Sun 68000 Board

User's Manual

Sun Microsystems Inc.

February 1983

Revision B

The Sun 68000 Board is a powerful single-board computer combining a 10 MHz 68000, virtual memory management, 256K bytes of main memory, and input/output on a single board compatible with the Intel Multibus (IEEE 796 Bus).

This document describes the architecture, programming, and installation of the Sun 68000 Board.

Table of Contents

List of Figures

1. Architecture Overview

1.1. Features

- 10-MHz M68000 CPU,
- virtual memory management,
- 256K bytes on-board RAM with byte parity,
- 0-wait state access to on-board memory
- 0-wait state 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 Intel Multibus (IEEE-796 Bus),
- single 5 Volt power supply.

1.2. Introduction

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

The Sun 68000 board has been designed to support a multi-tasking operating system such as Bell Labs' 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. The Sun 68000 board includes 256K bytes of 0-wait state RAM. 0-wait state memory is expandable to 2M bytes with Sun Memory Expansion Boards. Multibus memory and I/O space are expandable to 1 MByte each. Other on-board functions include two user-programmable, high-speed serial channels, five 16-bit timers, one 16-bit input port, and a multi-master Multibus interface.

1.3. 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.

1.4. Bus Structure

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

1.5. Memory Capabilities

The Sun 68000 processor board offers 256K bytes of on-board dynamic RAM which the 68000 can access without wait states. On-board memory features byte parity error detection. Four 28-pin sockets are provided for 2764-type EPROMs, offering 32K bytes of EPROM.

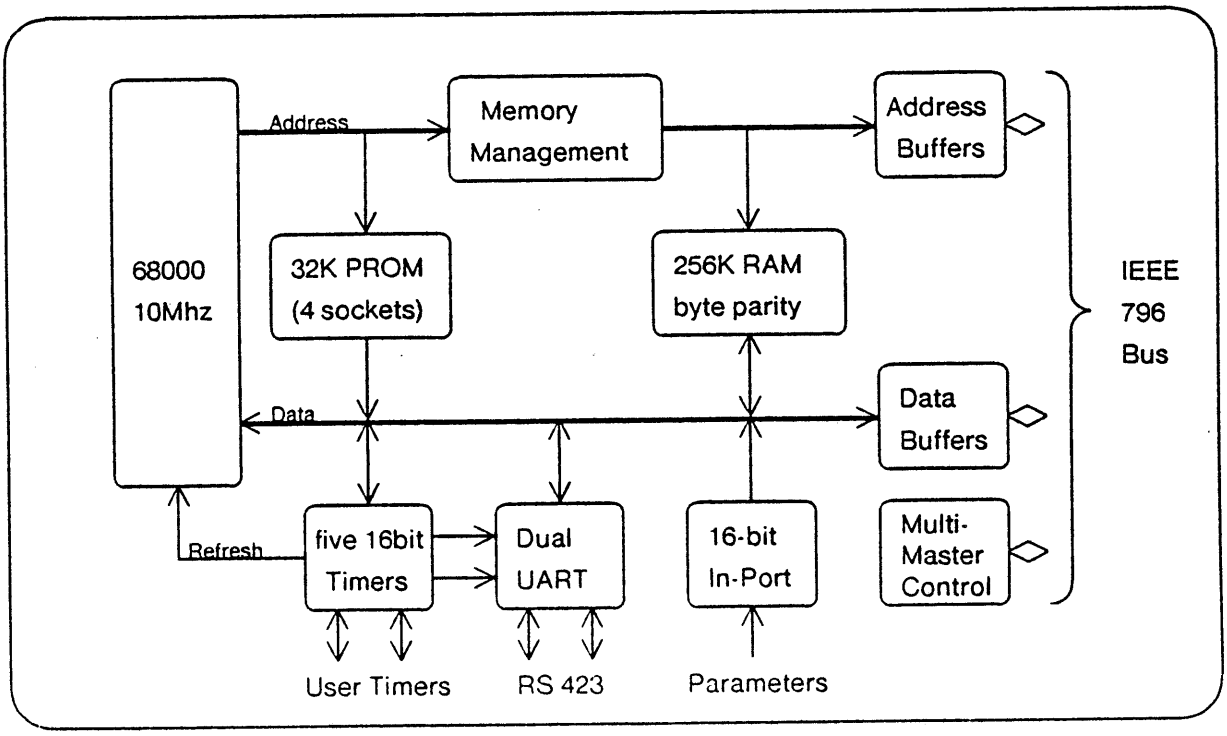


Figure 1-1: Sun 68000 Board Architecture

1.6. Memory Expansion

With the Sun Memory Expansion board, 0-wait state RAM can be increased to 1 MByte with one expansion board and 2 MByte with two expansion boards. The processor communicates with the expansion memory through a private high-speed memory bus carried over the P2 connector. The expansion memory offers the same byte parity protection as the local RAM on the Sun 68000 processor board. In addition, up to 1 MByte of RAM can be addressed on the Multibus (IEEE 796-Bus).

1.7. Input/Output Capabilities

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

The dual UART provides two high-speed serial I/O channels. Communication modes and baud rates are fully software programmable. One of the serial channels provides modem control signals. Line drivers are compatible with RS-423 signal levels.

The timer chip contains five independent 16-bit counters/timer. One timer is available for user applications, one timer is dedicated for the memory refresh task, two of the timers generate the baud rates for the two serial channels, and one timer is used as watchdog timer to reset the processor in case of unexpected halt. The watchdog timer allows remote or standalone systems to automatically reboot themselves in case of catastrophic failure.

The 16-bit parallel input port provides 16 lines of unbuffered input suitable for configuration control or parallel input devices.

1.8. Multibus Capabilities

The Sun 68000 board is fully compatible with the Multibus (IEEE 796-Bus). The Multibus is an asynchronous bus, accommodating devices with various transfer rates while maintaining maximum throughput. Using the 20 address lines of the standard Multibus, the board can address up to 1M bytes of memory and 1M bytes of input/output locations, even though most input/output devices only decode 64K bytes. An on-board timeout is provided to abort Multibus cycles if the addressed device does not respond within the timeout period.

The Sun 68000 board also has full multi-master capabilities that allow it to share the Multibus with 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 (BLCK), constant clock (CCLK), and initialization (INIT) for the Multibus. Init is generated by an on-board precision voltage reference when the supply voltage falls below 4.5V, or by executing a 68000 reset instruction.

1.9. Specification Summary

Processor

10 MHz 68000 CPU

Memory

RAM: 256k Bytes with byte parity, expandable to 2M Bytes via P2
PROM: four 28 pin sockets are provided for 2764-type EPROMs.

