Sun-3 Architecture Manual

Version 1.0

Company Confidential

Sun Microsystems Inc.

14 January 1985

Table of Contents

1. Introduction

This document describes the Sun-3 architecture. It is intended as a reference for Sun-3 software, hardware, and systems implementors.

Features. The Sun-3 architecture is an extension of the Sun-2 architecture. Its main features are support of the 68020 CPU, the 68881 FPP, 8 KB pages, eight 256 MB contexts, and a 32-bit VMEbus. The Sun-3 architecture does not necessarily apply to any future CPUs.

Scope. This Sun-3 architecture manual describes all devices on the CPU Board. It does not describe devices that are on the system bus.

Implementation. The main part of this document is independent of a particular implementations of the architecture. Implementation specific data, such as timing information, need to be defined for each implementation.

Correctness. An important goal of this document is correctness. Please report any errors, omissions, or oversights immediately so they can be corrected in future revisions.

1.1. Definitions

In the subsequent description of the Sun-3 architecture the following abbreviations are used:

CPU: Central Processing Unit

DVMA: Direct Virtual Memory Access

MMU: Memory Management Unit

PMEG: Page Map Entry Group

POR: Power-On-Reset

1.2. Architecture Overview

The Sun-3 architecture is divided into three spaces: the CPU space, Control space, and Device space.

The CPU space comprises all references of the 68020 CPU in function code 7.

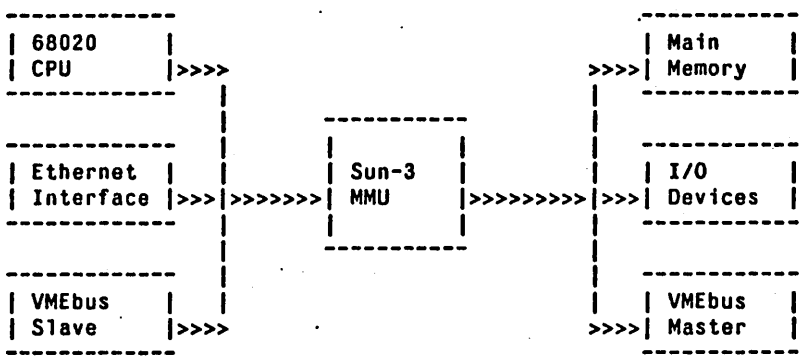
The Control space is the core of the Sun-3 architecture. It includes the Sun-3 memory management unit (the "MMU") as well as all other Sun-3 architecture extensions to the CPU, such as the bus error register, the system enable register, the diagnostic register, and the ID-PROM. The ID-PROM contains a unique serial number and indicates the implementation type of the architecture.

The Device space of the Sun-3 architecture defines what devices exist in the architecture and how they are accessed. These devices include main memory, the system bus, and I/O devices.

All CPU accesses to device space pass through the MMU and thus are translated and protected in an identical fashion. In addition, direct memory accesses by DVMA masters such as the Ethernet or the VMEbus slave interface also pass through the memory memory management and thus operate in a fully protected environment.

The figure below illustrates how the CPU, MMU, and devices are interconnected in the Sun-3 architecture. The CPU sends out a virtual address that is translated by the MMU into a physical address. The VME Master Interface, Main Memory, Video Memory, and I/O Devices are addressed with physical addresses on the right side of the MMU.

The CPU, Ethernet Interface, and VME Slave Interface arbitrate for and share the virtual address bus on the left side of the MMU. Of these three devices, the Ethernet has the highest priority, the VMEbus slave interface is the second highest, and the CPU is lowest.



1.3. Implementation Configurations

The Sun-3 architecture allows each implementations of the Sun-3 architecture to have its own configuration of devices. In addition, configurations may provide certain options in terms of main memory size and I/O devices. With this configuration flexibility comes a number of optional bits in registers. These Sun-3 implementation configurations are treated uniformly as follows:

- *Machine Configuration.* The machine type in the IDPROM indicates which devices the machine has, or has options for, and the address assignment of such devices. The result of an accesses to a physical address that is not defined for a given machine type is not specified.

- *Optional Main Memory.* Each Sun-3 machine type has a minimum and maximum main memory. The minimum main memory size on a Sun-3 is 2 megabytes, the maximum depends on the machine type. An access to memory that is addressable, but not physically present, responds with timeout/bus error.

- *Optional I/O Devices.* Optional I/O devices are devices that are defined for a given machine type, but not necessary installed on a given machine, such as an optional data encryption processor. An access to an optional I/O defined that is not physically present respond with timeout/bus error.

- *Optional Bits in Registers.* These bits are defined in the architecture, but are only used in certain implementations, such as many of the bits in the system enable register. These bits exist for all implementations but have only an affect when used in a particular implementation.

- *Unused bits in Page Map.* Unimplemented bits in the page map physical address field read back as 0s.

2. Address Spaces

The Sun-3 architecture uses three address spaces: CPU Space, Control Space, and Device Space. These address spaces are decoded with different processor function codes.

The following table describes how different CPU function codes are mapped to the CPU, Control, and Device space.

FC	Address Space
0	Reserved
1	Device Space (User Data)
2	Device Space (User Program)
3	Control Space
4	Reserved
5	Device Space (Supervisor Data)
6	Device Space (Supervisor Program)
7	CPU Space

2.1. CPU Space

CPU space consists of all cycles that use function code 7. These include coprocessor cycles, interrupt, breakpoint and ring-protection cycles.

2.2. Control Space

Control space consists of all cycles that use function code 3. This includes accesses to the memory management unit (the "MMU"), to the bus error register, the system enable register, the user enable register, the diagnostic register, the ID-PROM, and the cache if one is present.

2.3. Device Space

Device space includes all devices that are accessed by the CPU with data or program space instructions. These devices include main memory, the VMEbus, I/O devices, and so on. All devices are accessed via the MMU. This allows all devices to be protected, shared, and managed in a uniform manner in a multiprocess environment.

3. CPU Space

CPU space consists of all cycles that use function code 7. This includes coprocessor cycles, interrupt, breakpoint and ring-protection cycles.

