SUN 68000 Board

SUN 68000 Board

Features

- 10-MHz M68000 CPU,
- no wait states with on-board RAM,
- virtual memory management,
- 256K bytes on-board RAM with byte parity,
- memory expandable to 2 megabytes
- up to 32K bytes EPROM,
- two programmable high-speed serial I/O channels,
- five programmable 16-bit timers,
- one 16-bit input port,
- fully compatible with IEEE-796 Bus (Intel Multibus),
- single 5 Volt power supply.

Overview

The SUN 68000 board combines the processing power of the 10 MHz 68000, virtual memory management, 256k bytes of main memory, and input/output on a single board compatible with the IEEE 796 Bus (Intel Multibus).

The SUN 68000 board has been designed to support a multi-tasking operating system such as Bell Lab's UNIX at maximum performance. A multi-process two-level memory management unit with facilities for protection, code sharing, and demand paging is integral to the board. 256K bytes of local RAM are provided. Local memory is expandable to 512K bytes, which the 10 MHz 68000 can access at full speed without wait states. Nonlocal memory, accessed via the Multibus, is expandable to a total of 8 megabytes of accessible memory, of which 2 megabytes may be directly addressed at any one time depending on the entries in the page map. Other functions include two user-programmable, high-speed RS-423 UARTs, five 16-bit timers, one 16-bit input port, and a multi-master 796-Bus interface.

Description

Central Processing Unit

The SUN 68000 board is based on the Motorola 68000 processor, a high-performance microprocessor with a 32-bit architecture and a large, uniform memory space. The 68000 features eight 32-bit address registers, eight 32-bit data registers, a 32-bit program counter, three major data sizes (byte, word, and long word), supervisor and user states, and flexible addressing modes.

Bus Structure

The SUN 68000 Board uses two buses: an internal synchronous bus for communicating with local memory and I/O devices, and the 796-Bus (Multibus) system bus for referencing additional memory and offboard I/O devices. Local memory accesses do not require the 796-Bus, making the system bus available for use by other 796-Bus masters such as DMA devices. This also allows true parallel processing in a multiprocessor environment. On board memory is private and fully protected from the 796-Bus.

Memory Capabilities

The SUN 68000 processor board can be configured with 128K or 256K bytes of on-board dynamic RAM, equipped with byte parity. Four 28-pin sockets are provided for EPROM or ROM. The board accomodates Intel 2716, 2732, and 2764 type EPROMs, offering 8K, 16K, and 32K bytes of PROM, respectively.

