-26	+5	Termination Power (J15)	P2-A13
PC-28,30		Ground	
PC-32	ATN/	Attention	P2-A 16
PC-34		Ground	
PC-36	BSY/	SCSI Bus busy	P2-A 18
PC-38	ACK/	Transfer acknowledge	P2-A 19
PC-40	RST/	Reset	P2-A 20
PC-42	MSG/	Message	P2-A 21
PC-44	SEL/	Select	P2-A 22
PC-46	C/D/	Control/Data	P2-A 23
PC-48	REQ/	Transfer request	P2-A 24
PC-50	I/O/	Data movement direction	P2-A 25

Table 12-2. SCSI Pinouts

Recommended mating connectors are Ansley P/N 609-5001CE and Molex P/N 15-29-8508.

The terminating resistors, RN22, RN23 and RN24 should be installed only if the HK68/V30 is located at an *end* of the SCSI interface cable. Jumper J15 and Fuse F4 supply power for remote termination resistors, if needed.

Chapter 13

CENTRONICS PORT

13.1 INTRODUCTION

This 8-bit parallel port is designed for direct connection to a Centronics compatible (printer) device. Since the handshake lines (STROBE and INIT) are under software control, this interface can be used as a general purpose output port.

13.2 Centronics Port Configuration

P4 Pin	Centronics Pin	HK68/V30 Direction	Signal
P4-(2-24) (even)	(19-30)		Gnd
P4- 1	1	Output	STROBE/
P4- 3	2	Output	DATA1 (D0)
P4- 5	3	Output	DATA2
P4- 7	4	Output	DATA3
P4- 9	5	Output	DATA4
P4-11	6	Output	DATA5
P4-13	7	Output	DATA6
P4-15	8	Output	DATA7
P4-17	9	Output	DATA8 (D7)
P4-19	10	Input	ACK/
P4-21	11	Input	BUSY
P4-23	12	Input	PE
P4-25	13	Input	SELECT
P4-26	31	Output	INIT/
P4-27	14		Gnd
P4-28	32	Input	ERROR/
P4-29	15		n/c
P4-30	33	Input	spare 1
P4-31	16		Gnd
P4-32	34	Input	spare 2
P4-33	17		n/c
P4-34	35	Input	spare 3
	18		n/c
	36		n/c

Table 13-1. Centronics Pinout (Connector P4)

Recommended mating connectors are Ansley P/N 609-3401CE and Molex P/N 15-29-8348.

13.3 Control Port Addresses - Centronics

The Centronics interface logic uses the following physical memory addresses for data and control functions:

Address	Dir	Function
00C0,0000	W	Data Latch (see below)
00C0,0000	R	Status Port (see below)
00C0,0002	W	Turn STROBE on
00C0,0002	R	Turn STROBE off
00C0,0004	W	Turn INIT on
00C0,0004	R	Turn INIT off
00C0,0006	R	Clear Centronics Interrupt

00C0,0006	W	Centronics Interrupt enable
		D0 = 0, disable interrupt (default)
		D0 = 1, enable ACK interrupt

Table 13-2. Centronics Control Addresses

Bit	00C0,0000 (Write) Data Latch	00C0,0000 (read) Status Port
D7	DATA8	(spare 1)
D6	DATA7	(spare 2)
D5	DATA6	(spare 3)
D4	DATA5	ERROR/
D3	DATA4	SELECT
D2	DATA3	PE
D1	DATA2	BUSY
D0	DATA1	ACK/ (Negative true pulse)

Table 13-3. Centronics Data/Status Addresses

After power-on, the state of the Data Latch is indeterminate; STROBE and INIT will be false. Also, the interrupt will be disabled. The Data Latch is not changed by a board reset; however, STROBE and INIT will go false.

The Centronics port interrupt is derived from the ACK/ signal (P4-19). The interrupt turns on any time the interrupt is enabled and ACK/ is active, and stays on until cleared. To clear the interrupt, read from the interrupt clear location (00C0,0006) after the interrupt is disabled or ACK/ is inactive.

13.4 Centronics Port Programming Hints

ACK/ may be a short pulse - short enough that polling the status port may miss seeing the ACK/ active.

In order to use polling in that case, note that ACK/ is connected to CIO port B bit D0. Configure that bit with a one's catcher and poll using that bit instead of the status port.

Follow this procedure when using this port for a Centronics printer:

1. Wait for the printer BUSY signal to go false.

2. Write the character to port 00C0,0000.

3. Write 0x01 to 00C0,0006 to enable the interrupt.

4. Assert STROBE (write to 00C0,0002).

5. Delay at least one microsecond.

6. De-assert STROBE (read from 00C0,0002).

7. Wait for a Centronics port interrupt (or poll ACK/, using a one's catcher on CIO port B bit D0). The ACK/ signal at the Centronics status port (bit D0 of 0C00,0000) will be just a fleeting pulse.

8. Reset the ACK interrupt signal by reading from 00C0,0006.

9. Repeat for the next character.

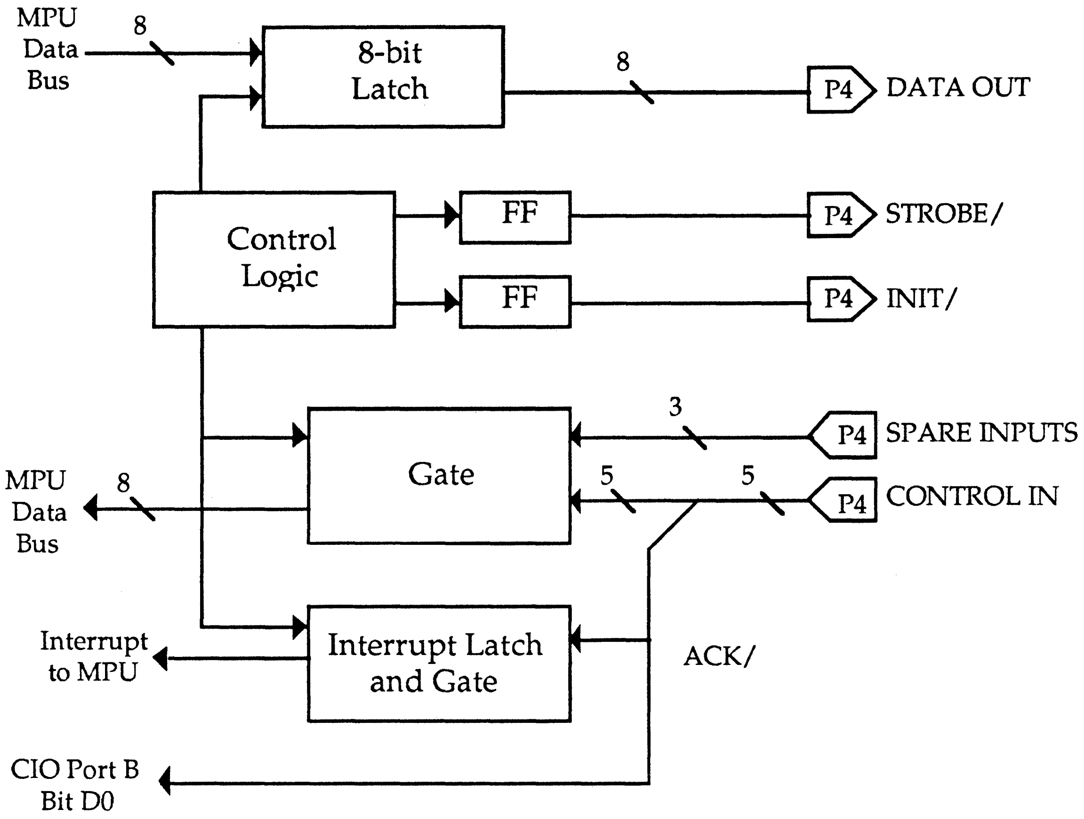


Fig. 13-1. Centronics Interface - Block Diagram

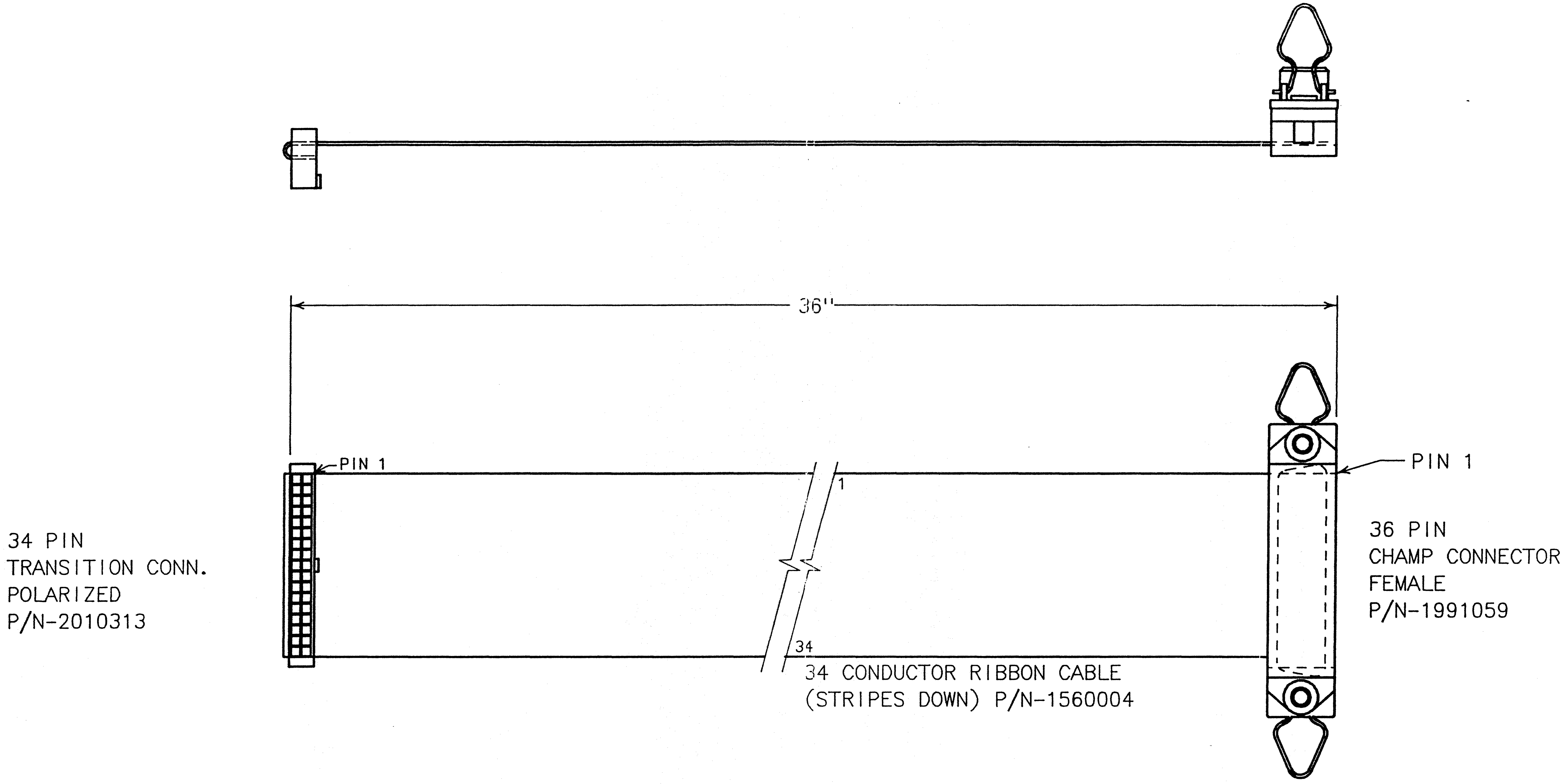


Fig. 13-2. Centronics Printer Interface Cable

CHANGE CONN. P/N'S	A	KES	8/89	
REVISIONS	REV	REV BY	DATE	ECO/APPVL

Copyright Heurikon Corporation. All rights reserved. No part of this document may be reproduced, stored in retrieval systems, or transmitted, in any form or by any means - mechanical, electronic, photocopying, duplicating, microfilming, videotaping or otherwise - without the prior written permission of Heurikon.