3.1. CPU Space Cycles

In the Sun-3 architecture, address bits A16 and A17 are decoded to determine the type of CPU space cycle as follows:

TYPE	A17	A16	RESPONSE
BREAKPOINT CYCLE	0	0	BERR
RINGPROTECTION	0	1	BERR
COPROCESSOR CYCLE	1	0	DSACK/BERR
INTERRUPT CYCLE	1	1	AVEC/DSACK/BERR

Breakpoint and ringprotection cycles are terminated with Bus Error.

For Coprocessor cycles, address bits A13 through A15 are also decoded. If A13 = 1 and A14 = 0 and A15 = 0 then the reference is directed to the floating point coprocessor and terminated normally. If no floating point coprocessor is physically present, i.e. if the FPP chip is not plugged in, a timeout/bus error is generated to allow software emulation of the instruction. Other coprocessor cycles are terminated with timeout/bus error as well.

Interrupt cycles are normally terminated with AVEC (autovector) for most onboard interrupts, and with DSACK for VME and vectored interrupts, unless the interrupt vector acquisition is aborted with BERR.

4. Control Space

Control Space includes the Sun-3 memory management unit and all Sun-3 architectural extensions to the CPU. These extensions include the bus error register, the system enable register, the user enable register, the diagnostic register, the ID-PROM, and the cache if one is present.

4.1. Access to Control Space Devices

Control space devices are decoded via the high-order address bits. For IDPROM and map accesses, additional virtual address bits determine which map entry is being modified. For accesses to the page map and segment map, the value of the context register determines which context's map will be modified. Thus, for user virtual address V, the map entries are accessed as follows:

REGISTER/MAP	ADDRESS BASE	SIZE	TYPE	RELEVANT BITS
ID PROM	0x00000000 + V	BYTE	READ	V & 0x0000001F
PAGE MAP	0x10000000 + V	LONG	R/W	V & 0x0FFFE000
SEGMENT MAP	0x20000000 + V	BYTE	R/W	V & 0x0FFE0000
CONTEXT REG.	0x30000000	BYTE	R/W	
SYSTEM ENABLE	0x40000000	BYTE	R/W	
USER ENABLE	0x50000000	BYTE	R/W	
BUS ERROR REG.	0x60000000	BYTE	READ	
DIAGNOSTIC REG.	0x70000000	BYTE	WRITE	
RESERVED	0x80000000..0xF0000000			

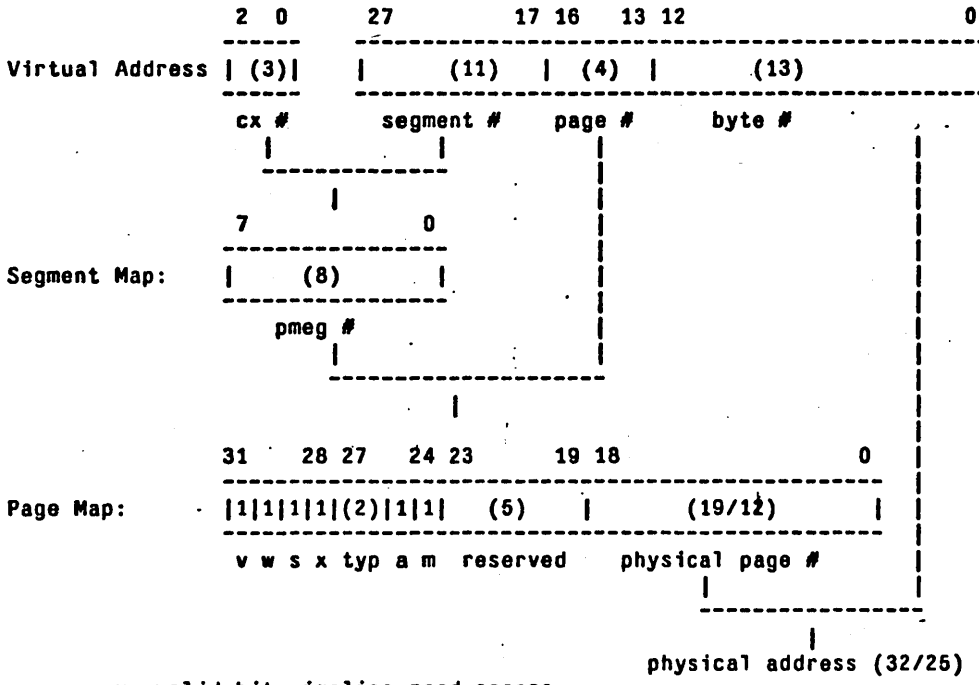
The reserved addresses are reserved for controlling cache operation in machines that have a cache. Accessing these locations has no effect if no cache is implemented.

4.2. Sun-3 Memory Management Unit Summary

4.3. Summary

```
page size:              8 KBytes
segment size:           128 KBytes
process size:           256 MBytes
# of contexts:          8
# of segments/context:  2048
# of pages/segment:     16
# of pmegs:             256
# of pages total:       4096
# of segments total:    16384
```

4.3.1. Address Translation




```
v: valid bit, implies read access
w: write access bit
s: system access bit
x: don't cache bit
a: accessed bit
m: modified bit

typ=00: onboard memory
typ=01: onboard i/o
typ=10: VMEbus 16-bit data
typ=11: VMEbus 32-bit data
```

4.4. MMU Overview

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 MMU uses a page size of 8K bytes and a segment size of 128K bytes. Eight contexts with an address space of 256M bytes each are provided.

4.5. Contexts

The Sun-3 MMU is divided into 8 distinct address spaces or "contexts". The current context is selected by means of a 3-bit *context* register. The same context applies to both user and supervisor state.

4.6. Segment Map

The segment map has 16384 entries. It is indexed by the 3 bits of the current context register and the 11 most significant bits of the virtual address, bits 17 through 27. Thus, the segment map is divided into 8 contexts of 2048 entries each. Segment map entries are 8 bits wide, pointing to a page map entry group (*pmeg*).