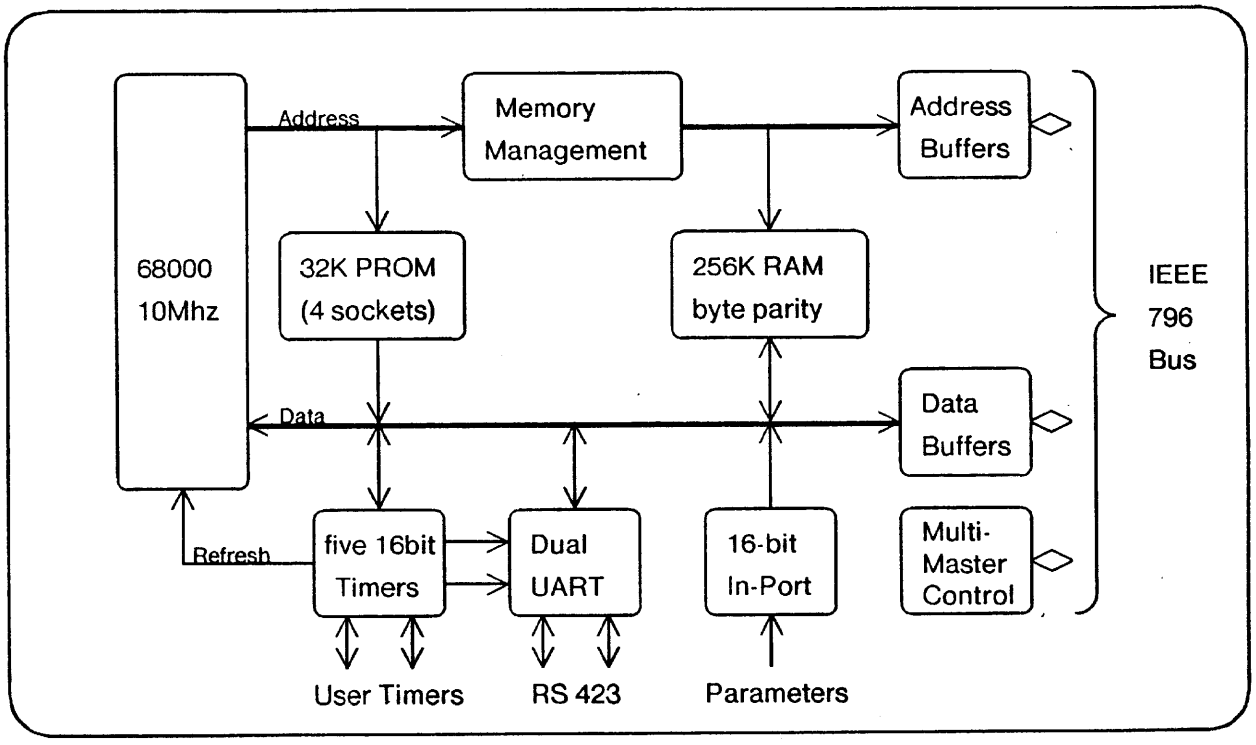

Figure 1: The SUN 68000 Board Architecture

Memory Expansion

Local RAM memory is expandable from 256k to 512K bytes with the SUN 68000 memory expansion board. This memory board communicates with the processor board through a private memory bus connecting the P2 connectors of both boards. The memory expansion board offers the same byte parity protection and high-speed operation as the local RAM on the SUN 68000 processor board. A Megabyte of memory can be added on the 796-Bus (Multibus).

Input/Output Capabilities

The board incorporates a dual UART chip (NEC 7201 or Intel 8274), a five-counter timer chip (AMD 9513) and a general purpose 16-bit input port.

The dual UART provides two high-speed serial i/o channels (up to 880k Baud). It can be programmed for asynchronous, IBM Bisync, and SDLC/HDLC protocols with automatic sync insertion and CRC generation. Baud rates and communication modes are fully software programmable. Line drivers and receivers are compatible with the RS-423 standard, connector pinout is defined according to RS-232. One channel is configured to communicate with a terminal, the other can be jumpered to talk either to a host computer or to a second terminal.

The timer chip contains five independent 16-bit counters/timer. Timers one and two are available for user applications. Timer three is dedicated for the memory refresh task. Timer four and five are used as programmable baud rate generators for the UART. Timer one can optionally be used as a watchdog timer to auto-reset the processor in case of unexpected halt. This allows remote or standalone systems to automatically reboot themselves in case of catastrophic failure.

The 16-bit parallel input port is intended to serve as a configuration or "option" port to read in parameters such as UART baud rates, boot options, or other user options.

796-Bus Capabilities

The SUN 68000 board is fully compatible with the IEEE-796 Bus (Intel Multibus). The IEEE-796 Bus is an asynchronous bus, accommodating devices with various transfer rates while maintaining maximum throughput.

Using the 20 address lines of the standard IEEE-796 Bus, the board can address up to 1 megabyte of memory and 1 megabyte of input/output locations. In conjunction with the 256k byte on-board memory and the 256k byte expansion memory board the maximum directly addressable memory is thus 1.5M bytes.