HEURIKON CORP.

MADISON, WISCONSIN

DWN: KES	CKD:	TOL: ANGLE ±1° 2 PL. DEC.±( ) 3 PL. DEC.±( )
DATE:8/15/89	SCALE: N.T.S.	
TITLE: V30 CENTRONICS I/F CABLE		
SHEET:1 OF 1	COPYRIGHT: 1989	DWG. NO:529A002-A

Chapter 14

REAL-TIME CLOCK (RTC) - OPTIONAL FEATURE

14.1 INTRODUCTION

As an option, the normal pROM socket (U68) can be replaced with a special Real-Time Clock (RTC) socket which has a built-in CMOS watch circuit and a lithium battery (Dallas Semiconductor, part number DS1216F).

The RTC logic does not generate interrupts; a CIO timer channel is still used for that purpose. The RTC contents, however, may be used to check for long-term drift of the HK68/V30 system clock, and as an absolute time and date reference after a power failure. Leap year accounting is included. Heurikon can provide complete operating system software support for the RTC module.

The RTC module time resolution is 10 milliseconds. The RTC internal oscillator is accurate to one minute per month, at 25 degrees C.

The clock contents are set or read using a special sequence of ROM read commands, as detailed in the program example, below. The RTC module "monitors" ROM accesses and, if a certain sequence of 64 ROM addresses occur, takes temporary control of the ROM space, allowing data to be read from or written to the module. Writing is done by twiddling an address line, which the module uses as a data input bit. There are never any MPU write cycles directed to the pROM space.

Do not execute the module access instructions out of ROM. The instruction fetch cycles will interfere with the module access sequence. Also, be certain the reset disable bit (rtc_data.day bit D4) is always written as a "1".

```
#define WATCHBASE (unsigned char *)0x00000000          /* ROM socket */
#define WR0_WATCH (unsigned char *)(WATCHBASE+2)       /* write 0    */
#define WR1_WATCH (unsigned char *)(WATCHBASE+3)       /* write 1    */
#define RD_WATCH  (unsigned char *)(WATCHBASE+4)       /* read       */
struct rtc_data {         /* D7 D6 D5 D4   D3 D2 D1 D0   range       */
    unsigned char dotsec; /* --0.1 sec-- : --0.01 sec- ; 00-99       */
    unsigned char sec;    /* --10 sec--- : --seconds-- ; 00-59       */
    unsigned char min;    /* --10 min--- : --minutes-- ; 00-59       */
    unsigned char hour;   /*  A  0  B Hr : --hours---- ; 00-23       */
    unsigned char day;    /*  0  0  0  1 : --day------ ; 01-07       */
    unsigned char date;   /* --10 date-- : --date----- ; 01-31       */
    unsigned char month;  /* --10 month- : --month---- ; 01-12       */
    unsigned char year;   /* --10 year-- : --year----- ; 00-99       */

}; /* "A" =  "0" for 00-23 hour mode, "1" for 01-12 hour mode        */
   /* "B" =  MSB of the 10 hours value (if 00-23 hour mode) else
           =  "0" for PM or "1" for AM  (if 01-12 hour mode)         */
rtc_wr(data)                            /* set the real-time clock
                 */
register unsigned char *data;           /* rtc_data pointer          */
{
    register int i, bit;
    unsigned char temp;
    static unsigned char key[] = {       /* the unlock pattern        */
        0xC5, 0x3A, 0xA3, 0x5C, 0xC5, 0x3A, 0xA3, 0x5C };
    if ( data ) {
        rtc_wr(0);                      /* send key pattern          */
    } else {              /* this is the unlock function             */
        i = *RD_WATCH;                  /* reset                     */
        data = key;
    }
    for( i=0; i<8; data++, i++ )
        for( bit = 1; bit & 0xff; bit <<= 1 )
            temp = ( *data & bit ) ? *WR1_WATCH : *WR0_WATCH;
}
rtc_rd(data)                            /* read the real-time clock  */
register unsigned char *data;           /* rtc_data pointer          */
{
    register int i, bit;
    rtc_wr(0);                          /* send key pattern          */
    for( i=0; i<8; data++, i++ ) {
        *data = 0;
        for( bit = 1; bit & 0xff; bit <<= 1 )
            *data |= (*RD_WATCH & 1) ? bit : 0 ;
    }
}
```

Fig. 14-1. Real-Time Clock, Example Software

Chapter 15

ETHERNET SUPPORT — THE XE MODULE (OPTIONAL)

15.1 INTRODUCTION

Heurikon provides support for Ethernet or Thin Ethernet on the HK68/V30 board in the form of an optional XE module. The XE module is implemented as a single-height VME card (3U) that plugs onto an HK68/V30 and into a P2 VMEbus connector. Equipped with the optional XE module, the HK68/V30 becomes the HK68/V30XE.

An HK68/V30XE uses 1.5 VME slots. Heurikon offers an optional front panel for the V30XE to permit the installation of another single-height VME card (such as a 4-channel serial card) on the P1 side to permit use of the remaining 1/2 VME slot.

The XE Ethernet/Thin-Ethernet module (3U) mounts to the component side of the V30 on the VMEbus P2 side. A PGA header connects to the DMAC (WE32104/WE32204) socket to provide the electrical interface to the V30.

The optional XE module contains:

- 32-Kbyte static RAM for packet buffering
- 32-byte ROM for the Ethernet Physical Node address
- National Semiconductor 8390 Network Interface Controller (NIC)
- National Semiconductor 8391 Serial Network Interface (SNI)
- National Semiconductor 8392 Coaxial Transceiver Interface (CTI)
- AT&T WE32104 4-channel DMA Controller
- 4 status LEDs

Although the XE module gets power/ground from the VMEbus P2, the board is not logically "on" the VMEbus. The XE module is logically located on-card on the peripheral side of the HK68/V30's DMAC, leaving the HK68/V30's VMEbus operations unhindered.

15.2 Block Diagram

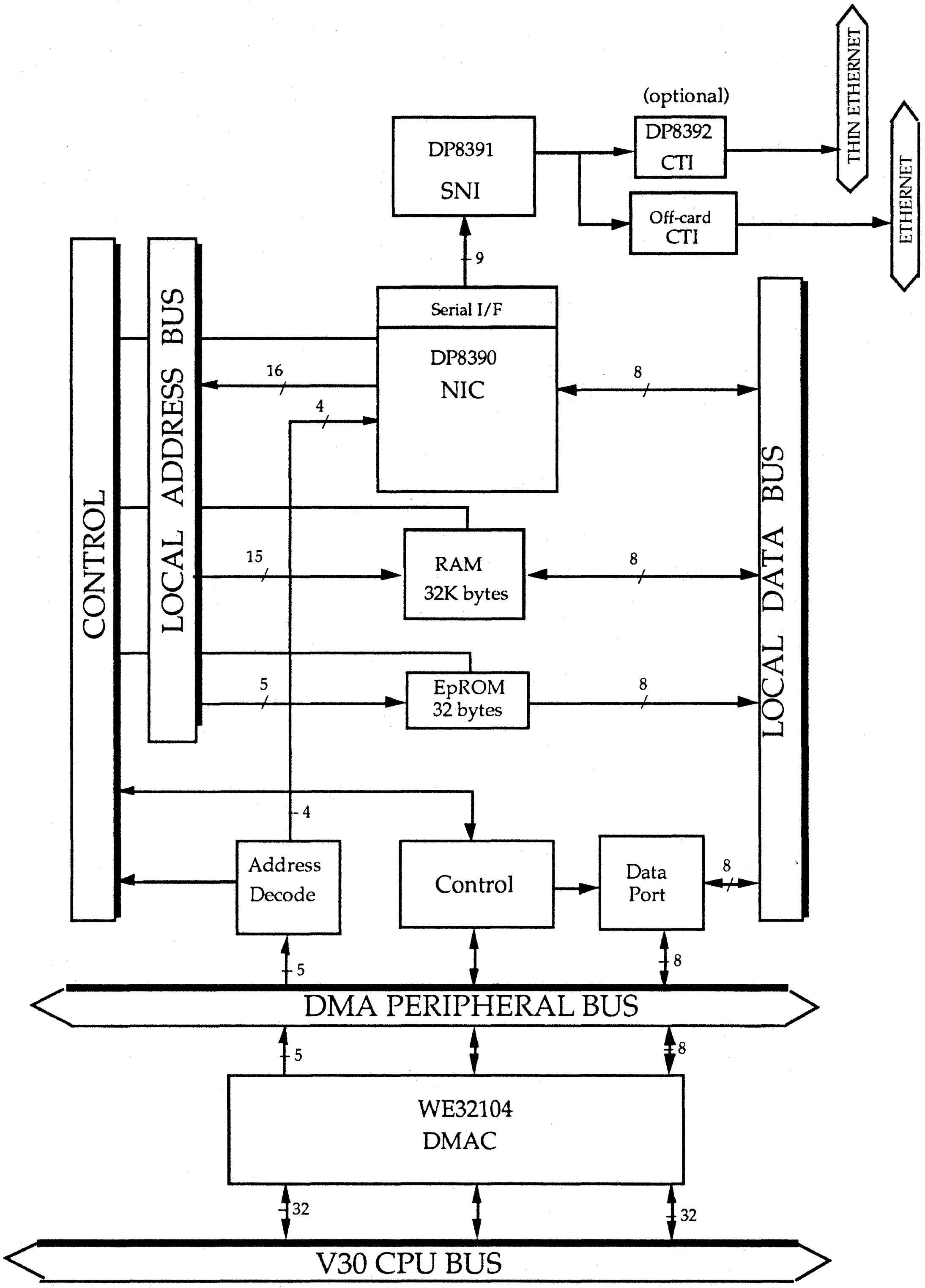


Fig. 15-1. XE Module Block Diagram

15.3 V30XE Components

15.3.1 Network Interface Controller (NIC)

The Network Interface Controller for the HK68/V30XE is the National Semiconductor DP8390, which implements all Media Access Control (MAC) layer functions for transmission and reception of packets in accordance with the 10 Megabit/second IEEE802.3 standard. It implements the CSMA/CD protocol, scheduling retransmission of packets up to 15 times on collisions according to the truncated binary exponential backoff algorithm. The Network Interface Controller supports physical, multicast, and broadcast addressing, and runs at a frequency of 20 MHz. The NIC also includes:

- Local and remote bidirectional DMA channels
- A 16-byte internal FIFO queue with programmable threshold
- Network statistics storage capabilities

The Network Interface Controller supports three forms of loopback for diagnostic purposes:

1. The HK68/V30XE may be run in loopback through the deserializer on the Network Interface Controller itself.
2. The HK68/V30XE may be run in loopback through the Serial Network Interface.
3. The HK68/V30XE may use a coaxial cable loopback to check and verify the link through the transceiver circuitry.