Input/Output

two programmable RS423 serial I/O channels with RS-232 pinout
five programmable 16-bit Timers
one 16-bit input port

Multibus 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 Temperature: 0-55 C

2. Programming the Sun 68000 Processor

2.1. Overview

This section describes the programming characteristics of the Sun 68000 board. For programming information on the components on the Sun 68000 board, consult the manufacturer's data sheets for the respective device. In the following, the notation *0x* indicates that an address or number is in hexadecimal.

2.2. Physical Address Space

The 24-bit address space of the 68000 on the Sun 68000 Board is divided into eight parts of 2 MBytes each. The following table describes these address spaces, with addresses in hex, together with access restrictions.

0x000000 - 0x1FFFFF: Mapped address space. The mapped address space is divided into 2K pages that can be mapped into 0-wait state RAM, Multibus Memory, or Multibus I/O space on a per-page basis (see section 2.7.5).

0x200000 - 0x3FFFFF. PROM0 on Read, Bootstate on Write (see section 2.4).

0x400000 - 0x5FFFFF. PROM1 on Read.

0x600000 - 0x7FFFFF. 8274 UART. Accessed as a Byte device. Channel A data register is at location 0x600000, command register at 0x600002, Channel B data is at 0x600004, and B command is at 0x600006.

0x800000 - 0x9FFFFF. 9513 Timer. Accessed as a Word device. Location 0x800000 is the Data register, 0x800002 the Command register.

0xA00000 - 0xBFFFFF. Page map. Word-only on write. The page map entry used to map virtual address X is addressed at virtual address X + 0xA00000. (Actually, the low-order 11 bits are ignored, except that the 68000 requires the low-order bit to be zero.) The context register and segment map entries for virtual address X should be set up before accessing the page map entries since the segment map entries determine which page map entries are accessed.

0xC00000 - 0xDFFFFF. Segment map. Word-only on write. The current value of the context register determines which group of segment map entries are addressed. The segment map entry used to map some virtual address X is addressed at virtual address X + 0xC00000. (Actually, the low-order 15 bits are ignored, except that the 68000 requires the low-order bit to be zero.) When read, the high-order 4 bits of any segment map entry give the current value of the context register.

0xE00000 - 0xFFFFFF. Context register (write), input port (read). Context register is loaded from the high 4 bits of the shortword. Input port returns the 16-bit value of the parallel input port.

2.3. Reset

There are three types of reset: Power-On Reset, Watchdog Reset, and 68000 Reset.

Power-On Reset. Power-On Reset (POR) is active for at least 100 milliseconds after the power

supply voltage reaches 4.5V. POR forces boot state, resets the 68000 processor, the Intel 8274 UART, parity enable state, and it asserts Multibus INIT. The 9513 timer is not reset by POR and needs to be initialized in software. However, the 9513 timer chip has an internal power-on reset that will initialize the chip whenever the power supply voltage is less than 3V.

Watchdog Reset. The Sun 68000 Board features a watchdog timer capability. When correctly initialized by software, the watchdog timer generates a signal equivalent to power-on reset (POR) in the case the 68000 halts. The result of a watchdog reset is identical to a POR, except for the 9513 timer chip that is not affected by watchdog resets. Thus the 9513 timer chip can be used to distinguish POR from watchdog resets.

68000 Reset. When the 68000 executes a Reset Instruction, it resets the UART chip and asserts the Multibus INIT line. No other devices are affected. Specifically, parity enable state is not changed and boot state is not set.

2.4. Booting

"Boot State" is entered by power-up reset or watchdog reset. In boot state the normal operation of the board is changed as follows:

1. PROM0, normally residing at address 0x200000, overlays RAM starting at location 0. At the same time, it is also accessible in its normal location. 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 mapped address space is disabled. However, write access to mapped address space is possible to allow initialization of memory.

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 can occur as soon as the interrupt mask in the status register is lowered to allow them.

Boot state is exited by writing once to any location in the PROM0 address space. Writing to PROM0 address space also enables or disables parity checking, see section 2.6. The only way to return to boot state is to force a watchdog reset.

The boot software should initialize the memory management tables, interrupt vectors, clear parity errors in RAM, clear devices, and then enable interrupts and parity checking.

2.5. Interrupts

To understand how the 68000 Board processes Multibus interrupts, it is helpful to first review the 68000 interrupt mechanism.

The 68000 CPU has seven interrupt levels, numbered 1 through 7, with level 7 being the highest priority and level 1 the lowest priority. Interrupts are recognized for all priority levels greater than the current processor priority level 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 Multibus standard defines 8 interrupt lines, INT0 through INT7, with INT0 being the highest priority. Also, the standard recommends the 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 Multibus interrupt lines were made to correspond to the definition of the 68000. Three interrupt levels are assigned to on-board interrupt sources as follows: level 7 for the Refresh Timer, level 6 for the User Timer, and level 5 for the UART. In the standard board configuration, these levels are not connected to the corresponding Multibus interrupt lines. Thus on the Multibus only INT1 through INT4 cause interrupts on level 1 through 4. Multibus INT0 is not implemented. 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 Multibus and the interrupt vector capabilities of the Multibus are not used.

2.6. Exception Handling

When a processor cycle can not be completed normally an exception is performed. Besides the exceptions caused by internal conditions, such as divide-by-0 or word-access to an odd-byte address, five external conditions can make it impossible to complete the current instruction or bus cycle. These external conditions which raise a Bus Error exception are: system space errors, segment map errors, page map errors, timeout errors, and parity errors.

System space errors are caused when a logical address greater than or equal to 0x200000 is accessed in user mode. These addresses are reserved for supervisor state to address the on-board 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. Bus timeouts accur if Multibus references are not acknowledged within approximately one millisecond. Most likely, nonexistent memory or a nonexistent device has been addressed. There are no timeouts for on-board references because the on-board bus is synchronous and all cycles are automatically acknowledged.

Parity errors occur if 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 cannot abort the cycle in which the error occured, but the next cycle. Parity errors are not detected if the following cycle references an address greater than or equal to 0x200000.

Parity checking is enabled under software control. To enable parity checking, a "1" is written to any address in the PROM0 address space. To disable parity checking, a "0" is written. Any such write also exits from Boot State (see section 2.4).

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 referenced was nonexistent. If none of the above caused the exception, then the exception was a timeout if the reference was performing a Multibus access. Finally, a bus error can indicate a parity error if the previous instruction has been a read memory cycle from 0-wait state RAM.

2.7. Memory Management

2.7.1. Overview

The Sun Memory Management Unit provides address translation, protection, sharing, and memory allocation for multiple processes executing on the 68000 processor All accesses of the 68000 CPU to on-board RAM memory, Multibus memory, and Multibus I/O space are translated and protected in an identical fashion.

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

The page size is 2048 bytes, the segment size is 32K bytes (giving 16 pages per segment), and up to 16 contexts can be mapped concurrently. The maximum logical address space for a context is 1024 pages (2M bytes). The maximum physical address space that can be mapped simultaneously is also 2M bytes.

The organization of the memory management system and the formats of its facilities are shown in the figures below.