The SUN 68000 board also has full multi-master capabilities that allow it to share the 796-Bus with several other processor boards or DMA devices. The on-board arbitration logic automatically arbitrates access to the bus when an off-board cycle is executed, yielding to higher priority bus masters. Up to three masters can be in a system utilizing the serial priority method, or up to 16 masters in a system with parallel priority arbitration.

The SUN 68000 board can optionally provide the bus clock, constant clock, and init signal for the 796-Bus. Init is generated by an on-board precision voltage reference when the supply voltage falls below 4.75V. Init can also be generated by the 68000.

An on-board timeout is provided to abort 796-Bus cycles if the addressed device does not respond within 15 microseconds. The short timeout period guarantees that in case of timeout the 68000 can resume normal operation and satisfy real-time requirements without compromising system integrity. In a multi-master environment, it also ensures that the bus is not locked out for excessive time.

Memory Management

The SUN 68000 Memory Management Unit has been designed to support a multitasking operating system such as Bell Lab's UNIX. It provides address translation, protection, sharing, and memory allocation for multiple processes executing on the 68000. All accesses of the 68000 to on-board RAM memory, 796-Bus memory, and 796-Bus I/O space are translated and protected in an identical fashion. Finally, the SUN 68000 memory management provides all the necessary mechanism for demand paging and virtual memory and will be fully compatible with the 68010 virtual memory processor when it becomes available.

The memory management consists of a context register, a segment map, and a page map. Virtual addresses from the 68000 are translated into intermediate addresses by the segment map and then into physical addresses by the page map.

The page size is 2 kilobytes, the segment size is 32 kilobytes, and up to 16 contexts can be mapped concurrently. The maximum logical address space for a context on the SUN 68000 board is 1024 pages or 2 MByte. The maximum physical address space that can be mapped simultanously is 2 Mbytes. A total of 8 Mbytes can be addressed from the page map, permitting an 8 Mbyte memory to play the role of a swapping device, with the swapping time being that required to rewrite the relevant page map entries.

The organization of the memory management system is shown in figure 2 below.