15.3.2 Serial Network Interface (SNI)

The Serial Network Interface for the HK68/V30XE is the National Semiconductor DP8391. The SNI is an IEEE802.3-compatible device that handles Manchester data encoding and decoding to and from the network and serial data transfer in Non-Return-to-Zero (NRZ) form to the Network Interface Controller.

The features of the Serial Network Interface include the following:

- 10 Megabit/second Manchester encoding and decoding with receive clock recovery
- Decodes Manchester data with up to 18 nanoseconds of jitter

- Loopback capability for diagnostics
- Externally selectable half- or full-step modes of operation at transmitter

15.3.3 Memory

The XE module includes 32 Kbytes of Static RAM for local packet storage and 32 bytes of ROM for on-board network address storage.

15.3.4 DMA

The HK68/V30XE uses the same 32-bit host DMA (DMA channel 2) as the V30. When the XE module is installed on the HK68/V30, the DMAC chip (AT&T WE32104/32204) is physically relocated to the XE plugover module.

15.3.5 Status LEDs

The XE module provides four status LEDs (shown in Figure 15-2) to allow visual monitoring of the following board and network activities:

1. Packet Transmission (labelled **T** in Figure 15-2).
2. Network Activity — when packets are being scanned (labelled **R** in Figure 15-2)
3. Loopback Mode Indicator (labelled **L** in Figure 15-2)
4. User Programmable (labelled **U** in Figure 15-2)

15.3.6 Thin Ethernet

The V30XE supports both Ethernet and Thin Ethernet. Thin Ethernet uses a transceiver on the V30XE, whereas Ethernet uses an off-card transceiver.

The National Semiconductor DP8392 Coaxial Transceiver Interface (CTI) provides the Thin Ethernet interface for the HK68/V30XE. The HK68/V30XE's Thin Ethernet includes both signal isolation pulse transformers and a power isolation DC-DC converter (+5 to –9 volts) to constitute an IEEE802.3-compatible transceiver.

NOTE: J1 must be in the A position (half-step mode) for Thin Ethernet configuration, because the V30XE's Thin Ethernet transceiver is IEEE802.3 compatible.

15.4 V30XE Configuration

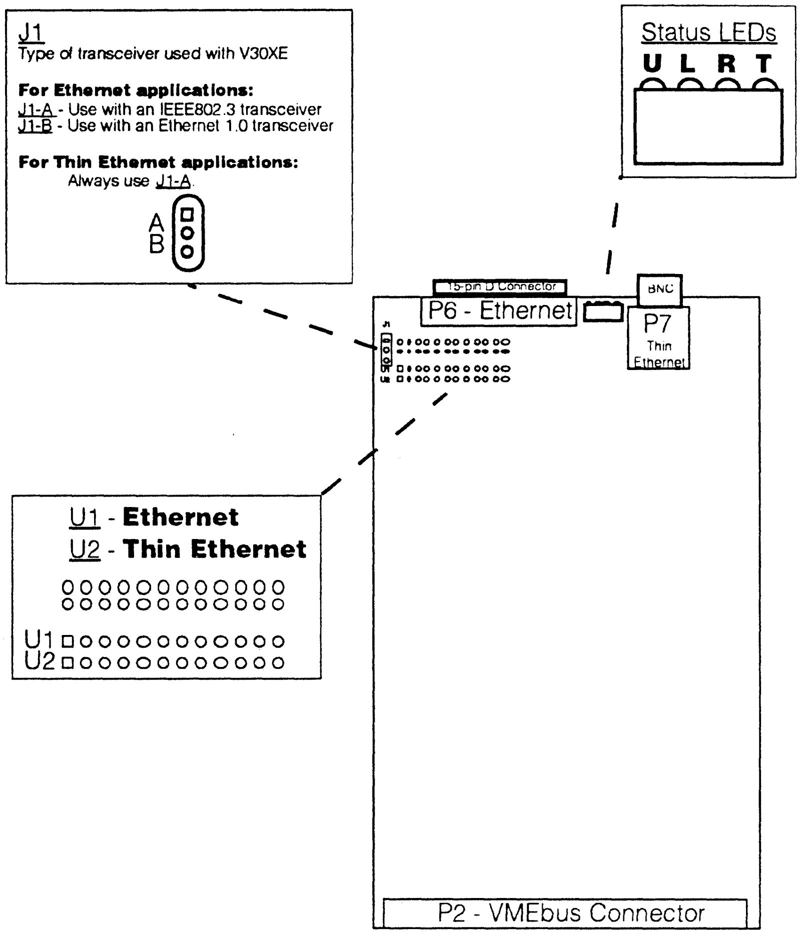

J1
Type of transceiver used with V30XE
For Ethernet applications:
J1-A - Use with an IEEE802.3 transceiver
J1-B - Use with an Ethernet 1.0 transceiver
For Thin Ethernet applications:
Always use J1-A.
A
B
Status LEDs
U L R T
15-pin D Connector
P6 - Ethernet
BNC
P7
Thin
Ethernet
U1 - Ethernet
U2 - Thin Ethernet
U1
U2
P2 - VMEbus Connector

Figure 15-2. Status LEDs and Jumper Configurations for Ethernet and Thin Ethernet

The XE module has two configurable modes of operation that are user-selectable:

- The transmit differential line for the XE module may be configured for either half- or full-step modes to facilitate its use with different types of transceivers via configuration jumper J1.
- The XE module may be configured to select the use of either standard Ethernet or Thin Ethernet (if the board is equipped for this option). This configuration is dependent on the location at which the 8391 Serial Network Interface (SNI) chip is installed.

The board jumper configuration diagram is shown in in Fig. 15-2. XE MODULE configuration is briefly summarized in the following tables.

Transmit differential line configuration (Jumper J1)	
Position	Configuration
J1-A	0 differential voltage on TX lines half-step mode IEEE802.3 transceiver *and* Thin Ethernet
J1-B	+ (positive) differential voltage on TX lines full-step mode Ethernet 1.0 transceiver

Table 15-1. Transmit Differential Line Configuration (J1)

SNI installed in	Configuration
U1	Ethernet (uses 15-pin D connector)
U2	Thin Ethernet (uses the BNC connector)

Table 15-2. Ethernet Selection Configuration (Locations U1/U2)

15.5 XE MODULE OPERATION

The Network Interface Controller and various XE module control functions are accessible via the host DMAC's (WE32104) peripheral bus. All operations to and from local-packet-memory/ROM and to and from HK68/V30 main memory occur via the Network Interface Controller's DMA channels and host DMAC channels.

Slave Operations

Slave accesses to the XE module from the HK68/V30 host board access the host DMAC's registers, the Network Interface Controller registers, or XE module control functions. The relationship between 68030 addresses and the XE module addresses is detailed in Table 15-3.

V30 Address	PA4	PA3	PA2	PA1	PA0	R/W	Function
0080,0103	0	0	0	0	0	R/W	NIC register 0x0
0080,0107	0	0	0	0	1	R/W	NIC register 0x1
0080,010B	0	0	0	1	0	R/W	NIC register 0x2
0080,010F	0	0	0	1	1	R/W	NIC register 0x3
0080,0113	0	0	1	0	0	R/W	NIC register 0x4
0080,0117	0	0	1	0	1	R/W	NIC register 0x5
0080,011B	0	0	1	1	0	R/W	NIC register 0x6
0080,011F	0	0	1	1	1	R/W	NIC register 0x7
0080,0123	0	1	0	0	0	R/W	NIC register 0x8
0080,0127	0	1	0	0	1	R/W	NIC register 0x9
0080,012B	0	1	0	1	0	R/W	NIC register 0xA
0080,012F	0	1	0	1	1	R/W	NIC register 0xB
0080,0133	0	1	1	0	0	R/W	NIC register 0xC
0080,0137	0	1	1	0	1	R/W	NIC register 0xD
0080,013B	0	1	1	1	0	R/W	NIC register 0xE
0080,013F	0	1	1	1	1	R/W	NIC register 0xF
0080,0143	1	0	0	0	0	R/W	Reserved
0080,0147	1	0	0	0	1	R/W	Reserved
0080,014B	1	0	0	1	0	R/W	Reserved
0080,014F	1	0	0	1	1	R/W	Reserved
0080,0153	1	0	1	0	0	R/W	Reserved
0080,0157	1	0	1	0	1	R/W	Reserved
0080,015B	1	0	1	1	0	R/W	Reserved
0080,015F	1	0	1	1	1	R/W	Reserved
0080,0163	1	1	0	0	0	R/W	DACK
0080,0167	1	1	0	0	1	R/W	DACK
0080,016B	1	1	0	1	0	R/W	DACK
0080,016F	1	1	0	1	1	R/W	DACK
0080,0173	1	1	1	0	0	R/W	Software reset on
0080,0177	1	1	1	0	1	R/W	Software reset off
0080,017B	1	1	1	1	0	R/W	User LED on
0080,017F	1	1	1	1	1	R/W	User LED off

Table 15-3. V30 Address/Slave Accesses

Master Operations

When the Network Interface Controller has control of the XE module's local bus during data movement, it functions as the master. The NIC functions as master during the following types of NIC transactions:

Local DMA	*from* network	*to* XE RAM	*during*	packet reception
Local DMA	*from* XE RAM	*to* network	*during*	packet transmission
Remote DMA	*from* XE RAM	*to* host RAM	*following*	packet reception
Remote DMA	*from* host RAM	*to* XE RAM	*preceding*	packet transmission

Once configured, the NIC handles all Local DMA operations during reception and transmission. Remote DMA operations are initiated by the CPU and work in conjunction with the host DMAC (WE32104) to transfer packets between host (HK68/V30) memory and the XE module's buffer RAM.

If a Remote DMA transfer is initiated or in progress when a packet is being transmitted or received on the network (that is, during Local DMA), the Remote DMA transfer will automatically be interrupted for the higher priority Local DMA transfers. When the Local transfer is completed, the Remote DMA will automatically resume.

When in the master mode, the Network Interface Controller accesses XE module ROM and RAM at the following addresses (the area between ROM and RAM mirrors the ROM):

Description	Address	Size
ROM	0000-001FH	32 bytes
ROM Mirrors	0020-7FFFH	~32 Kbytes
RAM	8000-FFFFH	32K bytes

15.6 XE Module Pinouts

15.6.1 P2 Connector Pinouts

The following are the *only* P2 connections:

Pin Number	Function
B-1	VCC
B-2	GND
B-12	GND
B-13	VCC
B-22	GND
B-31	GND
B-32	VCC

Table 15-4. P2 (VMEbus - Power and Ground)

15.6.2 P6 Connector Pinouts

Connector P6 is an Ethernet 15-pin D connector.