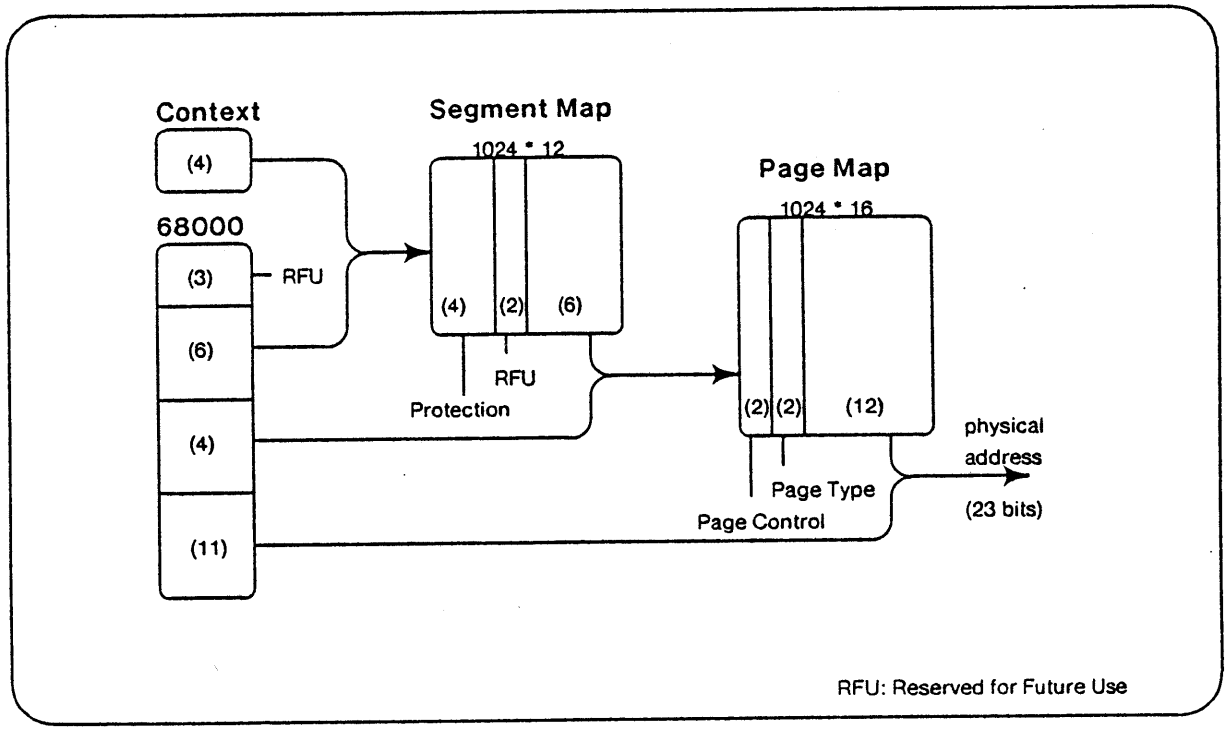


Figure 2-1: Sun 68000 Memory Management Organization

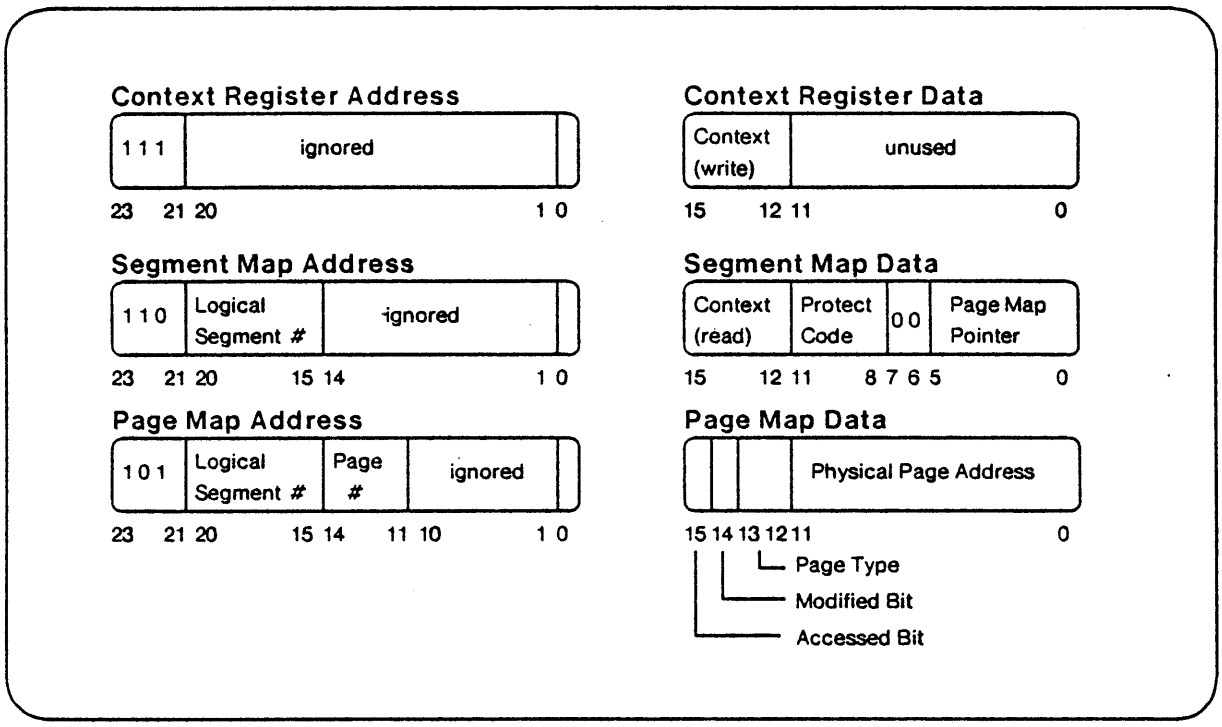


Figure 2-2: Sun 68000 Memory Management Formats

2.7.2. 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 4 bit register which can be set under supervisor control to switch between 16 sections of the segment map. 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 by writing the same values into the segment or page maps of multiple contexts.

A simple implementation of multitasking 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).

The context register is loaded by writing to location 0xE00000 with the desired value in the high 4 bits of the 16 bit data word. The currently executing instruction stream at the time of the write must be mapped into both the old and new contexts at the same address (or must be in PROM). The context register is read by reading any of the segment map entries (starting at location 0xC00000) and examining the high 4 bits.

The context register makes no distinction between user and supervisor states; interrupts and traps do not switch contexts. This requires that valid interrupt vectors must always be mapped in page 0 of each context, as well as a valid Supervisor Stack.

2.7.3. Segment Map

The segment map has 1024 entries, indexed by the 6 most significant bits of the virtual 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 6 high-order bits of a pointer into the page map and a 4 bit protection code, defined below.