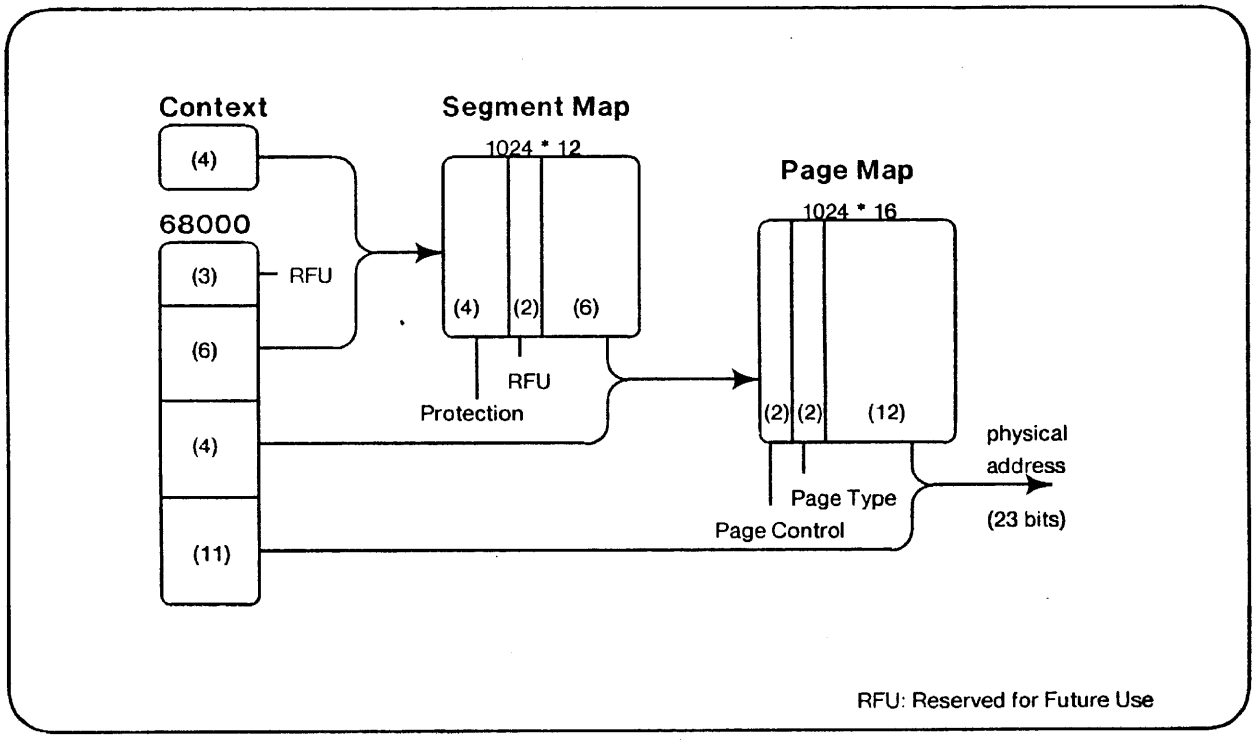

Figure 2: The SUN 68000 Memory Management

Context Register

In a multitask environment it is important to be able to switch between processes quickly without having to reload all the translation state information of a particular process. The context register is a four-bit register set under supervisor control that can switch between 16 sections of the segment map with one 68000 move instruction. This permits 16 contexts to be mapped concurrently; more than 16 contexts may be handled by treating the segment map as a context cache, replacing out-of-date contexts on a least-recently-used or other basis.

Each context has its own virtual address space. Sharing and intercontext communication may be implemented at either the segment or page level.

A simple implementation of multiprocessing will allocate one context per process. More complex schemes are possible in which a team of processes occupies one context, or in which one process extends over more than one context (with context changes managed via system calls).

Segment Map

The segment map has 1024 entries, indexed by the six most most significant bits of the logical address and the 4-bit context register. Thus, the segment map is divided into 16 sections of 64 entries, one section for each context. The segment map entry contains the six high-order virtual address bits and a four-bit protection code, defined below.

Each logical address space thus has up to 64 segments. The nominal segment size is 16 pages or 32k bytes. However segments can be made as small as a single 2 kilobyte page by invalidating the undesired pages in the page map, or as large as 1024 pages (2 MByte), by concatenating consecutive segment map entries and assigning each the same protection code.

Protection

Protection is associated with the segment map; each segment has a protection code permitting one of 16 classes of access modes allowing read, write, and execute cycles in each of supervisor and user states. The 16 protection codes are defined in the following table. Full access is denoted **"rwxrwx"**, with the first **"rwx"** being Read-Write-eXecute for the system and the second for the user. A **"-"** denotes the absence of that privilege.

0	------	No access
1	--x---	System execute only
2	r-----	System read only
3	r-x---	System read and execute
4	rw----	System read and write
5	rwx---	System read, write, and execute
6	r--r--	System read; User read
7	rw-r--	System read and write; User read
8	r--rw-	System read; User read and write
9	rw-rw-	System read and write; User read and write
10	rw-r-x	System read and write; User read and execute
11	rw-rwx	System read and write; User read, write, and execute
12	r-xr-x	System read and execute; User read and execute
13	rwxr-x	System read, write, and execute; User read and execute
14	rwx--x	System read, write, and execute; User execute;
15	rwxrwx	Full access

Page Map

The page map handles paging and the allocation of physical memory. A page map entry also indicates the physical address space in which a page is located, such as on-board or off-board memory. Further, the page map assists demand paging algorithms by maintaining reference and modified bits for each page.

In the page map, the virtual address from the segment map and the next four logical address bits from the 68000 are translated into a physical address and a physical address space. Thus each segment accesses a block of 16 consecutive pages.