Pin number	Name	Function
1	CIS	Control In Shield
2	CI+	Control In +
3	DO+	Data Out +
4	DIS	Data In Shield
5	DI+	Data In +
6	VC	Voltage Common
7	CO+	Control Out +
8	COS	Control Out Shield
9	CI-	Control In -
10	DO-	Data Out -
11	DOS	Data Out Shield
12	DI-	Data In -
13	VP	Voltage Plus
14	VS	Voltage Shield
15	CO-	Control Out -

Table 15-5. P6 Connector Pinouts (Ethernet)

15.6.3 P7 Connector Pinouts

Connector P7 is a Thin Ethernet BNC connector.

Pin number	Function
1	Center Conductor: Isolated Signal
2	Shield: Isolated Ground

Table 15-6. P7 Connector Pinouts (Thin Ethernet)

15.7 Electrical

The signal interface of the host V30 board to the XE module is through the DMAC chip socket. This interface is called the KBUS.

Both a +5v and GND are provided to the XE module via a VME P2 connector. The +5v and GND pins of this P2 connector *must* be connected to ensure proper operation.

The fused +12v, 500ma required for the Ethernet is provided by four KBUS pins.

An on-board DC-DC convertor provides the -9v for the Thin Ethernet feature.

15.8 Mechanical

The HK68/V30XE's piggyback design permits its implementation as a single-height VME board. Power and Ground for the HK68/V30XE board are provided by the on-board VME P2 connector. The XE module's KBUS piggyback connection to the HK68/V30 host board is implemented via the use of a 155-pin, 13x13 grid PGA that uses a custom PGA adaptor.

The XE module is supported at the front by standoffs attached to the front panel and the HK68/V30's Centronics connector. There are two additional standoffs attached at the P2 connector.

Two different front panel configurations are available for the HK68/V30XE:

1. A double-height front panel extended across the entire front width of the HK68/V30XE.

2. A double-height panel notched out to allow the installation of another single-height (3U) VME board on the VME P1 connector side of the board's front panel.

15.9 Programming Examples

The following C language programming examples demonstrate typical operations of the XE module. These are only examples, and may change, depending on the application.

They are provided to show the logic of common operations.

Included are the following:

#define.... C language declarations which describe a typical XE application.

xe_transmit( ) The xe_transmit( ) function demonstrates a low-level routine to transmit a packet onto a network.

xe_set_physaddr( ) This function fetches the Ethernet address from the XE module's ROM and initializes the NIC with it.

RAMRead( ) Demonstrates XE RAM accesses and DMA operations.

init( ) Demonstrates XE Reset, XE LED, and a typical initialization.

```
#define DMA_BASE      0x800000                  /* DMA Base Address                  */
#define DMA_DCR2      0x8060                    /* DMA Device Control register       */
#define DMA_MR2       0xA080                    /* DMA Mode register                 */
#define DMA_MASK2     0x000B                    /* DMA Mask Register                 */

#define XEBASE        0x00800100                /* V30 Base addr of XE               */
#define NIC ((struct device *) XEBASE)          /* V30 Base addr of NIC regs         */

#define XE_RAM_BOT    0x8000                    /* NIC Base addr of XE RAM           */
#define XE_RAM_TOP    0xFFFF                    /* Top  addr of XE RAM               */
#define XE_RAMSIZ     (XE_RAM_TOP-XE_RAM_BOT)   /* Size of XE RAM                    */

#define XE_ROM_BOT    0x0000   /* Base address of XE ROM for NIC local accesses      */
#define XE_ROM_TOP    0x001F   /* Top  address of XE ROM for NIC local accesses      */
#define XE_ROMSIZ     (XE_ROM_TOP - XE_ROM_BOT)           /* Size of XE ROM          */

#define NIC_TX_ADDR   0xf800   /* NIC transmit address and local tx buffer           */

#define NIC_RST_ON    ((char *) XEBASE + 0x73)  /* V30 addr for XE reset ON          */
#define NIC_RST_OFF   ((char *) XEBASE + 0x77)  /* V30 addr for XE reset OFF         */
#define LED_ON        ((char *) XEBASE + 0x7b)  /* V30 addr for XE LED ON            */
#define LED_OFF       ((char *) XEBASE + 0x7f)  /* V30 addr for XE LED OFF           */

#define PSTART        (XERAM_BOT >> 8)          /* Ring buffer Page Start adr        */
#define PSTOP         (NIC_TX_ADDR >> 8)        /* Ring buffer Page Stop  adr        */

#define START         (NIC_CR_RD2|NIC_CR_STA)   /* 0x22: activate NIC                */
#define STOP          (NIC_CR_RD2|NIC_CR_STP)   /* 0x21: De-activate NIC: SW reset   */
#define TRANSMIT      (NIC_CR_RD2|NIC_CR_TXP|NIC_CR_STA) /* 0x26: Transmit a packet */
#define REMOTE_R      (NIC_CR_RD0|NIC_CR_STA)   /* 0x0A: NIC command: Read  XE RAM   */
#define REMOTE_W      (NIC_CR_RD1|NIC_CR_STA)   /* 0x12: NIC command: Write XE RAM   */

#define NIC_RCR       (NIC_RCR_AB)              /* 0x04: Receive  config NIC reg     */
#define NIC_TCR       (0x00)                    /* 0x00: Transmit config NIC reg     */
#define NIC_DCR       (NIC_DCR_FT1|NIC_DCR_LS)  /* 0x48: Device Control NIC reg      */
#define NIC_IMR       (NIC_IMR_CNTE|            /* 0x3f: Interrupt Mask reg          */
                      NIC_IMR_OVWE|NIC_IMR_RXEE|
                      NIC_IMR_PRXE|NIC_IMR_TXEE|NIC_IMR_PTXE )

/* Low level Transmit routine: called from upper level software. Upon                */
/* completion, status can be checked in the interrupt service routine                */

xe_transmit(data,size)
register long *data;                             /* pointer to a packet              */
register short size;                             /* size of the packet               */
{
     RAMWrite(NIC_TX_ADDR,size,data);            /* write data to XE RAM to tx adr   */

     NIC->reg0 = PAGE0;
     NIC->reg4 = (NIC_TX_ADDR>>8);               /* Load the tx adr into NIC         */
     NIC->reg5 = (char)size;                     /* Load the size: LSB               */
     NIC->reg6 = (char)(size>>8);                /* Load the size: MSB               */
     NIC->reg0 = TRANSMIT;                       /* Transmit the packet              */
}
```

```
/* Get Ethernet address from XE ROM and put it in NIC registers */

xe_set_physaddr()
{
unsigned char phyadd[NICROMSIZ];

        RAMRead(XEROM_BOT,XEROMSIZ,phyadd); /* Read Ethernet address from XE ROM     */

        NIC->reg0 = PAGE1;                  /* point to physical address regs in NIC */

        NIC->reg1 = phyadd[0];  NIC->reg2 = phyadd[1];  NIC->reg3 = phyadd[2];
        NIC->reg4 = phyadd[3];  NIC->reg5 = phyadd[4];  NIC->reg6 = phyadd[5];

        NIC->reg0 = PAGE0;                  /* restore page pointer                  */
}

/*
 *  Routine that reads 'size' bytes from NIC RAM starting at 'start', and
 *  puts it at 'Data' in host (V30) RAM.
 *
 *  RAMWrite routine is very much the same except with 'sar' and 'dar' reversed
 *  and 'mr' programmed for mem-to-dev instead of dev-to-mem
 */

RAMRead(start,size,Data)
register unsigned int start,size;
register unsigned long *Data;
{
register unsigned long cnt=0,Stat;
register struct dma *dma = ((struct dma *)DMA_BASE);

      /* -------------- Set up NIC for a Remote-Read ---------------------------- */
      NIC->reg0 = PAGE0;
      NIC->reg8 = (char)start;                   /* RSAR0: LSB Remote Start adr   */
      NIC->reg9 = (char)(start >> 8);            /* RSAR1: MSB Remote Start adr   */
      NIC->rega = (char)size;                    /* RBCR0: LSB Remote bytecount   */
      NIC->regb = (char)(size >> 8);             /* RBCR1: MSB Remote bytecount   */
      /* -------------- Set up DMA for a peripheral-to-memory transfer ---------- */
      dma->dmachan2.sar = 0;                     /* DMA Source address            */
      dma->dmachan2.dar = (unsigned long) Data;  /* DMA Destination address       */
      dma->dmachan2.bar = 0;                     /* Not used                      */
      dma->dmachan2.tcr = size;                  /* DMA transfer count            */
      dma->dmachan2.mr  = DMA_MR2;               /* wordsize,burst,periph-to-mem  */
      dma->mask = dma->mask & DMA_MASK2;         /* Unmask channel 2              */
      /* -------------- Start DMA ----------------------------------------------- */
      dma->dmachan2.scr = DMA_SCR_STR;           /* Start we32104 DMA             */
      NIC->reg0 = REMOTE_R;                      /* Start Remote-DMA on NIC       */
      /* -------------- Completion ---------------------------------------------- */
      while (dma->dmachan2.scr & 0x100)          /* Poll DMA completion,          */
              for(cnt=0;cnt<0x2;cnt++);          /* could use interrupts instead  */

      return(dma->dmachan2.scr);                 /* Return status of transfer     */
}
```

```
/*
 *  Initializes the NIC onto an active network. It might make sense
 *  (depending on the application) to initialize everything "except"
 *  activating the NIC until the software is actually ready to receive packets.
 */

init()
{
register struct dma *dma = ((struct dma *)DMA_BASE);
unsigned long del;

      dma->dmachan2.dcr = DMA_DCR2; /* MUST DO THIS BEFORE ANYTHING ELSE ON XE    */

      temp = *NIC_RST_ON;           /* Hardware reset of XE (it powers up reset)  */
      for(del=0;del<0x80000;del++); /* wait...                                    */
      temp = *NIC_RST_OFF;          /* Release the reset on the XE                */

      *LED_ON = TRUE;               /* Turn on XE User LED                        */

      NIC->reg0 = STOP;             /* Software reset of NIC chip                 */
      while (!(NIC->reg7 & NIC_ISR_RST));  /* wait for NIC reset                  */

      NIC->rega = 0x00;      /* RBCR0:clear                                        */
      NIC->regb = 0x00;      /* RBCR1:clear                                        */
      NIC->reg1 = PSTART;    /* PSTART:   0x8000                                   */
      NIC->reg2 = PSTOP;     /* PSTOP:    0xF800                                   */
      NIC->reg3 = PSTART;    /* BNDRY:    0x8000                                   */
      NIC->reg7 = 0xFF;      /* ISR:      clear ints                               */
      NIC->reg4 = PSTART;    /* TPSR:     0x8000                                   */
      NIC->reg5 = 0x00;      /* TBCR0:    clear                                    */
      NIC->reg6 = 0x00;      /* TBCR1:    clear                                    */
      NIC->reg0 = 0x61;      /* CR:       Page 1                                   */
      NIC->reg7 = PSTART+1;  /* CURR:     0x8100                                   */

      xe_set_physaddr();     /* get Ethernet address from XE ROM and load NIC regs*/

      next_pkt = PSTART + 1; /* Next packet pointer in ring buffer,                */
                            /* update this upon packet reception                  */
      NIC->regf = NIC_IMR;   /* NIC Interrupt Mask Register                        */
      NIC->regc = NIC_RCR;   /* NIC Receive Config Register                        */
      NIC->regd = NIC_TCR;   /* NIC Transmit Config Register                       */
      NIC->rege = NIC_DCR;   /* NIC Device Control Register                        */

      NIC->reg0 = START;     /* Activate the NIC onto the network                  */

      *LED_OFF = TRUE;       /* Turn off user LED, or leave it on if you like.     */
}
```

15.10 Additional Literature

For additional technical information on the XE Module's component parts, please refer to these documents:

Topic	Part	Document
DMA	WE32104	AT&T WE32104 manual and application notes
CTI	DP8392	National Semiconductor DP8392 Coaxial Transceiver Interface Data Sheet
SNI	DP8391	National Semiconductor DP8391A Serial Network Interface Data Sheet
NIC	DP8390	Refer to the National Semiconductor documents: DP8390 Network Interface Controller: An Introductory Guide DP8390C Network Interface Controller Data Sheet DP8390 Data Sheet Addendum

Chapter 16

VMEBUS INTERFACE

16.1 INTRODUCTION

The VMEbus consists of P1 address, data, and control signals. P2 is used for the extended VMEbus address and data lines and the SCSI interface.