4.7. Page Map

The page map contains 4096 page entries each mapping an 8K byte page. Page map entries are composed of a valid bit, protection field, don't cache bit, type field, accessed and modified bits, and a page number.

The page map is divided into 256 sections of 16 entries each. Each section is pointed to by a segment map entry and is called a page map entry group, or *pmeg*.

4.7.1. Valid Bit

The valid bit means that the page entry is valid. It also allows read and execute access to the page.

4.7.2. Write Bit

The write bit allows write access to the page.

4.7.3. Supervisor Bit

If the supervisor bit is set, the read and write access protection applies only to the supervisor and no access is permitted to the user. If the supervisor bit is clear, the access protection applies both to the supervisor and user.

4.7.4. Don't Cache Bit

If this bit is set then the page referenced will not be cached. The don't cache bit is meaningful only for implementations of the Sun-3 architecture that include a cache. In machines without a cache the bit can be read and written but has no effect.

4.7.5. PageType

The 2-bit page type field provides for four physical address spaces, each starting at a physical address of 0. The four types are:

```
PMAP<27..26>: TYPE
        0 - Main Memory
        1 - I/O Devices
        2 - VMEbus 16-bit data
        3 - VMEbus 32-bit data
```

4.7.6. Statistics Bits: Accessed and Modified

The accessed and modified bits are set, as the name implies, whenever a page is accessed or modified (written into). The statistics bits are automatically updated for all cycles except if the page is invalid or protected. The statistic bits may not be correct if a cache is present.

```
PMAP<25>: ACCESSED
        0 - Not Accessed
        1 - Accessed
PMAP<24>: MODIFIED
        0 - Not Modified
        1 - Modified
```

In Sun-3 implementations with a cache, the accessed and modified bits are only updated on memory accesses that "miss" the cache. The section "MMU Access Bit" and "Modified Bits for the Cache and MMU" in the Cache chapter further discuss statistic bit updates in systems with a cache.

4.7.7. Reserved Field

The reserved field in the page map has no function. It can be written into, but it always reads back as 0.

```
PMAP<24..19>: RESERVED
```

4.7.8. Physical Page Number

The page number field in conjunction with the byte address generates the physical address. The page number field is either 12 bits or 19 bits wide. In conjunction with the 13-bit physical byte number, the 12-bit page number field generates a 25-bit physical address, whereas the 19-bit page number field generates a 32-bit physical address. In case of the 12-bit page number field, the unused bits <18..13> have no function; they can be written into, but they always reads back as 0.

```
PMAP<18..0>: PAGE NUMBER
```

4.8. ID PROM

The purpose of the ID PROM is to provide information about the machine. This includes basic information on the machine type, a unique serial number for software licensing and distribution, a unique Ethernet address, the date of manufacturing, and a checksum. In addition, the ID PROM stores configuration data for the machine.

The ID PROM is a 32 byte bipolar PROM that is not modifiable.

REGISTER	ADDRESS	SIZE	TYPE
ID PROM 0	0x0000	BYTE	READ-ONLY
ID PROM 1	0x0001	BYTE	READ-ONLY
ID PROM 2	0x0002	BYTE	READ-ONLY
...	...	...	...
ID PROM 0x1F	0x001F	BYTE	READ-ONLY

The content of the ID PROM is as follows:

Entry	Field	
(1)	Format	1 Byte
(2)	Machine Type	1 Byte
(3)	Ethernet Address	6 Bytes
(4)	Date	4 Bytes
(5)	Serial Number	3 Bytes
(6)	Checksum	1 Byte
(7)	Reserved	16 Bytes

0X = SUN 2
1X = SUN 3

X1 = MULTIBUS
X2 = VME

11 = carrera (160)
12 = M25 (50)
13 = SIRIUS (260)
14 = prism (110)
17 = SUN3F (60)

In detail:

(1) *Format*. The format of the ID PROM.

(2) *Machine Type*. A number specifying an implementation of the architecture.

(3) *Ethernet Address*. This is the unique 48-bit Ethernet address assigned by Sun to this machine.

(4) *Date*. The date the ID PROM was generated. It is in the form of a 32-bit long word which contains the number of seconds since January 1, 1970.

(5) *Serial Number*. This is a 3-byte serial number.

(6) *Checksum*. The checksum is defined such that the longitudinal XOR of the first 16 bytes of the PROM including the checksum yields 0.

(7) *Reserved*. This field will be specified in a future revision of this document.

4.9. System Enable Register

The System Enable Register enables system facilities and allows booting. This register can be read and written under software control and is cleared on power up (hardware reset) and watchdog reset, but not upon CPU reset. Bits are assigned as follows:

Initialization: cleared on power-up-reset

REGISTER	ADDRESS	DATA	TYPE
SYSTEM ENABLE	0x40000000	BYTE	READ/WRITE

The fields of the system enable register are as follows:

SYSTEM ENABLE REGISTER FIELDS

D0	EN.DIAG	Read back diagnostic switch
D1	(res)	reserved
D2	EN.COPY	Enable copy mode to video memory, if present
D3	EN.VIDEO	Enable video display ~~and copy mode if present~~
D4	EN.CACHE	Enable external cache if present
D5	EN.SDVMA	Enable system DVMA if present
D6	EN.FPP	Enable floating point processor if present
D7	EN.BOOT-	Enable Boot State (0 => boot, 1 => normal)

When cleared after power-up or watchdog reset, all bits are initialized to 0. In this state, boot state is active whereas all other enables are disabled.

EN.DIAG. This bit reads back the external diagnostic switch. A "0" bit read means that the switch is in its normal state (not-diagnostic), whereas a "1" means that the switch is activated (diagnostic).

EN.COPY. This bit enables the copy update mode to the video memory, if present.

EN.VIDEO. This bit enables the video signal to the video monitor, if present,

EN.CACHE. This bit enables the external cache, if present.

EN.SDVMA. This bit enables the system DVMA from the system bus, if present.