Only the 64 segments of the current context may be addressed at any one time. The current value of the context register determines which group of segment map entries are addressed. The segment map entry used to map some virtual address X is addressed at virtual address X + 0xC00000. Segment map entries must be written as 16-bit words.

Each virtual address space thus has 64 segments. Each segment can be mapped to 16 pages (32K bytes) or can be made inaccessible. The 16 page map entries pointed to by the segment map entry determine whether each 2K page exists and where it is located.

2.7.4. Protection

Protection is associated with the segment map; each segment has a 4-bit protection code. Since it is not possible to represent every combination of the six possible operations (user read, write, execute; supervisor read, write, execute) in four bits, 16 of the most useful combinations are provided. 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 supervisor and the second for the user. A "-" denotes the absence of that privilege.

Code		Access by Supervisor	Access by User	Example of Use
0	------	None	None	Unused segment
1	--x---	Execute	None	System code
2	r-----	Read	None	System fixed data
3	r-x---	Read, execute	None	Mixed system code/data
4	rw----	Read, write	None	System variable data
5	rwx---	Full	None	Mixed system code/data
6	r--r--	Read	Read	User fixed data
7	rw-r--	Read, write	Read	System to user transfer
8	r--rw-	Read	Read, write	User variable data
9	rw-rw-	Read, write	Read, write	System/user shared data
10	rw-r-x	Read, write	Read, execute	User-only code
11	rw-rwx	Read, write	Full	User-generated code
12	r-xr-x	Read, execute	Read, execute	Shared code
13	rwxr-x	Full	Read, execute	System-generated, shared
14	rwx--x	Full	Execute	Proprietary code
15	rwxrwx	Full	Full	Unprotected

2.7.5. Page Map

The page map handles the 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.

The 6 bits from the segment map entry concatenated with the next 4 logical address bits from the 68000 form an index into the page map. 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 (the low 11 bits of the virtual address) to form a 23-bit physical address. In addition, a page can be declared to be in on-board memory space, Multibus memory space, Multibus I/O space, or nonexistent, according to the following values of the page type field:

```
0 - on-board memory
1 - non-existent
2 - Multibus memory
3 - Multibus I/O
```

A non-existent entry indicates an invalid page. Accessing a non-existent page causes a bus error. The address bits of such a page are ignored and can be used by software. However, the accessed and modified bits of non-existent pages are updated when the page is accessed.

Each of the physical address spaces is 23 address bits (8M bytes) large. Since on-board memory is at most 2 megabytes with two memory expansion boards, and the address space on the standard Multibus is at most 1M byte for memory and 64K bytes for I/O, some high order address bits of the page map entries will be ignored. It is the responsibility of the memory management software to provide correct table entries for a particular system configuration.

The page map entry used to map some virtual address *x* is addressed at virtual address *x* + 0xA00000. The context register and segment map entries for virtual address *x* should be set up before accessing the page map entries since the segment map entries determine which page map entries are accessed. Page map entries must be written as 16-bit words.

2.7.6. Page Control

In addition to the page mapping information, each page entry has two associated statistic bits, "accessed" and "modified", that are set whenever that page has been accessed or written into, respectively. Bit 14 of the page map entry is the "modified" bit and bit 15 is the "accessed" bit.

These bits are updated automatically on all cycles for which access has been granted by the segment map protection field. They are also updated for cycles that are aborted because the page type is 1 (non-existent). The page control bits can be set and cleared by software.

2.8. UART

The Sun 68000 Board uses an Intel 8274 dual UART. For information regarding programming the 8274, see the Intel data sheet.

Serial port A is wired as a data communication equipment (DCE) interface and features full modem control. Serial port B is wired as a data terminal equipment (DTE) interface and has no modem control. More information on interfacing the UART can be found in section 3.7.

2.9. Timer

For an explanation of the terminology used in this section, see the AMD 9513 data sheet, referenced in section 4. For further information on the setup of the timer chip see section 2.10.

The 9513 timer chip is configured with an input frequency of 5 MHz and with the FOUT output connected to the Gate 1 input. It contains five timers, whose usage is described in the following table.

Timer Usage

	Usage	Operating Mode	Normal Frequency
1	Watchdog. [1] Causes BERR/Reset.	Out=TC, Repetitive Retriggered by software	interval=2.98 msec [2]
2	User timer. OUT causes INT6.		
3	Refresh timer. OUT causes INT7.	Out=toggle, repetitive Retriggered by software	interval= 2 msec [2]
4	UART A Clock. Drives TXCA/RXCA.	Out=toggle, repetitive.	16 times UART baud rate
5	UART B Clock. Drives TXCB/RXCB	Out=toggle, repetitive.	16 times UART baud rate.

Notes:
[1] Causes Reset if Refresh Timer Out (#3) is Low.
[2] Difference between timers 1 and 3 determines Multibus timeout period

2.10. Memory Refresh, Bus Timeout, and Watchdog

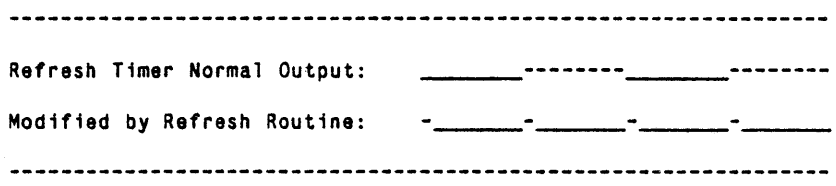
This section describes the memory refresh, Multibus access timeout, and watchdog reset features of the Sun processor board. These three functions are intimately related. They are implemented by two timers and assorted minor circuitry, and are set up and maintained by software. The two timers used are timers 1 and 3 of the 9513 System Timing Controller.

2.10.1. The Refresh Function

The refresh timer, timer 3, is set up to produce a square wave which changes state (toggles) every 2 milliseconds. Its output is connected to the non-maskable interrupt of the 68000 processor. If the processor is in a normal processing state, the non-maskable interrupt will be acknowledged, invoking a piece of code in supervisor state to accomplish memory refresh.

Sun processor boards and memory expansion boards use dynamic memories, which must be "refreshed" at least every 2 milliseconds or they lose the data stored in them. The dynamic memory is refreshed by reading out each "row" of the RAM, of which there are 128. The easiest way for the 68000 to refresh memory is to execute 128 sequential instructions (typically all NOPs). The memory is decoded in such a way that this will refresh all of memory.

The code invoked by the non-maskable interrupt must also clear the output of the Refresh timer before it returns to the user program. This changes the usual output of the timer from a square wave (changing every 2 ms) to a set of pulses (one of which appears every 2 ms, and lasts as long as it takes for the interrupt service software to run). The waveforms look like this:



The non-maskable interrupt software also deals with the Watchdog timer, as explained later.