16.2 P1 (VMEbus) Pin Assignments

Not all of the signals are used on the HK68/V30. See section 8 for details and signal descriptions.

P1 Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	D00	BBSY*	D08
2	D01	BCLR*	D09
3	D02	ACFAIL*	D10
4	D03	BG0IN*	D11
5	D04	BG0OUT*	D12
6	D05	BG1IN*	D13
7	D06	BG1OUT*	D14
8	D07	BG2IN*	D15
9	Gnd	BG2OUT*	Gnd
10	SYSCLK	BG3IN*	SYSFAIL*
11	Gnd	BG3OUT	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	Gnd	BR3*	A23
16	DTACK*	AM0	A22
17	Gnd	AM1	A21
18	AS*	AM2	A20
19	Gnd	AM3	A19
20	IACK*	Gnd	A18
21	IACKIN*	SERCLK	A17
22	IACKOUT*	SERDAT*	A16
23	AM4	Gnd	A15
24	A07	IRQ7*	A14
25	A06	IRQ6*	A13
26	A05	IRQ5*	A12
27	A04	IRQ4*	A11
28	A03	IRQ3*	A10
29	A02	IRQ2*	A09
30	A01	IRQ1*	A08
31	-12V	+5V STDBY	+12V
32	+5V	+5V	+5V

Table 16-1. P1 (VMEbus) Connector Pinout

16.3 P2 (VMEbus, SCSI) Pin Assignments

P2 is used for the extended VMEbus signals (row B) and the SCSI interface (rows A and C).

P2 Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	PC-DB0	+5	PC-Gnd
2	PC-DB1	Gnd	PC-Gnd
3	PC-DB2	(reserved)	PC-Gnd
4	PC-DB3	A24	PC-Gnd
5	PC-DB4	A25	PC-Gnd
6	PC-DB5	A26	PC-Gnd
7	PC-DB6	A27	PC-Gnd
8	PC-DB7	A28	PC-Gnd
9	PC-DBP	A29	PC-Gnd
10	PC-Gnd	A30	PC-Gnd
11	PC-Gnd	A31	PC-Gnd
12	PC-Gnd	Gnd	PC-Gnd
13	PC-+5 Term Par	+5	PC-Spare
14	PC-Gnd	D16	PC-Gnd
15	PC-Gnd	D17	PC-Gnd
16	PC-ATN	D18	PC-Gnd
17	PC-Gnd	D19	PC-Gnd
18	PC-BSY	D20	PC-Gnd
19	PC-ACK	D21	PC-Gnd
20	PC-RST	D22	PC-Gnd
21	PC-MSG	D23	PC-Gnd
22	PC-SEL	Gnd	PC-Gnd
23	PC-C/D	D24	PC-Gnd
24	PC-REQ	D25	PC-Gnd
25	PC-I/O	D26	PC-Gnd
26	n/c	D27	n/c
27	n/c	D28	n/c
28	n/c	D29	n/c
29	n/c	D30	n/c
30	n/c	D31	n/c
31	n/c	Gnd	n/c
32	n/c	+5	n/c

Table 16-2. P2 (VMEbus, SCSI) Connector Pinout

The use of P2 is *required* in order to meet VME power specifications. "PC-" represents SCSI signals (section 12).

16.4 Power Requirements

Voltage	Current	Usage
+5	8.5A (typ), 9.0A (max) 11.0A (typ) with XE option	All logic
+12	0.5A, max	RS-232 I/F Ethernet I/F (XE option)
-12	0.5A, max	RS-232 I/F

Table 16-3. Power Requirements

The "+5" and "Gnd" pins on P2 row B *must* be connected to assure proper operation.

The XE option is connected to the VMEbus +5V and Ground through its own P2 connector. The +5V and Ground pins on P2 row B of the XE board *must* be connected to assure proper operation.

16.5 Environmental

Operating temperature: 0 to +55 degrees Centigrade, ambient, at board.

Humidity: 0% to 85%.

Storage temperature: -40 to +70 degrees C.

NOTICE: Power dissipation is about 45 watts (55 watts with the XE option). Fan cooling is required if the HK68/V30 board is placed in an enclosure or card rack. Fan cooling is also recommended when using an extender board for more than a few minutes.

16.6 Mechanical Specifications

Width	Depth	Height	(above board)
9.187 in.	6.299 in.	0.6 in.	0.8 in.
233.35 mm	160 mm	15.25 mm	20.35 mm

Table 16-4. Mechanical Specifications

Standard board spacing is 0.8 inches. The HK68/V30 is a 10-layer board.

Chapter 17

SUMMARY INFORMATION

17.1 SOFTWARE INITIALIZATION SUMMARY

This section outlines the steps for initializing the facilities on the HK68/V30 board. Certain steps must be performed in sequence, while others may be rearranged or omitted entirely, depending on your application.

1. The MPU automatically fetches the reset vector following a system reset and loads the supervisor stack pointer and program counter. The reset vector is in the first 8 bytes of ROM.

2. Recall the NV-RAM contents. (Reference: section 7.8)

3. Determine RAM configuration. (Reference: section 7.4)

4. Set the Bus Control Latch. (Reference: section 8.6)

5. Clear on-card RAM to prevent parity errors due to uninitialized memory reads. (Reference: section 6.2)

6. Load the 68030 Vector Base Register with the location of your exception vector table (usually at the start of RAM).

7. Initialize the exception vector table in RAM (at the selected base address.) This step links the various exception and interrupt sources with the appropriate service routines. (Reference: section 3.3)

8. Initialize the CIO. (Reference: section 10.7)

9. Initialize the DMAC (Reference: section 4.3)

10. Initialize the serial ports. (Reference: section 11.5)

11. Initialize the SCSI port. (Reference: section 12)

12. Initialize the Centronics port. (Reference: section 13)

13. Initialize the User LED port. (Reference: section 9.1)

14. Monitor the VME Logic Init Complete signal (via the CIO) and wait for true. (Reference: section 10.3)

15. Initialize the Mailbox logic. (Reference: section 8.9)

16. Release the VMEbus SYSFAIL line. (Reference: section 8.8)

17. Initialize the Ethernet interface (if the optional XE module is installed.

18. Initialize off-card memory and I/O devices, as necessary.

19. Enable system interrupts, as desired. (Reference: section 3.2)

17.2 On-Card I/O Addresses

This section is a summary of the on-card port addresses. It is intended as a general reference for finding additional information about a particular device. Refer to section 7.6 for a pictorial description of the system memory map.

Hex Address	Type	Device	Reference Section
0400,0000	R/W	base of VMEbus (Extended Adrs mode)	8.4
03xx,xxxx	R/W	HK68/V30 on-card RAM	7.3
02xx,xxxx	R/W	VMEbus (Standard Space)	8.4
0100,xxxx	R/W	VMEbus (Short Space)	8.4
00C6,0000	W	Mailbox Base Address	8.9
00C5,0000	W	Bus Control Latch	8.6
00C4,000E	W	VME SYSFAIL Release	8.8
00C4,000C	W	Watchdog Disable	3.8
00C4,000A	W	SCSI Reset	12.4
00C4,0008	W	SCSI Burst Mode	12.4
00C4,0006	W	Mailbox Interrupt Clear	8.9
00C4,0004	W	Mailbox Enable	8.9
00C4,0002	W	Serial DTR B	11.7
00C4,0000	W	Serial DTR A	11.7
00C3,000C	W	DMAC Channel 3 (SCC) Buffer Flush	4.4, 11.8
00C3,000A	W	DMAC Channel 1 (SCC) Buffer Flush	4.4, 11.8
00C3,0008	W	MPU Cache Enable	3.6
00C3,0006	W	User LED 4	9.1
00C3,0004	W	User LED 3	9.1
00C3,0002	W	User LED 2	9.1
00C3,0000	W	User LED 1	9.1
00C1,000x	W	VMEbus Interrupt Request	8.7
00C0,000x	R/W	Centronics	13.3
00B0,0000	R	NV-RAM Recall	7.8
00B0,0000	W	NV-RAM Store (tas)	7.8
00A0,00xx	R/W	NV-RAM Data	7.8
0090,00xx	R/W	CIO	10
0080,0200	R/W	DMAC register base	4
0080,0100	R/W	Ethernet base (XE option)	15
0080,0080	R/W	SCC base	11.7
0080,0000	R/W	SCSI base	12.4
0000,0000	R	EpROM base	7.2

Table 17-1. Address Summary

17.3 Hardware Configuration Jumpers

Jumper settings are detailed in the manual section pertaining to the associated device. This section can be used as a cross reference for finding additional information about the jumpers.

The default jumper settings configure the HK68/V30 as a system controller requesting the VMEbus on level 3.