EN.FPP. This bit enables the floating point coprocessor (FPP), if present. If this bit is deasserted, then accesses to the FPP cause a ~~bus error~~ trap. If the bit is asserted, accesses are directed to the FPP. In the later case, if no FPP is present, then the access still will result in bus error.

EN.BOOT. Boot state forces all supervisor program fetches to the EPROM device independent of the setting of the memory management. All other types of references are unaffected and will be mapped as during normal operation of the processor.

4.10. User DVMA Enable Register

On implementations of the architecture that allow user DVMA, this register controls which contexts have DVMA access. For each context, a separate enable bit is provided.

Initialization: cleared on power-up-reset

REGISTER	ADDRESS	DATA	TYPE
USER DVMA EN.	0x50000000	BYTE	READ/WRITE

The fields of the user DVMA enable register are as follows:

SYSTEM ENABLE REGISTER FIELDS

D0	EN.CX0
D1	EN.CX1
D2	EN.CX2
D3	EN.CX3
D4	EN.CX4
D5	EN.CX5
D6	EN.CX6
D7	EN.CX7

When cleared after power-up or watchdog reset, all bits are initialized to 0. In this state, all user DVMA is disabled.

4.11. Bus Error Register

When a bus error occurs, the bus error register latches its cause to allow software to identify the source of the bus error. The bus error register always latches the cause of the most recent bus error. Thus, in the case of stacked bus errors, the information relating to the earlier bus errors is lost.

The bus error register is a read-only register.

REGISTER	ADDRESS	DATA	TYPE
BUS ERROR	0x60000000	BYTE	READ-ONLY

The fields of the bus error registers are defined as follows:

BIT	NAME	MEANING
D0	0	Watchdog Reset
D1	0	
D2	0	
D3	0	
D4	VMEBERR	VMEbus Bus Error
D5	TIMEOUT	Timeout Error
D6	PROTERR	Protection Error
D7	INVALID	Invalid Page

In more detail, the bus error conditions are as follows:

- *INVALID* means that the valid bit in the page map was not set.
- *PROTERR* means that the page protection bits did not allow the kind of operation attempted.
- *TIMEOUT* results from accessing non-existing devices, both on-board and off-board.
- *VMEBERR* indicates a VMEbus cycle acknowledged with a bus error.

4.12. Diagnostic Register

The diagnostic register drives an 8-bit LED display for displaying error messages. A "0" bit written will cause the corresponding LED to light up, a "1" bit to be dark. Upon power-on-reset, the diagnostic register is initialized to 0 causing all LEDs to light up.

Initialization: none

REGISTER	ADDRESS	DATA	TYPE
DIAGNOSTIC REG.	0x70000000	BYTE	WRITE-ONLY

5. Device Space

Device space includes all the devices of the system that are accessed through the memory management. This includes main memory, video memory, and input/output devices.

In the following, each device is described in terms of its initialization, interrupts, exceptions, reference, and register mapping.

Not all devices are present in all implementations of the architecture. Which devices are present and their physical addresses are described in the implementation section for each machine type. However, the following devices are required for all implementations:

- Main Memory
- Memory Error Register
- Interrupt Register
- EPROM
- EEPROM
- Clock

5.1. Main Memory

Main memory is the primary system memory. It has a minimum size of 2 Megabytes and it is contiguous in physical addresses. The addressing hardware decodes all bits present in the page number field of the MMU. An access to addressable but not-existing memory causes a timeout.

LOCATION	ADDRESS	DATA	TYPE
0x00000000	0x00000000	LONG	READ-WRITE
....			

In most implementations, main memory is built from dynamic RAM chips. The dynamic RAMs are refreshed in hardware.

For main memory equipped with parity checking, parity must be initialized by writing all of memory. A Parity exception is caused if parity read is different from parity written and parity checking is enabled in the parity error register.

5.2. Frame Buffer

A Sun-3 implentation has one of three frame buffer options: no frame buffer, main-memory frame buffer, and dual-ported frame buffer. These options are further described below.

Additional, external display devices and frame buffers can be added to those implementations of the architecture that include a system bus. Those external frame buffers are not within the scope of this document.

The main-memory and the dual-ported frame buffer have the same architecture. In both cases, the frame buffer is mapped to the display screen as follows:

Data bit 15 of Word 0 of frame buffer is the first visible pixel in the upper left corner of the display. Consecutive words are displayed along the horizontal scanline left to right. After <display-width> number of pixels have been displayed, the next word is displayed at the beginning of the next horizontal line, up to <display-height> number of lines. <display-width> and <display-height> are implementation constants. The display data polarity is such that "1" bits are black on the screen and "0" bits are white.

```
N = <display-width> / 16
M = <display-height>

 15            0 15           0 15           0  15           0
-------------------------------------------------------------------
| WORD 0        | WORD 1       | ...          | WORD N-1      |
-------------------------------------------------------------------
| WORD N        | WORD N+1     | ...          | WORD 2*N-1    |
-------------------------------------------------------------------
| WORD 2*N      | WORD 2*N+1   | ...          | WORD 2*N-1    |
-------------------------------------------------------------------
| ...           |   ...        | ...          |    ...        |
-------------------------------------------------------------------
| WORD (M-1)*N  |   ...        | ...          | WORD (M-1)*N-1|
-------------------------------------------------------------------
```

5.2.1. No Frame Buffer

If there is no frame buffer, then the Video Enable Bit and the Copy Enable Bit of the System Enable Register are not used. + read back at 20 MHz

5.2.2. Main-Memory Frame Buffer

In this alternative the frame buffer is resident in main memory and the video display is refreshed out of main memory.

The visible display area starts at memory address 1 megabyte and extends to the size of the display. The maximum size of the visible display area is 128 kilobytes.

REGISTER	ADDRESS	DATA	TYPE
0x000000	0x100000	LONG	READ-WRITE
....			
0x01FFFC	0x11FFFC	LONG	READ-WRITE

Relevant bits in the system enable register are the Video Enable Bit and the Copy Enable Bit. The Video Enable Bit turns the display on and off. The Copy Enable Bit is not used.