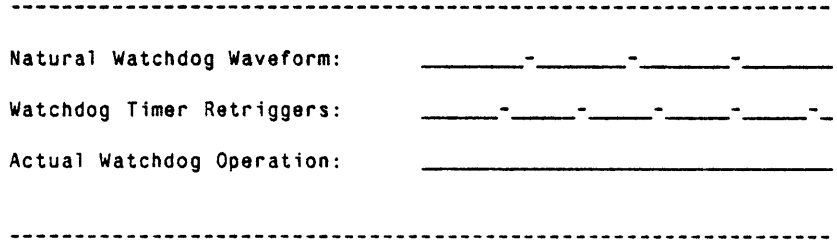
2.10.2. The Timeout Function

The Multibus is an asynchronous bus; that is, when the processor reads or writes a memory location or I/O register, there is no specific time at which the bus cycle is assumed to be done. Rather, the addressed memory or device provides an acknowledge signal (XACK) to the processor, which indicates that it is finished and another bus cycle can start.

This causes a problem for the processor board when it tries to read or write a nonexistent device or memory location. Since there is no device to provide an acknowledge signal, the processor could wait forever.

To recover from this condition, the processor board must detect that the system has not received the acknowledge signal for a long time. To do this, it uses a timer, which cancels the bus cycle if it ever times out. However, it must not time out during normal operation, or valid bus cycles would get cancelled.

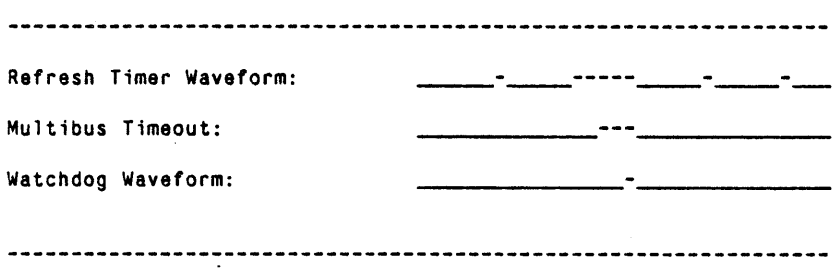
The Watchdog timer, timer # 1, is used for this. It is initialized by software at power-up time to generate a short pulse every 2.98 milliseconds. However, it is retriggered every 2 milliseconds by the non-maskable interrupt software. Each re-triggering causes it to start counting off 2.98 millisecond again. Therefore, in normal operation, it will never produce an output pulse, as shown below. However, if the non-maskable interrupt software is ever delayed by 0.98 milliseconds, it will generate an output pulse.



When a Multibus cycle is hung because the addressed location is not responding, the 68000 processor is stopped in the middle of an instruction, and cannot accept any interrupts, including non-maskable interrupts. Since it's not taking interrupts, memory is not being refreshed, and the Watchdog timer is not being re-triggered.

In this situation the following event happen: The Refresh clock toggles the non-maskable interrupt line on, but the 68000 does not accept the interrupt, since it is in mid-instruction. The Refresh timer output will remain high, rather than being reset soon by the non-maskable interrupt software. After 0.98 milliseconds, the Watchdog timer output pulses sine the watchdog timer has not been reloaded.

The output of the Watchdog timer is ANDed with the output of the Refresh timer and then drives the Bus Error pin (BERR) of the 68000 processor. This aborts the current bus cycle and forces bus error exception handling in the 68000. Since it is again executing instructions, the 68000 immediately takes the pending non-maskable interrupt. The software that services this interrupt then refreshes memory and resets the Refresh and Watchdog timers. Upon return from the non-maskable interrupt, the supervisor goes on processing the Bus Error trap. So, the complete waveforms for a Multibus timeout look like this:



Since the Refresh timer gets delayed by about .98 ms every time such a bus error occurs, it cannot be used for precision timing. However, it can be useful for other timing purposes.

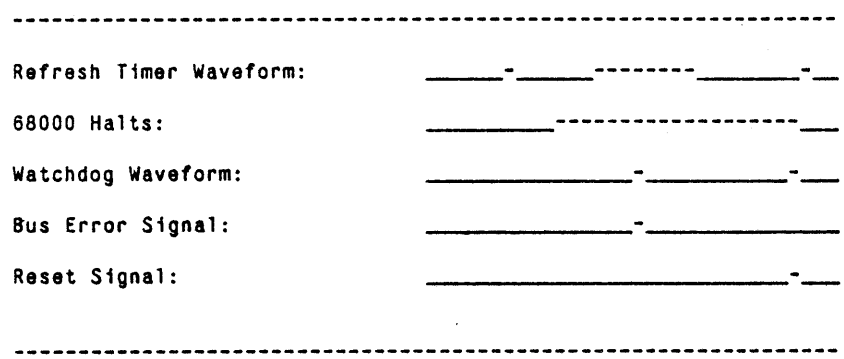
2.10.3. The Watchdog Function

The function of the watchdog is to detect a halted or out-of-control processor and to bring the system back to a functioning state. In a system where memory refresh is done in software, this is critical in retaining the content of memory. It is also useful in systems that operate without local human intervention, so they can restart themselves in case of a catastrophic error.

This function, as well as Multibus timeout recovery, is performed by the Watchdog timer. However, the Bus Error generated by the Multibus recovery circuitry is often not enough to restore control of a failing system. For example, if there are no valid interrupt vectors or if the system stack pointer is invalid, the system will halt with a "Double Bus Fault", and not respond to anything except the Reset pin. Therefore, the Watchdog timer can also generate a Reset pulse as well as a Bus Error pulse.

If the system is in a state where a Bus Error will not permit a non-maskable interrupt to occur, then the Watchdog timer still generates the Bus Error pulse, but software does not come back and reset the Refresh and Watch- dog timers. Therefore, at the next Watchdog pulse, the more serious action of pulsing RESET occurs.

Recall that the Watchdog pulse is ANDed with the Refresh clock to generate the Bus Error pulse. If a Watchdog pulse occurs and the Refresh clock is low, instead of high, a RESET pulse is generated instead. This will always restart the system, since it resets the 68000 processor the same as when the system powers up. In waveforms, this sequence appears as follows:



The relationship of the two timers is critical to the correct operation of this scheme. The Watchdog timer must have a period that is greater than the Refresh timer's, but is less than 1.5 times the Refresh timer's, in order that it will see the Refresh timer high on the first pulse and low on the second. If the Watchdog timer's period is too short, the non-maskable interrupt software might not get a chance to reset it in time to prevent its generating a Bus Error.

The PROM code invoked by RESET can reliably tell whether it was invoked on a power-up or a watchdog reset, by looking to see if the Watchdog timer has been initialized. The 9513 timer chip powers-up to a known state, different than the state it is in when a Watchdog reset occurs.

While the Watchdog facility described detects conditions which cause the Refresh timer to be reset, or which destroy certain parts of low memory (such as the Refresh vector), or which hang the 68000 in a Halted state, there are many system deadlock conditions which will not be automatically detected. However, if a Watchdog Reset does not occur, non-maskable interrupts are almost certain to be operating correctly. Software can then check for other abnormal operating conditions.