The output of the page map is 12 bits of physical address that is concatenated with the 11-bit byte address to form a 23-bit physical address. In addition, a page can be declared to be in on-board memory space, off-board memory space, off-board i/o space, or non-existent.

A non-existing entry indicates an invalid page, causing instruction abort. Notice that each of these address spaces is 23 address bits or 8 MByte large. Since on-board memory is at most 256K bytes (512K bytes with memory expansion board) and the address space on the standard 796-Bus is at most 1 MByte for memory and 64K bytes for Input/Output, it is the responsibility of the memory management software to provide correct table entries for a particular system configuration.

Page Control

In addition to the page mapping information, each page entry has two associated statistic bits, *modified* and *accessed*, that are set whenever that page has been accessed and has been written into. These bits are updated automatically on all cycles for which access has been granted by the protection mechanism.

Virtual Memory Capabilities

The organization of the page map in conjunction with the page control bits provides all the necessary mechanisms to implement demand paging and virtual memory.

If either the segment map or the page map indicates an invalid segment or page during translation, the MMU will send a bus error signal to the processor. The operating system will then check whether the access was in error or whether the fault was due to a missing page. In the latter case, the missing page needs to be loaded into main memory. If all page entries are in use, a page must be replaced. The *modified* and *accessed* bits can assist the page replacement algorithm. When the page is loaded, the aborted instruction can be continued.

Unfortunately, the current version of the 68000 processor does not cannot fully recover from page faults because it does not save sufficient state information to continue an aborted instruction. For a limited set of operations, such as load and store instructions, recovery is possible. With additional software assistance, recovery from more complex addressing modes appears feasible. The SUN 68000 board has been designed to be upwards compatible with the 68010 virtual-memory version of the 68000 processor when it becomes available.

Exception Handling

When a processor cycle cannot be completed normally an exception is performed. Besides the exceptions caused by internal processing, such as divide-by-0 or word-access to an odd-byte address, a variety of external conditions can make it impossible to complete the current instruction or bus cycle. These external conditions abort the current cycle via the Bus Error exception. Bus Error exceptions arise from five error conditions: as system space errors, segment map erros, page map errors, timeout errors, and parity errors.

System space errors are caused when an address greater than or equal to 20000 (Hex) is accessed in user mode. These addresses are reserved for supervisor state to address the onboard system facilities. A segment map error indicates that the protection bits in the segment map did not allow the type of operation attempted. A page map error is caused by accessing an invalid page. Timeout errors occur for off-board references to the 796-Bus that are not acknowledged within 15 microseconds. Most likely, non-existent memory or a non-existing device has been addressed. There are no timeouts for on-board references because the on-board bus is synchronous and all cycles are always acknowledged. Parity errors occur is a byte or word with odd parity is read from local RAM. Since parity can only be checked at the end of a memory read cycle, the 68000 can not abort the cycle in which the error occured, but the next cycle.

When a bus error occurs, the cause of the error can be determined by checking whether the attempted access was to system space in user mode, whether a mapped access violated the segment protection code, or whether the page refererenced was non-existent. If none of the above caused the exception, then the exception was a timeout for bus accesses, or a parity error local accesses in the previous memory cycle.

Interrupts

The 68000 has seven interrupt levels, numbered 1 through 7, with level 7 being the highest priority and level 1 the least priority. Interrupts are recognized for all priority levels greater than the current processor priority contained in the 68000 status register. When an interrupt is acknowledged, the processor priority is set to the level of the interrupt request.

A level 7 interrupt is special in that it is recognized even if the mask in the 68000's status register is set to 7, thus providing a non-maskable interrupt capability. A level 7 interrupt is acknowledged every time the interrupt request changes from a lower level to level 7, that is, level 7 interrupts are "edge-triggered".

The 796-Bus standard defines 8 interrupt lines, INT0 through INT7, with INT0 being the highest priority. Also, the standard recommends that interrupts be level-triggered instead of edge-triggered to allow multiple interrupt sources on each interrupt line.