5.2.3. Dual-Ported Frame Buffer

In this configuration, the frame buffer is located in a dedicated 128K byte video memory. This video memory is dual-ported; one port performs video refresh, the second port provides processor access.

REGISTER	ADDRESS	DATA	TYPE
0x000000	0x000000	LONG	READ-WRITE
....			
0x01FFFC	0x01FFFC	LONG	READ-WRITE

Relevant bits in the system enable register are the Video Enable Bit and the Copy Enable Bit. The Video Enable Bit turns the display on and off. The Copy Enable Bit enables the copy mode (see below).

The video memory can be updated in two ways. First, it can be read and written directly like memory. As such, it is visible as a 128 KByte block of memory locations. Second, the video memory can be written in copy mode as a side-effect of writing into special region of main memory.

Main memory shadowes video memory in the range of physical addresses starting at 1 megabytes and extending for 128 kilobytes. This area of main memory is called the *copy region*. If the copy enable bit in the system enable register is set, then data written into this copy region is also written into the video memory at the same location within the 128K region. A read from the copy region returns the data in main memory and does not affect the video memory.

5.3. Memory Error Register

All Sun-3 implementations have either memory parity error detection or memory equipped with error correction. The error reporting from these memory error mechanism is performed by the memory error register described in this section.

The memory error register consists of a control and an address register. If an error occurs, the control register stores information relevant to the error. The memory error address register stores the virtual address, the context number, and the CPU/DVMA bit of the memory cycle at which the error was detected.

Errors are reported via the non-maskable level 7 interrupt. In case of multiple (stacked) memory errors, the information relating to the first error is latched in the memory error register. The interrupt is held pending and the error information in the memory error register is latched (frozen) until it is cleared (unfrozen) by a write to bits <31..24> of the memory error address register.

```
Interrupt:      level 7 Autovector
Initialization: cleared on power-up-reset
-------------------------------------------------------
ADDRESS REGISTER                DATA    TYPE
-------------------------------------------------------
0       MEMORY ERROR CONTROL    BYTE    READ-WRITE
4       MEMORY ERROR ADDRESS    LONG    READ-WRITE
-------------------------------------------------------
MEMORY ERROR ADDRESS REGISTER
-------------------------------------------------------
BIT       NAME          MEANING
-------------------------------------------------------
D<31>     DVMA-BIT      Set if DVMA cycle caused parity error
D<30..28> CX<2..0>      Context Number (3 bit)
D<27..00> VA<27..00>    Virtual Address (28 bit)
-------------------------------------------------------
```

The definition of the memory error control register depends on the error reporting mechanism and is detailled below for parity and ECC error detection.

5.3.1. Parity Error Register

For systems equipped with parity main memory, the memory error control register provides the necessary control and information to deal with parity error.

It stores the information on the byte causing the parity error, it indicates parity error interrupts pending, and it provides functions to test parity error checking.

```
-------------------------------------------------------
PARITY ERROR CONTROL REGISTER
-------------------------------------------------------
BIT     NAME                    TYPE
-------------------------------------------------------
D<0>    PARITY ERROR 00         read-only
D<1>    PARITY ERROR 08         read-only
D<2>    PARITY ERROR 16         read-only
D<3>    PARITY ERROR 24         read-only
D<4>    PARITY CHECK            read-write
D<5>    PARITY TEST             read-write
D<6>    PARITY INTERRUPT ENABLE read-write
D<7>    PARITY INTERRUPT        read-only
-------------------------------------------------------
```

The four parity error bits are set when a parity error was detected in the corresponding byte. Parity

check is set to enable parity checking on memory read cycles. Parity test is set to write parity with the inverse polarity to test the operation of the parity error circuitry. With parity test off, correct parity is generated on all memory write cycles. Parity interrupt enable enables level 7 interrupts if a parity error is detected. Parity interrupt is true if a parity interrupt is pending.

5.3.2. ECC Error Register

For systems equipped with ECC main memory, the memory error control register provides the necessary control and information to deal with ECC error. The format of the ECC error register is spelled out in the section on ECC memory.

5.4. Clock

The timer is an Intersil 7170 time-of-day clock with battery backup. The timer crystal has a frequency of 32.768 kHz. It is expected that the clock interrupt output is driven in the 100 Hz periodic mode. This clock interrupt output signal causes an interrupt request on level 5 or 7 via the interrupt register, if the respective levels are enabled.

```
Interrupt:      Level 5 or 7 autovector
Initialization: None
Reference:      Intersil 7170 Data Sheet
-----------------------------------------
REGISTER        ADDRESS DATA    TYPE
-----------------------------------------
CLOCK REG 0x0   0       BYTE    READ/WRITE
.......
CLOCK REG 0x11  0x11    BYTE    READ/WRITE
-----------------------------------------
```

5.5. Interrupt Register

The interrupt register provides for the generation of software interrupts and controls the video and clock hardware interrupts on the board. It has the following fields:

```
Initialization: cleared on reset
Interrupt:      Level 1,2,3,4,5,7, autovectored
-------------------------------------------------------
REGISTER                ADDRESS DATA    TYPE
-------------------------------------------------------
VIDEO CONTROL REGISTER  0       BYTE    READ-WRITE
-------------------------------------------------------
BIT     NAME    MEANING
-------------------------------------------------------
D0      EN.INT  Enable all Interrupts           read-write
D1      EN.INT1 Software Interrupt Level 1      read-write
D2      EN.INT2 Software Interrupt Level 2      read-write
D3      EN.INT3 Software Interrupt Level 3      read-write
D4      EN.INT4 Enable Video Interrupt Level 4  read-write
D5      EN.INT5 Enable Clock Interrupt Level 5  read-write
D6      EN.INT6 (reserved)                      read-write
D7      EN.INT7 Enable Clock Interrupt Level 7  read-write
-------------------------------------------------------
```

EN.INT. This bit enables all interrupts. If this bit is off, no interrupts will occur.

EN.INT[1..3]. These bits cause software interrupts on the corresponding level. The interrupt request caused by an EN.INT[1..3] bit stays active until software clears the corresponding bit.