The actual implementation of the above requires careful programming of the AMD 9513 chip. In particular, the following must be observed:

- It is not possible to write to a counter within about 70 nanoseconds of a tick on that timer's input pin. If you do so, the counter gets a totally garbage value. This is independent of whether the timer is enabled or disabled. Since the main timing signal supplied to the chip is 5MHz (200 ns between ticks), writing within 70 nanoseconds of a tick is pretty common.

- Most registers on the chip are accessed by a special protocol which involves writing their register number into the "Data Pointer Register" and then doing writes or reads to the Data register. However, the Data Pointer register cannot be read back by the processor, so it is impossible to save and restore it in an interrupt routine. This means that two interrupt routines, or an interrupt routine and mainline code, cannot use most of the registers on the chip, since the higher-priority interrupt routine might be entered at any time and would alter the Data Pointer Register setting of the mainline or lower-level routine.

It is therefore necessary for the non-maskable interrupt software to retrigger the Refresh and Watchdog counters without getting caught by the 70ns glitch and without using commands that modify the data pointer.

This can be done by using the FOUT facility of the timer chip. The FOUT facility is a special divider

which can take the main input frequency and divide it by any number between 1 and 16, producing a square wave on the FOUT pin of the chip. The FOUT pin has special commands which enable and disable it without disturbing the Data Pointer Register.

The FOUT output is wired to the Gate 1 input, and Gate 1 is used as the input source for the Watchdog and Refresh timers. This allows software to turn off or on the input source to these timers at will, without disturbing the Data Pointer Register. It can be turned off just before reloading the counters (to avoid having an input source tick within 70 ns of when the timer is reloaded), then turned back on afterward. (Reloading the counters does not require the Data Pointer Register.)

The use of FOUT also provides control over the width of the pulse produced by the Watchdog timer. The pulse width is the 5 MHz input clock divided by 2 or 400 nanoseconds, which is acceptable for both Bus Error and Reset.

3. Preparation for Use

3.1. Introduction

This chapter provides information on installing the Sun 68000 Board. Included are instructions for unpacking, inspection, switch and jumper setting, and interfacing the Sun 68000 Board with other Multibus peripherals.

3.2. Unpacking Instructions

Inspect the shipping carton immediately upon receipt for evidence of damage. If the shipping carton is severely damaged, request that the carrier's agent be present when the carton is opened. If the carrier's agent is not present when the carton is opened and the contents are damaged, keep the content and carton for the agent's inspection.

It is suggested that salvageable shipping cartons and packing material be saved for future use in the event the product must be reshipped.

3.3. Installation Considerations

The board is designed for installation into a Multibus or Intel Multibus compatible backplane or cardcage. Make sure that the system provides adequate power and cooling for the Sun 68000 Board.

The Sun 68000 Board requires a 5V power supply and draws a maximum current of 5 Amps. When installing the board in an enclosed environment or under restricted airflow conditions, ensure that the internal operating temperature does not exceed 130 degrees F or 55 degrees C.

CAUTION: To prevent possible equipment damage, do not install board in a cardcage while power is on. Also, to prevent damage due to static voltages, avoid exposing the board to plastic materials.

3.4. Repair Information

To return a Sun 68000 board or a Sun 68000 Memory Expansion board for repair, obtain a return material authorization number (RMA) from the address below and send the board with the RMA number and a detailed description of the problem to the following address:

```
Sun Microsystems Inc
Att: Service Department
2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

415-960-1300
```

3.5. Diagnostic LEDs

the Sun 68000 board features two LEDs (light emitting diodes) for visual diagnostic information. During normal operation of the board, both LEDs should be turned off.

The LED closest to the card ejector displays the state of the 68000 Halt pin. This LED will be on during Reset and whenever the 68000 is in Halt state.

The other LED displays the state of the parity enable flipflop. It is turned on when parity checking is disabled. This LED is normally used by diagnostic software. This LEDs will also turn on during reset, since parity checking is disabled upon reset.

3.6. Multibus Interface

This section descibes how the Sun 68000 Board interfaces to other Multibus boards. Additional information about the operation of the Multibus can be found in the references cited in chapter 4.

3.6.1. Multibus Signal Overview

The following summary describes the Multibus signal lines.

A0..A19: Address Bus. These 20 lines carry the address of the memory location or input/output device to be accessed.

BCLK: Bus Clock. Synchronizes the bus arbitration on all Multibus masters.

BHEN: Byte High Enable. Enables 16-bit Multibus transfers when active.

BPRN: Bus Priority In. Indicates to a bus master that no higher priority bus master is requesting the bus. BPRN must be grounded for the highest priority master in a sequential priority chain.

BPRO: Bus Priority Out. Transmits the bus priority to the next lower-priority master in a serial priority resolution scheme. Bus Priority Out is daisy chained to the next Bus Priority In within the sequential bus master priority chain.

BREQ: Bus Request. Indicates that a bus master wants to get mastership of the bus. Used in parallel priority resolution schemes only. BREQ is not used for sequential priority schemes.

CBRQ: Common Bus Request. Indicates to a bus master that another bus master wants to get access to the bus. This allows a higher prioriy bus master to retain bus mastership until a lower priority master requests it, thereby increasing throughput. Note: older Multibus boards do not support CBRQ.

D0..D15: Data Bus. These are the 16 bidirectional data lines.

INIT: Initialize. Reset the system to a known state.

INT0..7. These are the eight interrupt request lines.

IORC: I/O Read Command. Read Strobe for input/output devices.

IOWC: I/O Write Command. Write Strobe for input/output devices.

MRDC: Memory Read Command. Read Strobe for memory.

MWTC: Memory Write Command. Write Strobe for memory.

XACK: Transfer Acknowledge. Indicates that a slave has completed the specified read or write operation.

3.6.2. Multibus Interface Overview

The Sun 68000 Board has a number of characteristics that need to be observed to successfully interface it to other Multibus boards.

Data Transfers. The Sun 68000 Board allows word and byte transfers via the Multibus.

Byte Order. In order to offer a consistent model for 68000 programmers, the Sun 68000 board generates 68000 byte order on the Multibus. This means that the low-order or the even byte is placed in D8 through D15, whereas the high-order or odd byte is placed in bits D0 through D7. If the Sun 68000 board communicates with a byte-organized Multibus device, it is typically necessary to reverse the byte-order in software.

Addressing. The Sun 68000 Board drives the 20 address lines on the standard Multibus. Using the 20 address lines of the standard Multibus, the board can address up to 1 Megabyte of memory and 1 Megabyte of input/output locations.

Interrupts. The Sun 68000 board operates the 68000 in auto-vector mode, that is interrupt vectors are generated internally to the 68000 and peripheral devices do not provide interrupt vectors. The interrupt vector capabilities of the Multibus are not used.