Jumper	Function	Reference Section	Standard Configuration
J1	-12V to Serial Interface (P3-33)	11.2,11.10	removed
J2	RS-232 Defaults	11.2, 11.10	J2-B (True)
J3	+12V to Serial Interface (P3-32)	11.2,11.10	removed
J4	+5V to Serial Interface (P3-26)	11.2,11.10	removed
J5	Serial Port A RI, DCD Select	11.2, 11.10	J5-B (RI)
J6	Serial Port B RI, DCD Select	11.2, 11.10	J6-B (RI)
J7	SCC Clock		(factory set)
J8	DMAC Chnl 1 & 3 Transfer Req.	11.8	1-2,5-6 (A full duplex)
J9	FPU Clock		(factory set)
J10	RAM Timing		(factory set)
J11	ROM type	7.2	J11-B (27512)
J12	ROM type	7.2	J12-A (27512)
J13	ROM type	7.2	J13-A (27512)
J14	ROM type	7.2	J14-A (27512)
J15	SCSI Terminator Power	12.5	removed
J16	VME Slave Window Size	8.5	J16-A (4 Meg)
J17	VME Slave Window Size	8.5	J17-B (4 Meg)
J18	VME Slave Window Size	8.5	J18-A (4 Meg)
J19	VME Slave Window Size	8.5	J19-B (4 Meg)
J20	VME Bus Request	8.4	J20-D (level 3)
J21	VME Bus Grant Daisy Chain 2	8.3, 8.4	(level 3 system controller)
J22	VME Bus Grant Daisy Chain 0	8.3, 8.4	(level 3 system controller)
J23	BCLR*	8.3	installed
J24	SYSCLK*	8.3	J24-B (output)
J25	SYSRESET*	8.3	J25-A (input)
J26	VME Bus Grant Daisy Chain 3	8.3, 8.4	(level 3 system controller)
J27	VME Bus Grant Daisy Chain 1	8.3, 8.4	(level 3 system controller)
J28	VME Timeout (BERR)	8.3	installed (enabled)
J29	ACFAIL*	3.2, 8.2	installed

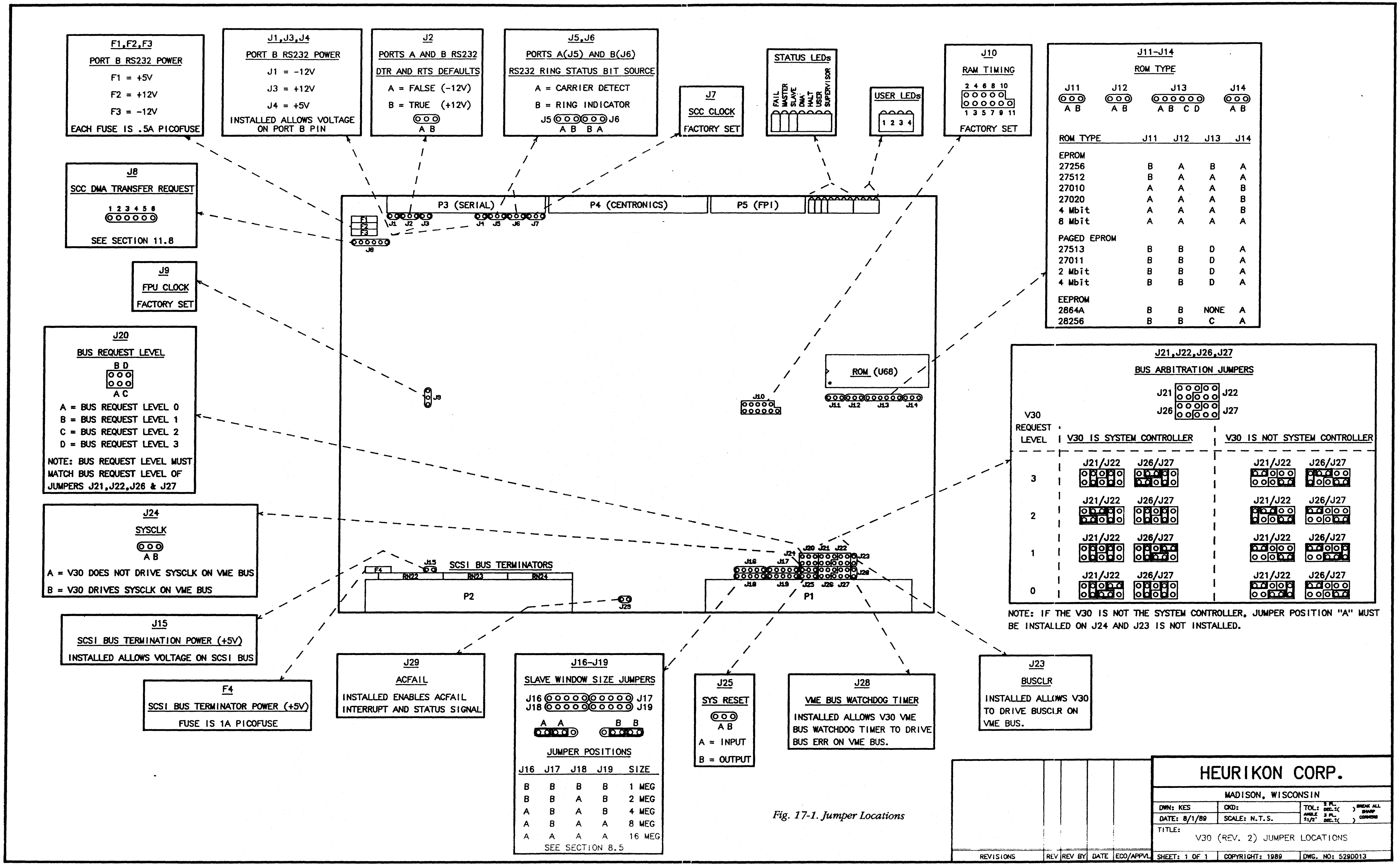
Table 17-2. Jumper Summary

The HK68/V30 has been designed to have a minimum number of configuration jumpers. As many options as possible are under software control. The NV-RAM can be used to store board and system configuration information.

17.4 Fuses

Fuse	Function	Reference Section	Rating	Standard Configuration	Heurikon Part #
F1	+5V to Serial I/F (P3-26)	11.2,11.10	0.5A	installed	2959002
F2	+12V to Serial I/F (P3-32)	11.2, 11.10	0.5A	installed	2959002
F3	-12V to Serial I/F (P3-33)	11.2, 11.10	0.5A	installed	2959002
F4	+5V to SCSI bus (P2-A13)	12.5	1A	installed	2959001

Table 17-3. HK68/V30 Fuses



F1,F2,F3

PORT B RS232 POWER

F1 = +5V

F2 = +12V

F3 = -12V

EACH FUSE IS .5A PICOFUSE

J1,J3,J4

PORT B RS232 POWER

J1 = -12V

J3 = +12V

J4 = +5V

INSTALLED ALLOWS VOLTAGE ON PORT B PIN

J2

PORTS A AND B RS232 DTR AND RTS DEFAULTS

A = FALSE (-12V)

B = TRUE (+12V)

A B

J5,J6

PORTS A(J5) AND B(J6) RS232 RING STATUS BIT SOURCE

A = CARRIER DETECT

B = RING INDICATOR

J5 A B B A J6

J7

SCC CLOCK

FACTORY SET

STATUS LEDs

FAIL MASTER SLAVE DMA HALT USER SUPERVISOR

USER LEDs

1 2 3 4

J10

RAM TIMING

2 4 6 8 10

1 3 5 7 9 11

FACTORY SET

J11-J14

ROM TYPE

J11 A B — J12 A B — J13 A B C D — J14 A B

ROM TYPE	J11	J12	J13	J14
EPROM				
27256	B	A	B	A
27512	B	A	A	A
27010	A	A	A	B
27020	A	A	A	B
4 Mbit	A	A	A	B
8 Mbit	A	A	A	A
PAGED EPROM				
27513	B	B	D	A
27011	B	B	D	A
2 Mbit	B	B	D	A
4 Mbit	B	B	D	A
EEPROM				
2864A	B	B	NONE	A
28256	B	B	C	A

J8

SCC DMA TRANSFER REQUEST

1 2 3 4 5 6

SEE SECTION 11.8

J9

FPU CLOCK

FACTORY SET

J20

BUS REQUEST LEVEL

B D

A C

A = BUS REQUEST LEVEL 0

B = BUS REQUEST LEVEL 1

C = BUS REQUEST LEVEL 2

D = BUS REQUEST LEVEL 3

NOTE: BUS REQUEST LEVEL MUST MATCH BUS REQUEST LEVEL OF JUMPERS J21,J22,J26 & J27

J21,J22,J26,J27

BUS ARBITRATION JUMPERS

J21 J22

J26 J27

V30 REQUEST LEVEL	V30 IS SYSTEM CONTROLLER		V30 IS NOT SYSTEM CONTROLLER	
3	J21/J22	J26/J27	J21/J22	J26/J27
2	J21/J22	J26/J27	J21/J22	J26/J27
1	J21/J22	J26/J27	J21/J22	J26/J27
0	J21/J22	J26/J27	J21/J22	J26/J27

NOTE: IF THE V30 IS NOT THE SYSTEM CONTROLLER, JUMPER POSITION "A" MUST BE INSTALLED ON J24 AND J23 IS NOT INSTALLED.

J24

SYSCLK

A B

A = V30 DOES NOT DRIVE SYSCLK ON VME BUS

B = V30 DRIVES SYSCLK ON VME BUS

J15

SCSI BUS TERMINATION POWER (+5V)

INSTALLED ALLOWS VOLTAGE ON SCSI BUS

F4

SCSI BUS TERMINATOR POWER (+5V)

FUSE IS 1A PICOFUSE

J29

ACFAIL

INSTALLED ENABLES ACFAIL INTERRUPT AND STATUS SIGNAL

J16-J19

SLAVE WINDOW SIZE JUMPERS

J16 J17

J18 J19

A A — B B

JUMPER POSITIONS

J16	J17	J18	J19	SIZE
B	B	B	B	1 MEG
B	B	A	B	2 MEG
A	B	A	B	4 MEG
A	B	A	A	8 MEG
A	A	A	A	16 MEG

SEE SECTION 8.5

J25

SYS RESET

A B

A = INPUT

B = OUTPUT

J28

VME BUS WATCHDOG TIMER

INSTALLED ALLOWS V30 VME BUS WATCHDOG TIMER TO DRIVE BUS ERR ON VME BUS.

J23

BUSCLR

INSTALLED ALLOWS V30 TO DRIVE BUSCLR ON VME BUS.

Fig. 17-1. Jumper Locations

REVISIONS	REV	REV BY	DATE	ECO/APPVL

HEURIKON CORP.

MADISON, WISCONSIN

DWN: KES	CKD:	TOL: 2 PL. DEC. ±( ) ANGLE ±1/2° 3 PL. DEC. ±( ) BREAK ALL SHARP CORNERS
DATE: 8/1/89	SCALE: N.T.S.	

TITLE: V30 (REV. 2) JUMPER LOCATIONS

SHEET: 1 OF 1	COPYRIGHT: 1989	DWG. NO: 529D013

17.5 ADDITIONAL TECHNICAL LITERATURE

Additional information is available on the HK68/V30 peripheral chips, either from Heurikon sales or directly from the chip manufacturers.

Device	Number	Document	Ref. Section
MPU	MC 68030	Motorola 68030 User's Manual	3
DMAC	WE32104	AT&T WE 32-bit Microprocessors and Peripherals AT&T WE 32104 DMA Controller Information Manual	4
FPP	MC68881 or MC68882	Motorola MC68881 /MC68882 User's Manual	5
CIO	Z8536	Zilog CIO Technical Manual	10
SCC	Z8530	Zilog SCC Technical Manual	11
SCSI	WD33C93	WD33C93 Technical Spec	12
RTC	DS1216F	Dallas Semiconductor Data Book	14
Ethernet		See Section 15.10 of this manual	

Table 17-4. Additional Technical Literature

HK68/V30 Board Revision Differences

A.1 Introduction

The body of this HK68/V30 User's Manual is written for board revision levels 2 or greater. This appendix contains a listing of the differences which apply to board revision levels P and 1.

A.2 Major Board Revision Level Changes

The major differences between board revisions P and 1 and revision levels of 2 or greater may be summarized as follows:

1. Board revisions 2 or greater support VMEbus operand caching by the processor.
2. Board revisions 2 or greater provide the ability to flush the DMAC buffers for channels 1 and 3.
3. The jumper numbers are different for board level revisions P and 1.

The following section of this manual lists the differences between board revision levels, listing and cross-referencing them according to the sections of this User's Manual.

A.3 Revision Level Change Listing

User's Manual Section	Changes Affecting Board Revision Levels P & 1
3.2	ACFAIL cannot be disabled; there is no ACFAIL jumper.
3.6	The control panel interrupt INTR* has no pullup resistor.
3.6	VMEbus operand caching is not implemented.