To avoid confusion for 68000 programmers, the numbering and the priorities of the interrupt lines on the 796-Bus was made corresponding to the definition of the 68000. Thus INT7 on the 796-Bus is the highest priority interrupt and INT1 the lowest. INT0 is not implemented. In addition, INT7 is non-maskable and edge-triggered, whereas all other interrupts are maskable and level-triggered.

Three interrupt lines are assigned to on-board interrupt sources: **INT7:** Refresh timer, **INT6:** User Timer, **INT5:** UART. Jumpers are provided if alternate interrupt assignments are desired.

Interrupts are acknowledged by the 68000 in auto-vector mode, that is, the interrupt vector is generated internally by the 68000 and is not supplied by the device. Thus the INTA signal on the 796 Bus and the interrupt vector capabilities of the 796-Bus are not used.

Initialization

After hardware reset, the 68000 Board comes up in a special "Boot state". In this boot state, the normal operation of the board is changed as follows:

1) The on-board PROM, normally residing in system address space, overlays RAM starting at location 0, but is also accessible under its normal location as well. Thus the initial program counter and stack pointer are fetched from PROM at locations 0 to 3, whereas other bootstrap code can execute from normal PROM addresses.

2) Since the PROM is overlaid at location 0, read access to the on-board RAM and to the 796-Bus is disabled. However, write access is possible allowing initialization of exception and interrupt vectors in RAM.

3) All interrupts, including the non-maskable interrupt, are disabled in hardware. After leaving boot state, non-maskable interrupts can occur at any time, and maskable interrupts as soon as the interrupt mask in the status register is lowered to allow them.

Boot state is exited by writing once to the BOOTEXIT location in system space.

PROM Monitor

The PROM resident monitor provides for system initialization, memory refresh, emulator traps, and console I/O. The monitor commands are as follows:

```
A <n>    Open address register <n>.
B        Set a breakpoint.
C<addr>  Continue a program.
D        Open data register.
E<addr>  Open memory location <addr>.
F<name>  Load the file <name> over the network.
G<addr>  Start program execution with a subroutine call.
H        Help, prints out the list of commands.
I<mode>  Set UART input mode: A: monitor; B: remote; T: transparent.
K        Kill, soft reset.
L<arg>   Load a file <arg> over serial line.
M<m>     Open segment map <m>.
N<name>  Load the file <name> over the network and start execution.
O<addr>  Open the byte location <addr>.
P<p>     Open page map <p>.
R        Open system registers SS,US,SR,PC.
S<s>     Load the S-record <s>.
X<char>  Set transparent mode character to <char>.
```

Initialization after powerup involves initializing UARTs and TIMERs, setting up segment and page maps, sizing local RAM and writing all RAM to clear parity errors. The monitor also sets up a memory refresh routine that executes 128 memory cycls every 2 milliseconds.

Finally, the monitor initializes the watchdog timer to reset the processor should it ever get into a halt state without losing main memory content.

Specifications

Processor

10 MHz 68000

Memory

RAM: 256k Bytes with byte parity, expandable to 512k Bytes
PROM: four 28 pin sockets are provided for 2716, 2732, or 2764 EPROMs.

Input/Output

two programmable RS423 serial I/O channels
five programmable 16-bit Timers
one 16-bit input port

796-Bus Compatibility

D16 M20 I20 VOL.

Electrical Characteristics

VCC = +5V +-5%
ICC = 5A max.

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.50 in. (1.27 cm)
Weight: 16 oz. (447 g)

Environmental Characteristics

Operating Temparature: 0-50 C

References

Motorola Inc., "MC68000 16-bit Microprocessor User's Manual, IMC68000UM, 1980.
Intel Corp., 8274 Data Sheet, 1981.
AMD INC., 9513 Data Sheet, 1979.
R.W.Boberg, "796 Microprocessor Bus Standard", Computer, Oct. 1980.

List of Figures