Multimaster Operation. The Sun 68000 board has full multi-master capabilities that allow it to share the Multibus with 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. All on-board devices and memory are private to the board and cannot be accessed from the Multibus.

Clock Generation. The Multibus defines two clocks, Bus Clock (BCLK) and Constant Clock (CCLK). Bus Clock is used for the multimaster arbitration logic. Constant Clock provides a centralized clock source to generate timing delays on peripheral boards. The bus standard specifies both clocks to be 10 Mhz or less and that they are driven by a single busmaster only. The standard configuration of the Sun 68000 Board drives both clocks.

Init Generation. The Sun 68000 Board contains an on-board precision voltage reference for power-on reset whenever the supply voltage falls below 4.5 Volt. Power-on reset also drives Multibus INIT. In addition, Multibus INIT can be generated by executing the 68000 RESET instruction. For a desciption of alternate INIT options, see section 3.6.8 below.

Bus Timeout. If the 68000 accesses the Multibus and does not receive an data transfer acknowledge within 1 to 3 milliseconds, the access will be aborted by a bus error. This timeout period includes the Multibus aquisition time. Thus, if other bus masters lock up the Multibus for more than 1 millisecond, timeouts can occur.

P1/P2 Connector. The Sun 68000 Board uses both the P1 and P2 cardedge connectors. P1 implements a standard Multibus interface, whereas P2 carries the Sun memory expansion bus.

3.6.3. Byte order and A0 Address Generation

The Sun 68000 Board uses 68000 Byte order on the Multibus to offer a consistent memory model for the 68000. Notice that the 68000 byte order is incompatible with the Multibus byte order in that the 68000 numbers the upper byte (Data bits 8 through 15) the even byte whereas the Multibus calles the lower byte (Data bits 0 thru 7) the even byte.

	D15 .....D8	D7 .....D0
68000 Byte Order	Byte 0	Byte 1
Multibus Byte Order	Byte 1	Byte 0

In a multimaster system, masters with different byte orders might be connected to the same bus. Care needs be taken to avoid confusion when addressing byte operands.

3.6.4. Addressing

The Sun 68000 Board addresses the Multibus via the on-board page map. By setting the page type in the page map (bits 12:13) to 2 or 3, the processor can address the Multibus memory and I/O space, respectively. The physical address actually sent to the Multibus is composed of the byte address A0 (see above), the byte offset address A1 through A10, and the physical page address in the page map (bits 0 to 8) forming address bits A11 through A19. Thus, the Sun 68000 Board generates a 20-bit Multibus address for both Multibus Memory space and Multibus I/O space. If a Multibus device decodes fewer address bits than the Sun 68000 Board generates, the undecoded address bits are simply meaningless. Since the Multibus Standard specifies only 16-bit addresses for input/output, the high-order four address lines are typically ignored for input/output addressing.

3.6.5. Interrupts

The default mapping of Multibus interrupts to 68000 interrupts connects Multibus interrupt levels 1 through 4 to 68000 interrupt levels 1 through 4 and ignores all other Multibus interrupts. These interrupts are level-triggered and are acknowledged by the 68000 in autovector mode. Multibus interrupt vectoring and the Multibus INTA signal are not used.

Multibus interrupt level assignments can be modified via Jumper J904. See section 3.9.

3.6.6. Multimaster Operation

The Sun 68000 Board can be used both in single master systems and in Multimaster systems. The following discusses the default configuration of the Sun 68000 Board for multimaster operation.

1. The Sun 68000 Board is the *highest* priority master. This means that the Sun 68000 Board is installed into the highest priority card slot of a cardcage, and that other masters are installed into lower priority slots.

2. The BPRN (Bus Priority In) jumper of the Sun 68000 Board is installed to terminate the priority chain. All other masters receive their BPRN from the priority daisy chain.

3. The CBRQ (Common Bus Request) jumper of the Sun 68000 Board is installed if any other bus master does not support CBRQ. The CBRQ jumper forces the processor board to release bus mastership after every bus cycle. This will also cause three additional wait states for every Multibus access from the Sun 68000 board. However, if all other bus masters support Common Bus Request then the CBRQ jumper is not installed. In this case, the Sun 68000 Board will retain bus mastership until any other master requests it.

4. BCLK (Bus Clock) and CCLK (Constant Clock) is driven by the Sun 68000 Board. Other masters should not drive these clocks.

5. INIT (Initialize) is driven by the Sun 68000 board. Other boards should not drive INIT.

Other relevant information concerning multimaster system configuration is contained in the chapters about clocks and Multibus initialization.

Single Master Systems: For Multibus systems in which the Sun 68000 Board is the sole bus master, the BPRN (Bus Priority In) Jumper needs be installed to enable access to the Multibus. Also, the CBRQ (Common Bus Request) jumper should be not be installed in order to maintain continuous bus mastership. The BCLK (Bus Clock) and the CCLK (Constant Clock) jumpers should be installed such that the Sun 68000 Board drives both of these clocks.

3.6.7. Multibus Clock Generation

The Sun 68000 board is factory wired to provide the Multibus *Bus Clock* and the Multibus *Constant Clock* at a frequency of 9.8304 MHz. If the Sun 68000 board is used in a system where another board provides the bus clock or the constant clock, the respective clock needs to be disabled. See section 3.9.

3.6.8. Multibus Initialization

In its standard configuration, the Sun 68000 board drives INIT to the Multibus and does not receive INIT from the Multibus. In this situation, Multibus INIT is active during power-on reset, watchdog reset, and 68000 Reset. For this configuration, jumper J901-3::4 is installed. This option does not fully conform to the Multibus standard in that B.INIT is a Tri-state driver instead of Open Collector.

Another Reset Option exists by opening J901-2::3 and installing J901-1..2. In this option, Multibus INIT is driven from power-on or watchdog reset on the 68000 board. At the same time, the Multibus can also reset the board by asserting INIT. However, the 68000 cannot drive Multibus INIT directly by executing RESET instruction.

3.7. J1-Connector

The J1 connector serves as the interface for the two serial ports. Two interfaces with standard RS-232 pinout are achieved by splitting the 50-wire flat cable into two 25-wire sections and mating each section with a DB-25 pin flat cable connector.

Pin 1 of the 50-pin connector is Pin 1 of Serial Port A, which is prewired to be a data communication equipment (DCE) interface. Serial Port B starts at pin 25 of the 50-pin connector and

is prewired as a data-terminal equipment (DTE) interface. The following tables gives the pin assignment of the serial port connectors and their relationship to the J1 pinout. Note that serial port A implements a complete asynchronous modem interface whereas the serial port B implements no modem control. To change a port from a DCE to a DTE interface or the other way around, a null modem can be used.