EN.INT4 enables video interrupt requests on level 4. When enabled, a level 4 interrupt request is set at the rising edge of vertical retrace. The level 4 interrupt request is cleared by momentarily turning off the EN.INT4 bit.

EN.INT5 enables clock interrupt requests on level 5. When enabled, a level 5 interrupt request is set on the rising edge of the clock interrupt output. The level 5 interrupt request is cleared by momentarily turning off the EN.INT5 bit.

EN.INT6 is a reserved bit. It can be read and written but has no effect.

EN.INT7 enables clock interrupt requests on level 7. When enabled, a level 7 interrupt request is set on the rising edge of the clock interrupt output. The level 7 interrupt request is cleared by momentarily turning off the EN.INT7 bit.

5.6. EPROM

Device EPROM consists of one 27128, 27256, or 27512 type EPROM providing 16K, 32K, or 64K bytes of PROM storage, respectively.

```
Reference:      none
Interrupt:      none
Initialization: none
-------------------------------------------------------
REGISTER        ADDRESS DATA    TYPE
-------------------------------------------------------
BYTE 0          0       BYTE    READ-ONLY
BYTE 1          1       BYTE    READ-ONLY
....
-------------------------------------------------------
```

Unlike other devices, the EPROM is addressed directly with virtual address bits from the CPU. Thus, even though each 8K page must be enabled with its own entry in the page map, the physical page number in the page map is ignored and the low-order bits of the virtual address are used instead.

The EPROM device is also accessed in boot state. In boot state, all supervisor program fetches are forced to fetch from the EPROM device, independent of the setting of the memory management.

Required ?

5.7. EEPROM

Device EEPROM consists of one 2816 type EEPROM providing 2K Bytes of electrically erasable storage.

```
Reference:      none
Interrupt:      none
Initialization: none
-------------------------------------------------------
REGISTER        ADDRESS DATA    TYPE
-------------------------------------------------------
BYTE 0          0       BYTE    READ-WRITE
BYTE 1          1       BYTE    READ-WRITE
....
-------------------------------------------------------
```

To modify the EEPROM, each byte must be written separately. After writing each byte a 10 millisecond pause must be observed before the EEPROM can be read or written again.

5.8. Serial Port

Serial ports are implemented with the Zilog 8530 SCC (serial communication controller). The SCC features two high-speed, fully symmetrical and highly programmable serial channels with built-in baud-rate generators. Channel A is connected to the UART A, channel B to UART B. The clock input to the SCC is a 4.9152 MHz clock, independent of the CPU clock.

The SCC is mapped as follows:

```
Interrupt:      Level 6 Vectored (preferred) or Autovectored
Initialization: Needs to be initialized in software
Reference:      Zilog 8530 SCC data sheet
Recovery Time:  1.6 microseconds
-------------------------------------------------
REGISTER        ADDRESS DATA    TYPE
-------------------------------------------------
CH B CONTROL    0       BYTE    READ/WRITE
CH B DATA       2       BYTE    READ/WRITE
CH A CONTROL    4       BYTE    READ/WRITE
CH A DATA       6       BYTE    READ/WRITE
-------------------------------------------------
```

5.9. Keyboard/Mouse UART

These serial ports are implemented with the Zilog 8530 SCC (serial communication controller). The SCC features two high-speed, fully symmetrical and highly programmable serial channels with built-in baud-rate generators. Channel A is connected to the Keyboard, channel B to the mouse. The clock input to the SCCs is a 4.9152 MHz clock, independent of the CPU clock. Control lines are not used.

The SCC is mapped as follows:

```
Interrupt:      Level 6 Vectored (preferred) or Autovectored
Initialization: Needs to be initialized in software
Reference:      Zilog 8530 SCC data sheet
Recovery Time:  1.6 microseconds
-------------------------------------------------
REGISTER        ADDRESS DATA    TYPE
-------------------------------------------------
CH B CONTROL    0       BYTE    READ/WRITE
CH B DATA       2       BYTE    READ/WRITE
CH A CONTROL    4       BYTE    READ/WRITE
CH A DATA       6       BYTE    READ/WRITE
-------------------------------------------------
```

5.10. Encryption Processor

The Encryption processor is an AMD 8068 data ciphering processor providing high-speed NBS DES encryption. To access an internal register in the 8068, the address register must be written first. Once the address register is setup, the selected register can be accessed repeatedly.

```
Initialization: none
Interrupts:     none
Reference:      AMD 8068 data sheet.
Recovery Time:  1.6 microseconds
-----------------------------------------
REGISTER        ADDRESS DATA    TYPE
-----------------------------------------
DATA REGISTER   0       BYTE    READ/WRITE
ADDRESS REG.    2       BYTE    WRITE-ONLY
-----------------------------------------
```

5.11. AMD Ethernet Interface

The AMD Ethernet Interface uses the AMD 7990 chip. The 7990 accesses the top 16 Megabytes of the current virtual address space with a supervisor data function code. The 7990 must be configured

in BCON = 0 mode in its CSR register.

Bus cycles The 7990 must access TYPE0 space only, otherwise they will not complete and the 7990 will post a timeout error. The 7990 can also post a timeout error because of a protection error, or a parity error on read operations.

```
Initialization: reset on all resets
Interrupts:     Level 3, Autovector
Reference:      AMD 7990 data sheet.
-----------------------------------------
REGISTER        ADDRESS DATA    TYPE
-----------------------------------------
DATA PORT       0       WORD    READ/WRITE
CONTROL PORT    2       WORD    READ/WRITE
-----------------------------------------
```

5.12. Intel Ethernet Interface

The Intel Ethernet Interface uses the Intel 82586 chip. Configured in maximum mode, the 82586 accesses the top 16 Megabytes of the current virtual address space with a supervisor data function code.

The 82586 must access TYPE0 space only, otherwise it gets a bus error. The 82586 also can get a bus error because of a protection error, or a parity error on read operations. If a bus error occurs during an 82586 operation, the error bit in the Ethernet control register is set and further activity is inhibited until the 82586 is reset.