User's Manual Section	**Changes Affecting Board Revision Levels P & 1**
3.8	The on-card watchdog timeout is 800 microseconds instead of 200 microseconds.
4.4	DMAC buffer flushing for channels 1 and 3 is not implemented.
5.1	Jumper J10 selectes the FPP clock source.
6.2	The on-card watchdog timeout is 800 microseconds rather than 200 microseconds. The HK68/V30's VMEbus watchdog timeout is 200 microseconds instead of 100 microseconds. For an access to the HK68/V30 *from* the bus, a timeout will return a bus error to the bus master. No local *bus error* exception occurs. The timeout interval is set by the on-card access timeout if the local bus watchdog timer is enabled and if no VMEbus watchdog timer terminated the cycle first. If there is no response to a local bus timeout either for an on-card access by the MPU or DMAC or for an access to the HK68/V30 from the VMEbus, the board will reset in another 0.8 milliseconds (It is possible that the system will reset also. See section 8.2 for more information.).
7.2	The ROM chip position is U109. The ROM jumper settings are shown in the following table:

EPROM	ROM		----Jumper Positions----	
Type	Capacity	J13	J14	J15
27256	32 Kbytes	B	B	A
27512	64 Kbytes	A	B	A
270x0	128 to 1024 Kbytes	A	A	A
27513 Paged	64 Kbytes	D	B	B
1 Mbit 2 Mbit Paged		D	B	B
2864 R/W EEpROM	8 Kbytes	open	B	B
256K bit EEpROM	32 Kbytes	C	B	B

Table A-1. ROM Capacity and Jumper Positions - Revisions P and 1.

User's Manual Section	Changes Affecting Board Revision Levels P & 1	
7.7	Memory timing is controlled by jumper J9.	
8.2.1	There is no ACFAIL jumper. ACFAIL is always connected to CIO port C, bit 0 and MPU interrupt level 7.	
	The BCLR* jumper is J12.	
	BERR* can be generated due to a parity error only during a slave access, or by the VMEbus watchdog timer associated with jumper J27.	
	The SYSCLK jumper is J23.	
	The SYSRESET* jumper is J24.	
8.3	The jumper numbers are as follows:	
	SYSCLK	J23
	SYSRESET*	J24
	BCLR*	J12
	VMEbus watchdog	J27
	HK68/V30 Bus Request Level	J20-J22 J25,J26
8.3.3	The VMEbus watchdog timeout is 200 microseconds.	
8.4.2	The bus request jumper numbers are as given in 8.2.1 above.	
8.4.3	HK68/V30 board revisions P and 1 do not support VMEbus operand caching. Bit D23 of the Bus Control Latch is not used.	
8.5	The "Release With Hold" time is approximately 5 microseconds.	
8.6	Bit D23 is not used, as noted in 8.4.3 above.	
8.10	The jumper numbers are different. A complete listing of the jumpers for board revision levels P and 1 are shown in Table A-3 below.	
	There is no ACFAIL* interrupt jumper. The connection is hardwired.	

User's Manual Section — **Changes Affecting Board Revision Levels P & 1**

11.8.1 The SCC DMA request jumper block J8 is arranged as follows on board revision levels P and 1:

J8-1	SCC Port B Transmit Data Request (DTRB pin)
J8-2	DMA Channel 3 Request
J8-3	SCC Port A Transmit Data Request (DTRA pin)
J8-4	SCC Port A Receive Data Request (W/REQA pin)
J8-5	DMAC Channel 1 Request
J8-6	SCC Port B Receive Data Request (W/REQB pin)

11.8.2 DMAC buffer flushing for channels 1 and 3 is not implemented.

11.10 Jumpers J1, J3, and J4 do not exist on board revision levels P and 1. Fuses F1, F2 and F3 also serve as jumpers for power for the P3 connector.

12.5 The SCSI terminating resistors are RN25, RN26 and RN27.

The SCSI termination power jumper does not exist on board revision levels P and 1. Fuse F4 also serves as a jumper for SCSI termination power.

14.1 The pROM position on the board is U109.

The RTC must be plugged into the existing socket because there are components under the pROM. This makes the board profile wider (assuming standard pROM thickness), as shown in Table A-2.

Configuration	Component Height Above Board	Minimum Board Spacing
RTC module plugged into existing pROM socket:	.75 in.	.85 in. (2 slots)

Table A-2. RTC module, physical effects

User's Manual Section	Changes Affecting Board Revision Levels P & 1
14.1	Only one card slot is required if the board is in the end slot. Otherwise, it requires two slots.
14.6	If the Real-Time Clock (RTC) module is installed, see the note above for section 14.1.
17.2	DMAC channels 1 and 3 buffer flush addresses are not implemented.
17.3	The default jumper settings for board revision levels P and 1 are as shown in Table A-3. The default jumper settings configure the HK68/V30 as a system controller requesting the VMEbus on level 3.

Jumper	Function	Reference Section	Standard Configuration
J2	RS-232 Defaults	11.2, 11.10	J2-B (True)
J5	Serial Port A RI, DCD Select	11.2, 11.10	J5-B (RI)
J6	Serial Port B RI, DCD Select	11.2, 11.10	J6-B (DCD)
J7	SCC Clock		(factory set)
J8	DMAC Chnl 1 & 3 Data Rdy	11.8	J8-2,4 & 5,6
J9	RAM Timing		(factory set)
J10	FPU Clock		(factory set)
J11	SCSI Timing		(factory set)
J23	BCLR*	8.3	(installed)
J13	ROM type	7.2	J13-A (27512)
J14	ROM type	7.2	J14-B (27512)
J15	ROM type	7.2	J15-A (27512)
J16	VME Slave Window Size	8.5	J16-B (1 Meg)
J17	VME Slave Window Size	8.5	J17-B (1 Meg)
J18	VME Slave Window Size	8.5	J18-B (1 Meg)
J19	VME Slave Window Size	8.5	J19-B (1 Meg)
J20	VME Bus Request	8.4	J20-D (level 3)
J21	VME Bus Grant Daisy Chain 2	8.3, 8.4	(level 3 system controller)
J22	VME Bus Grant Daisy Chain 0	8.3, 8.4	(level 3 system controller)
J23	SYSCLK*	8.3	J23-B (output)
J24	SYSRESET*	8.3	J24-A (input)
J25	VME Bus Grant Daisy Chain 3	8.3, 8.4	(level 3 system controller)
J26	VME Bus Grant Daisy Chain 1	8.3, 8.4	(level 3 system controller)
J27	VME Timeout (BERR)	8.3	(installed) (enabled)

Table A-3. Jumper Summary

User's Manual Section	**Changes Affecting Board Revision Levels P & 1**
17.4	Fuses F1-F3 are removed.

F1,F2,F3

PORT B RS232 POWER

F1 = +5V

F2 = +12V

F3 = -12V

EACH FUSE IS .5A PICOFUSE

J2

PORTS A AND B RS232

DTR AND RTS DEFAULTS

A = FALSE (-12V)

B = TRUE (+12V)

A B

J5,J6

PORTS A(J5) AND B(J6)

RS232 RING STATUS BIT SOURCE

A = CARRIER DETECT

B = RING INDICATOR

J5 A B B A J6

J7

SCC CLOCK

FACTORY SET

STATUS LEDs

FAIL MASTER SLAVE DMA HALT USER SUPERVISOR

USER LEDs

1 2 3 4

J9

RAM TIMING

1 2 3 4 5 6 7 8 9 10 11

FACTORY SET

J8

SCC DMA TRANSFER REQUEST

1 2 3 4 5 6

SEE SECTION 11.8

J10

FPU CLOCK

FACTORY SET

J11

SCSI INTERFACE TIMING

FACTORY SET

J20

BUS REQUEST LEVEL

B D
A C

A = BUS REQUEST LEVEL 0

B = BUS REQUEST LEVEL 1

C = BUS RECUEST LEVEL 2

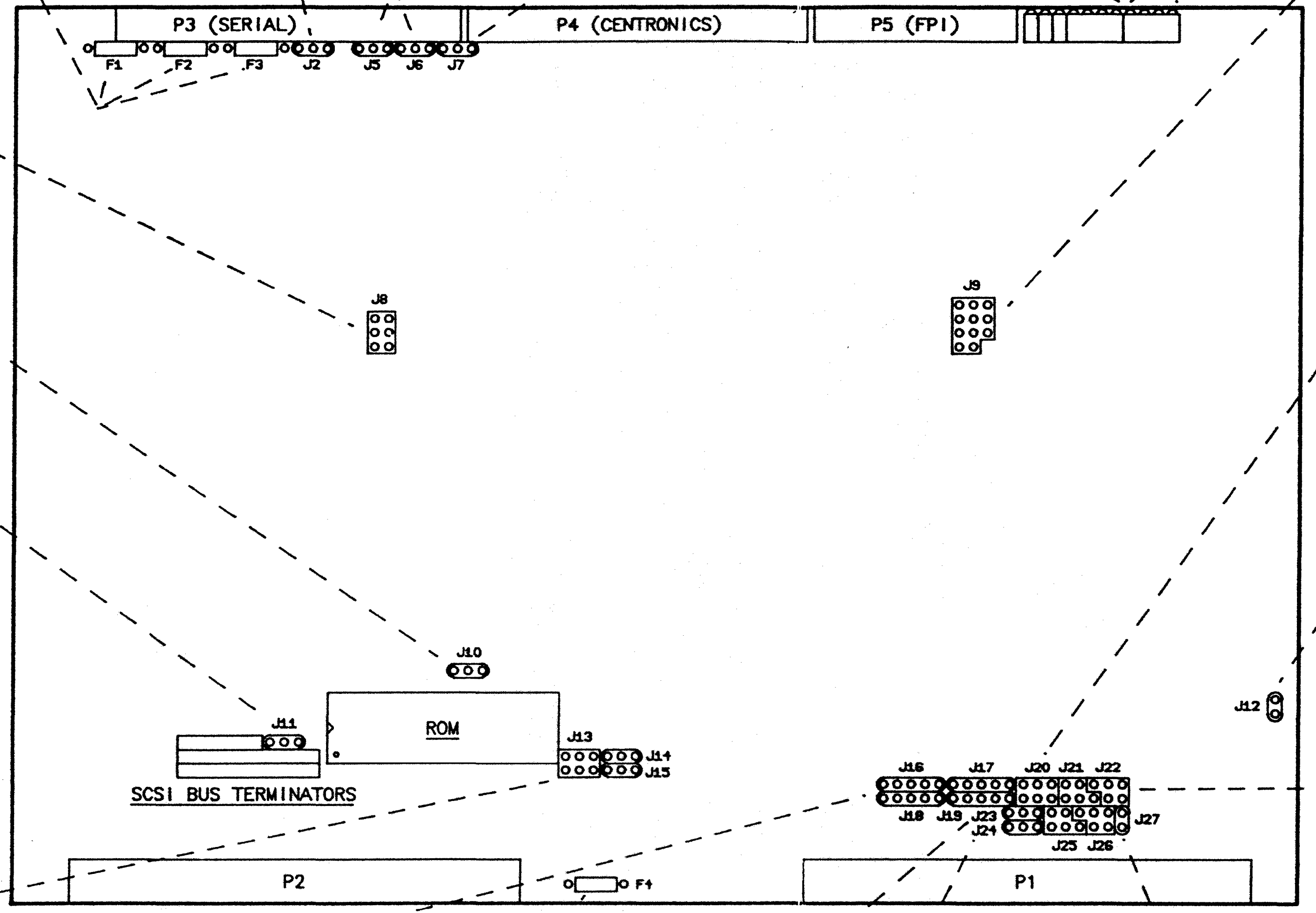
D = BUS RECUEST LEVEL 3

NOTE: BUS REQUEST LEVEL MUST MATCH BUS REQUEST LEVEL OF JUMPERS J21,J22,J25 & J26

J12

BUSCLR

INSTALLED ALLOWS V30 TO DRIVE BUSCLR ON VME BUS.