Serial Port A <DCE>

SIGNAL	DB-25	J1-PIN	UART	TYPE
TXD	2	J1-03	RXDA	IN
RXD	3	J1-05	TXDA	OUT
RTS	4	J1-07	CTSA	IN
CTS	5	J1-09	RTSA	OUT
DSR	6	J1-11	DTRA	OUT
GND	7	J1-13	GND	GND
DTR	19	J1-14	DCDA	IN

Serial Port B <DTE>

SIGNAL	DB-25	J1-PIN	UART	TYPE
TXD	2	J1-28	TXDB	OUT
RXD	3	J1-30	RXDB	IN
GND	7	J1-38	GND	GND

3.8. J2-Connector

The J2 Connector brings out the 16-bit parallel input port, as well as some additional control lines to allow connection of reset switches and the halt indicator. The 16-bit input port is typically used for configuration control or to connect special devices.

SIGNAL	J2-PIN	DESCRIPTION
IN0	J2-01	Input Bit 0
IN1	J2-03	Input Bit 1
IN2	J2-05	Input Bit 2
IN3	J2-07	Input Bit 3
IN4	J2-09	Input Bit 4
IN5	J2-11	Input Bit 5
IN6	J2-13	Input Bit 6
IN7	J2-15	Input Bit 7
IN8	J2-17	Input Bit 8
IN9	J2-19	Input Bit 9
IN10	J2-21	Input Bit 10
IN11	J2-23	Input Bit 11
IN12	J2-25	Input Bit 12
IN13	J2-27	Input Bit 13
IN14	J2-29	Input Bit 14
IN15	J2-31	Input Bit 15
GND	J2-02..48	All even pins are grounded

3.9. Jumper Summary

Jumpers are identified by location and a pair of pin numbers, e.g. J901-1..2 indicating location J901, pins 1 and 2. A ".." between the pin numbers indicates a jumper pair normally not connected, whereas a "::" indicates a jumper pair that is normally installed.

Jumper pin 1 is identified by a square pad and by a mark in the silk screen. Jumpers with more than 2 pins have odd and even pin numbers on opposite rows, with pin 1 being accross from pin 2.

Multibus Signal	Options:	Meaning:
B.INIT\	J901-1..2	Sun 68000 Board is Multibus INIT Slave.
B.INIT\	J901-3::4	Sun 68000 Board is Multibus INIT Master.
B.BCLK\	J901-5::6	Sun 68000 Board drives BCLK to Multibus.
B.BPRN\	J901-7::8	Sun 68000 Board is highest priority master.
B.CCLK\	J903-1::2	Sun 68000 Board drives CCLK to Multibus.
B.CBRQ\	J902-3..4	Sun 68000 Board allows Common Bus Requests.
	J902-1::2	Sun 68000 Board gives up Multibus Mastership after every access.

Interrupt Level	Assignment:	Meaning:
B.INT7	J904-1..2	NMI, used by refresh timer
B.INT6	J904-3..4	used by on-board Timer2
B.INT5	J904-5..6	used by on-board UART
B.INT4	J904-7::8	
B.INT3	J904-9::10	
B.INT2	J904-11::12	
B.INT1	J904-13::14	
B.INT0	J904-15::16	unused

3.10. Appendix A: Multibus Interface Signal DC Characteristics

Signal	Symbol	Parameter Description	Test	Min	Max	Unit
A0..19	VOL	Output Low Voltage	IOL = 24mA		0.5	V
	VOH	Output High Voltage	IOH = -2.6 mA	2.4		V
	CL	Capacitive Load			15	pF
BCLK\	VOL	Output Low Voltage	IOL = 64mA		0.5	V
	VOH	Output High Voltage	IOH = -15mA	2.0		V
	CL	Capacitive Load			15	pF
BPRN\	VIL	Input Low Voltage			0.8	V
	VIH	Input High Voltage		2.0		V
	IIH	Input High Current			0.06	mA
	CL	Capacitive Load			18	pF
BPRO\, BREQ\	VOL	Output Low Voltage	IOL = 20mA		0.45	V
	VOH	Output High Voltage	IOH = -0.4mA	2.4		V
	CL	Capacitive Load			12	pF
BUSY\, CBRQ\	VIL	Input Low Voltage			0.8	V
	VIH	Input High Voltage		2.0		V
	IIH	Input High Current			0.06	mA
	CL	Capacitive Load			18	pF
	VOL	Output Low Voltage	IOL = 20mA		0.45	V
	CL	Capacitive Load			12	pF
CCLK\	VOL	Output Low Voltage	IOL = 64mA		0.5	V
	VOH	Output High Voltage	IOH = -15mA	2.0		V
	CL	Capacitive Load			15	pF
D0..15	VOL	Output Low Voltage	IOL = 48mA		0.5	V
	VOH	Output High Voltage	IOH = -10 mA	2.4		V
	VIL	Input Low Voltage			0.8	V
	VIH	Input High Voltage		2.0		V
	IIL	Input Low Current			-0.2	mA
	IIH	Input High Current			-0.2	mA
	IOD	Output Disabled Current			-0.2	mA
	CL	Capacitive Load			18	pF
IORC\, IOWC\, MRDC\, MWTC\, XACK\, INIT\						
	VOL	Output Low Voltage	IOL = 64mA		0.5	V
	VOH	Output High Voltage	IOH = -15mA	2.0		V
	CL	Capacitive Load			15	pF
INT0..7	VIL	Input Low Voltage			0.8	V
	VIH	Input High Voltage		2.0		V
	IIL	Input Low Current			-0.36	mA
	IIH	Input High Current			0.02	mA
	CL	Capacitive Load			15	pF

3.11. Appendix B: Multibus Interface Signal AC Characteristics

Parameter	Description	Min	Max	Unit
TBCY	Bus Clock Period	100	103	nsec
TCCY	Constant Clock Period	100	103	nsec
TBW	Bus Clock Pulse Width	40		nsec
TBREQ	Bus Clock to BREQ\ delay		35	nsec
TBPRNS	BPRN\ to BCLK\ setup time	15		nsec
TBPRO	BCLK to BPRO delay		40	nsec
TBPRNO	Bus Priority In to Priority Out		25	nsec
TBUSY	BCLK to BUSY		60	nsec
TAS	Address Setup Time	50		nsec
TDS	Data Setup Time	50		nsec
TDSW	Data Setup Time Write	50		nsec
TAH	Address Hold Time	50		nsec
TDHW	Data Hold Time Write	50		nsec

4. References

MC68000 16-bit Microprocessor User's Manual, Motorola Inc, IMC68000UM, 1980.

8274 Data Sheet, Intel Corp., 1981.

9513 Data Sheet, Advanced Micro Devices, 1979.

R.W.Boberg, "796 Microprocessor Bus Standard", *Computer*, October 1980.

796-Microprocessor Bus Standard, IEEE, December 1981.

Intel Multibus Specification, Intel Corporation, Revised April 1981.