The 82586 is connected to the system in a permanent byte-reversed mode, i.e. 82586 bits 0 through 7 are connected to 68000 bits 8 through 15 and vice versa. This causes Ethernet data to be stored in memory in CPU byte order, whereas 82586 control blocks in memory are byte swapped.

Overall operation of the Ethernet Interface is controlled by the Ethernet control register that has the following definition.

```
Initialization: cleared on all resets
Interrupts:     Level 3, Autovector
Reference:      Intel 82586 data sheet.
-----------------------------------------
REGISTER        ADDRESS DATA    TYPE
-----------------------------------------
CONTROL REG.    0       BYTE    READ/WRITE
-----------------------------------------
```

The fields of the Ethernet control register are assigned as follows:

```
ETHERNET CONTROL REGISTER FIELDS
-------------------------------------------------------------
BIT     NAME                            TYPE
-------------------------------------------------------------
D0      INT     Interrupt Pending       Read-Only
D1      ERR     Error Pending           Read-Only
D2      0       0                       Read-Only
D3      0       0                       Read-Only
D4      INTEN   Interrupt Enable.       Read-Write
D5      CA      Channel Attention       Read-Write
D6      LOOPB*  Loopback                Read-Write
D7      RESET*  Reset                   Read-Write
-------------------------------------------------------------
```

INT signals Interrupt from the 82586 ~~or an error pending condition (ERR = 1)~~.

ERR indicates that a Bus Error occured during an 82586 channel operation, inhibiting further channel activity. To reset the *ERR* condition, the *RESET* bit in the Ethernet control register must be activated.

INTEN enables 82586 interrupts to the CPU.

CA signals channel attention to the 82586.

*LOOPB** controls whether the front-end encoder/decoder is configured in loopback mode (LOOPB* = 0) or connected to the transceiver cable (LOOPB* = 1).

RESET initializes the 82586 when active (RESET* = 0) and allows normal operation when inactive (RESET* = 1). It also clears the *ERR* condition when active.

5.13. VMEbus Master Interface

The VMEbus interface is dual-ported. The VMEbus Master Interface provides access from the CPU to the VMEbus, whereas the VMEbus Slave Interface provides access from the VMEbus to the CPU. Neither the slave or the master interface supports sequential access modes or read-modify-write cycles.

The VMEbus interface does not support multiple Sun-3 boards in one backplane, except for testing purposes. It does not implement an interrupter function to the VMEbus. Other specifications of the VMEbus interface are:

- Address Bus Option: A32 MASTER, A32 SLAVE
- Data Bus Option: D32 MASTER, D32 SLAVE
- Timeout Period: 100 microseconds minimum excluding bus acquisition
- Arbiter Option: ONE (single level), can be disabled
- Requestor Option: ROR (release on request)
- Interrupt Handler Options: IH(1-7)

5.13.1. VMEbus Master Interface

The Master VMEbus Interface uses two page map types: one for 16-bit data, and one for 32-bit data. For each type, three VMEbus address spaces are supported: 4 Gbytes minus the top 16 MBytes for 32-bit addressing, the top 16 MBytes minus the top 64 KBytes for 24-bit addressing, and the top 64 KBytes for 16-bit addressing.

```
Initialization: Processor Reset causes VMEbus INIT
Interrupts:     Level 1 through 7, Vectored
Exceptions:     Timeout after 200 microseconds
Reference:      Motorola VMEbus Specification
-------------------------------------------------------
TYPE    ADDRESS          ADDRESS SPACE
-------------------------------------------------------
2       32-bit           VMEbus 16-bit data

        [0x00000000]     VMEbus 32-bit address space
        [0xFF000000]     VMEbus 24-bit address space
        [0xFFFF0000]     VMEbus 16-bit address space
-------------------------------------------------------
3       32-bit           VMEbus 32-bit data

        [0x00000000]     VMEbus 32-bit address space
        [0xFF000000]     VMEbus 24-bit address space
        [0xFFFF0000]     VMEbus 16-bit address space
-------------------------------------------------------
```

5.14. VMEbus Slave Interface

The VMEbus Slave Interface provides access from the VMEbus to the CPU. The VMEbus Slave Interface causes a range of VMEbus memory addresses to be treated as though they were a range of virtual addresses generated by the processor.

There are two kinds of VMEbus DVMA in the Sun-3: System DVMA and User DVMA. Both modes have the following attributes:

- Byte, Word, and Longword transfers are supported.
- Only physically existing on-board memory (TYPE = 0) may be accessed. Access to non-existing memory or other devices is not defined.
- VMEbus Bus Error is signalled if the DVMA cycle encounters a protection error, attempt to access a page type that is not equal to 0, or on read cycles that cause a memory parity or uncorrected double-bit ECC error. The memory parity error is synchronous unlike processor parity errors. Memory parity errors are also reported to the CPU via interrupts.
- Implementations of DVMA can offer high-bandwidth burst modes that allow fast DVMA devices to increase throughput.

5.14.1. System DVMA

System DVMA responds to the lowest megabyte of the VMEbus address range in both the 24-bit and 32-bit address spaces and shifts the reference to the highest megabyte in virtual address space. System DVMA is enabled via a bit in the system enable register. System DVMA cycles use supervisor function code in accessing memory; a bus error is signalled if the page being accessed is not valid or if a write is attempted to a read-only page.

VME-Address A24, A32	Virtual Address
[0x00000000..0x000FFFFF]	[0xFFF00000..0xFFFFFFFF]

5.14.2. User DVMA

User DVMA responds to the most significant 2 GBytes of the VMEbus 32-bit address space. A user DVMA reference is mapped to the virtual address contained in bits 0 through 27 of the VMEbus address and to the context contained in bits 28 through 30 of the VMEbus address. User DVMA is enabled via the user DVMA enable register which has one bit per context. If a context is not enabled for user DVMA, then the CPU does not respond to the addresses on the VME cycle at all; this allows sharing of the upper 2 gigabytes of the VME address space with other VME devices.