J13–J15

ROM TYPE

J13 A B C D J14 A B J15 A B

ROM TYPE	J13	J14	J15
EPROM			
27256	B	B	A
27512	A	B	A
27010	A	A	A
2 Mbit	A	A	A
4 Mbit	A	A	A
8 Mbit	A	A	A
PAGED EPROM			
27513	D	B	B
27011	D	B	B
2 Mbit	D	B	B
4 Mbit	D	B	B
EEPROM			
2864A	NONE	B	B
256 Kbit	C	B	B

J16–J19

SLAVE WINDOW SIZE JUMPERS

J16 J17
J18 J19

A A B B

JUMPER POSITIONS

J16	J17	J18	J19	SIZE
B	B	B	B	1 MEG
B	B	A	B	2 MEG
A	B	A	B	4 MEG
A	B	A	A	8 MEG
A	A	A	A	16 MEG

SEE SECTION 8.5

F4

SCSI BUS TERMINATOR POWER (+5V)

FUSE IS 1A PICOFUSE

J23

SYSCLK

A B

A = V30 DOES NOT DRIVE SYSCLK ON VME BUS

B = V30 DRIVES SYSCLK ON VME BUS

J24

SYS RESET

A B

A = INPUT

B = OUTPUT

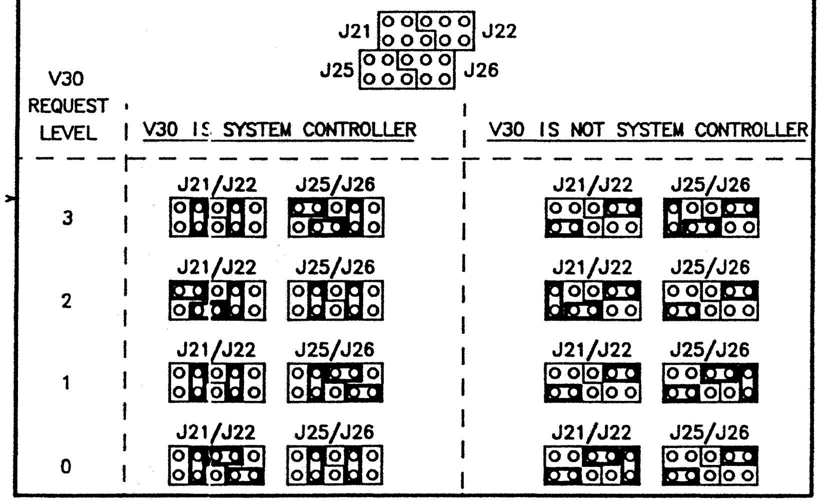
J27

VME BUS WATCHDOG TIMER

INSTALLED ALLOWS V30 VME BUS WATCHDOG TIMER TO DRIVE BUS ERR ON VME BUS.

J21,J22,J25,J26

BUS ARBITRATION JUMPERS



NOTE: IF THE V30 IS NOT THE SYSTEM CONTROLLER, JUMPER POSITION "A" MUST BE INSTALLED ON J23 AND J12 IS NOT INSTALLED.

Fig. A-1. V30 Rev. P Jumper Locations

REVISIONS	REV	REV BY	DATE	ECO/APPVL
REVISED J5, J6, J8, J9 AND J16–J19	B	KES	8/89	

HEURIKON CORP.

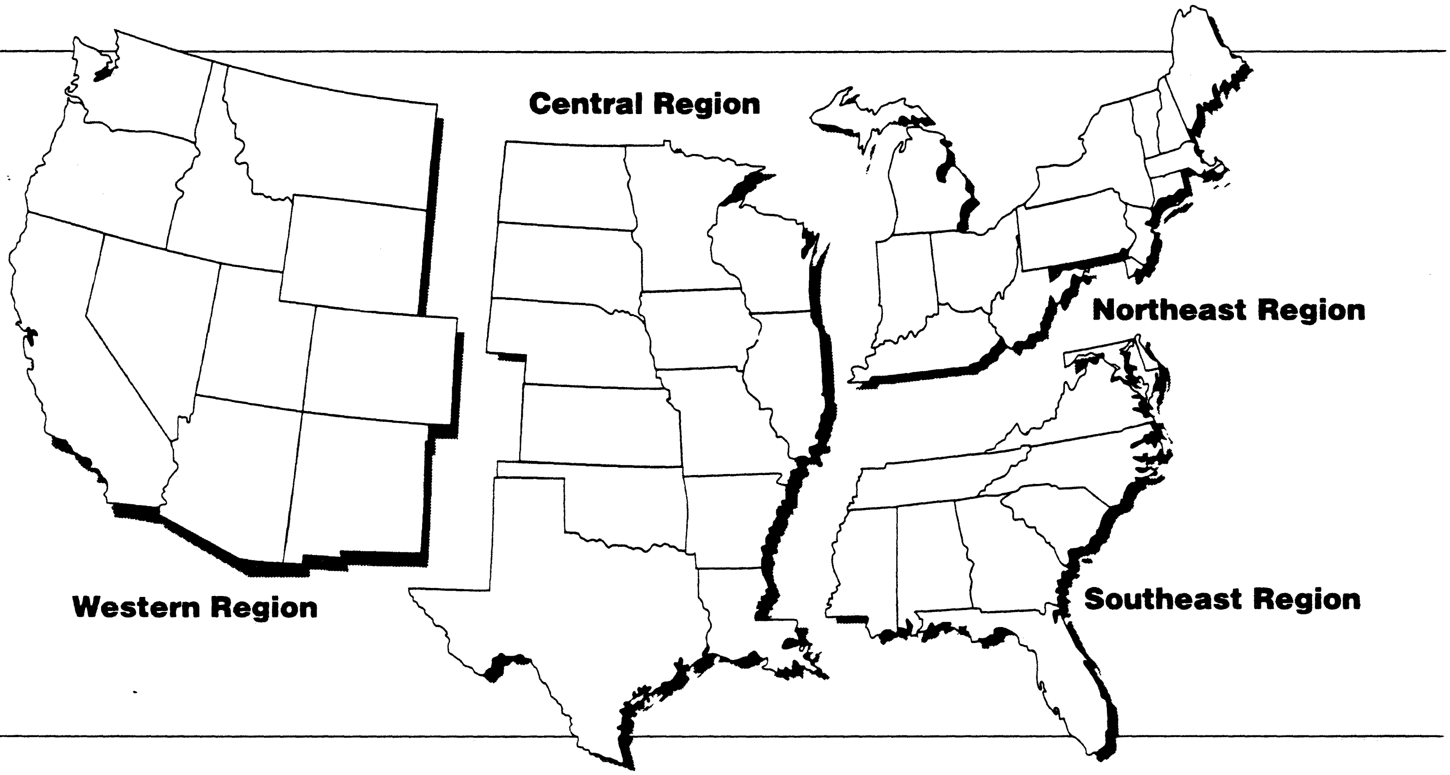
MADISON, WISCONSIN

DWN: KES	CKD:	TOL: 2 PL. DEC.±( ) 3 PL. DEC.±( ) ANGLE ±1/2° BREAK ALL SHARP CORNERS
DATE: 8/1/89	SCALE: N.T.S.	

TITLE: V30 (REV. P) JUMPER LOCATIONS

SHEET: 1 OF 1	COPYRIGHT: 1989	DWG. NO: 529D004-B

Sales and Customer Service Offices



Heurikon Corporate Office
8000 Excelsior Drive
Madison, WI 53717
Watts: 800-356-9602
Phone: 608-831-0900
Fax: 608-831-4249

Heurikon Customer Support and Factory Service Office
8310 Excelsior Drive
Madison, WI 53717
Watts: 800-327-1251
Phone: 608-831-5500

Heurikon Northeast Regional Office
67 South Bedford, Suite 400 W.
Burlington, MA 01803
Phone: 617-229-5831
Fax: 617-272-9115

Heurikon Southeast Regional Office
2010 Corporate Ridge, Suite 700
McLean, VA 22102
Phone: 703-749-1474
Fax: 703-556-0955

Heurikon Central Regional Office
13100 West 95th Street, Level 4D
Lenexa, KS 66215
Phone: 913-599-1860
Fax: 913-599-1918

Heurikon Western Regional Office
23121 Plaza Pointe Dr., Suite 109
Laguna Hills, CA 92653
Phone: 714-581-6400
Fax: 714-581-8509

HEURIKON CORP.
OPEN SYSTEMS :: OPEN TOOLS

revised 1.91

Regional Sales Representatives

• NORTHEAST REGION

CT, DE, ME, MA, NH, NJ, NY, Eastern PA, RI and VT

Daner-Hayes, Inc.
62 West Plain Street
Wayland, MA 01778
Tel: (508) 655-0888
Fax: (508) 655-0939

IN, KY, MI, OH, Western PA and WV

Systems Components, Inc.
1327 Jones, Suite 104
Ann Arbor, MI 48105
Tel: (313) 930-1800
Fax: (313) 930-1803

• CENTRAL REGION

AR, LA, OK and TX

Acudata, Inc.
720 Avenue F, Suite 104
Plano, TX 75074
Tel: (214) 424-3567
Fax: (214) 422-7342

MN, ND, SD and Northwest WI

Micro Resources Corp.
4640 W. 77th Street,
Suite 109
Edina, MN 55435
Tel: (612) 830-1454
Fax: (612) 830-1380

IL, IA, KS, MO, NE and Southeast WI

Panatek
2500 West Higgins Road,
Suite 305
Hoffman Estates, IL 60195
Tel: (708) 519-0867
Fax: (708) 519-0897

• SOUTHEAST REGION

MD, VA and Washington, DC

Spectro Associates
1107 Nelson Street, #203
Rockville, MD 20850
Tel: (301) 294-9770
Fax: (301) 294-9772

• WESTERN REGION

AZ, CO, NV, NM and UT

Compware Marketing
100 Arapahoe Ave.
Suite 7
Boulder, CO 80302
Tel: (303) 786-7045
Fax: (303) 786-7047

ID, MT, OR, WA, WY and Canada (Alberta and British Columbia)

Electronic Component Sales
9311 S. E. 36th Street
Mercer Island, WA
98040-3795
Tel: (206) 232-9301
Fax: (206) 232-1095

CA

Qualtech
333 West Maude Avenue,
Suite 108
Sunnyvale, CA 94086
Tel: (408) 732-4800
Fax: (408) 733-7084

HEURIKON CORP.
OPEN SYSTEMS :: OPEN TOOLS

revised 1.91

INDEX