User DVMA cycles use user function code in accessing memory; a bus error is signalled if the page being accessed is not valid for user access. This bus error is not visible to the CPU. VMEbus masters that expect bus error support from the CPU must then post an interrupt to the CPU and must make the appropriate information about the bus error available to the CPU.

VME-Address	Virtual Address
[0x80000000..0x8FFFFFFF]	CX=0 [0x00000000..0x0FFFFFFF]
[0x90000000..0x9FFFFFFF]	CX=1 [0x00000000..0x0FFFFFFF]
[0xA0000000..0xAFFFFFFF]	CX=2 [0x00000000..0x0FFFFFFF]
[0xB0000000..0xBFFFFFFF]	CX=3 [0x00000000..0x0FFFFFFF]
[0xC0000000..0xCFFFFFFF]	CX=4 [0x00000000..0x0FFFFFFF]
[0xD0000000..0xDFFFFFFF]	CX=5 [0x00000000..0x0FFFFFFF]
[0xE0000000..0xEFFFFFFF]	CX=6 [0x00000000..0x0FFFFFFF]
[0xF0000000..0xFFFFFFFF]	CX=7 [0x00000000..0x0FFFFFFF]

6. CPU Reset

Three types of reset need to be distinguished: Power-On Reset, Watchdog Reset, and CPU Reset.

Power-On Reset. Power-On Reset (POR) is active for 100 milliseconds minimum after the power supply voltage reaches 4.5V. POR resets the CPU and clears the System Enable register forcing boot state, and it resets the diagnostic register, lighting all the LEDs.

Watchdog Reset. The Sun-3 architecture provides a watchdog circuit which generates a signal equivalent to power-on reset (POR) whenever the CPU halts with a double bus fault. The result of a watchdog reset is identical to a POR, as far as the CPU and the system is concerned.

CPU Reset. When the CPU executes a reset instruction, it resets all on-board and off-board I/O devices that offer an external reset function. No other devices are affected. Specifically, Control Space devices such as the system enable register and the diagnostic register are not affected by CPU Reset.

7. CPU Interrupts

The devices defined in the Sun-3 architecture use autovectored interrupts, except for the SCC UARTs that use either vectored or autovectored interrupts, with vectored being the preferred implementation. Devices on the VMEbus use vectored interrupts. A list of the interrupt assignments is in the table below:

Level	Device(s)
7	Parity Error or Clock
6	SCCs
5	Clock
4	Video
3	Ethernet or System Enable Register 3
2	System Enable Register 2
1	System Enable Register 1

8. The Sun-3 Cache Architecture

8.1. The Sun-3 Cache: Its Structure and Operation

The Sun-3 cache architecture describes the cache structure for a whole class of systems, extending beyond systems with 68020 processors. All of the caches encompassed by this architecture have a common structure and common operation. In particular, they all are direct mapped caches (i.e., one way set associative). They vary only in the number of cache blocks and, of course the cache access timing (not an architectural parameter).

This cache architecture finds application in the Sirius system, the first Sun-3 workstation to incorporate a high speed local cache as a part of its memory hierarchy. Important architectural considerations result from the use of this cache. Its structure and operation are described in this appendix from this viewpoint.

The description below will generally apply to the entire class of Sun-3 caches. Examples unique to the Sirius system will be so identified.

8.1.1. The Sun-3 Cache: Overview

The Sun-3 cache is organized as a direct mapped virtual addressed cache containing 16 byte blocks (or lines). Its size is variable, from 1K blocks (16K bytes) for the Sirius system up to 8K blocks (128K bytes) for the largest allowable Sun-3 cache.

Data are organized as 8KB pages within 128KB contexts for all cache sizes. Sun-3 caches, like the MMU, support 8 virtual contexts with 28 bit virtual address spaces.

In the Sirius system, cache tags are addressed by A13:A4; in the largest allowable Sun-3 cache, tags are addressed by A16:A4. The cache tags include sufficient virtual address to define the 28 bit virtual address space, a 3 bit Context ID (CID) field, protection bits, and controls. (See below for a description of all tag bits.)

The Sun-3 cache is a Write Back cache: at any instant the cache may contain valid modified data that are not in the main memory. Modified data are only transfered to main memory upon block replacement or block flush.

All Sun-3 cache based systems are linked to memory over the 64 bit Sirius bus. ECC memory is used in all Sun-3 cache systems. Data are transfered between the cache and memory in block transfers of 128 bits, with two data transfers per memory cycle. The data path between the memory and the cache for the Sirius system is 64 bits wide. Data are stored in eight 2Kx8 static RAM's accessed by address bits A13:A3.

The Sun-3 cache may only contain data that can be obtained in units of 16 bytes over the Sirius bus. It does NOT have provision for data obtained through Programmed I/O accesses by the processor to I/O devices. (This data can only be accessed in units of 4 bytes or less.)

Since both System and User DVMA map through the MMU into main memory, the cache can contain DVMA data. In particular, it supports virtual I/O transfers (User DVMA) over the 32 bit VME bus. Note that for User DVMA transfers, the Context Identifier is obtained from VME address bits A30:A28. (See the Sun-3 User DVMA section for a complete description.)

8.1.2. Cache Tags

Sun-3 cache tags are listed below. The use of these tags in cache control is explained in the next section.

- Valid (1 bit): Self explanatory.
- Modified (1 bit): Indicates that the cache block has been modified by one (or more) Write cycles.
- Virtual Address field (up to 14 bits): Sufficient virtual address bits to define a 28 bit virtual address space. These bits are compared against the access virtual address for all read or write operations to the cache, and for some cache Control Space operations. For the 16K byte Sirius cache, the Virtual Address field is A27:A14; for the 128K byte Sun-3 cache, the VA field is A27:A17.
- Protection (2 bits): Write allowed and Supervisor access protection bits, identical to those in the MMU. The use of the Protection bits is explained below under Definition of Cache Protection.
- Context ID (CID) field (3 bits): compared against the CID register (or VME address bits A30:A28) for cache read/write operations and for some cache Control Space operations.

```
Sirius Cache Tag format (for Read/Write Tag Control Space operations):

 D31 D30 D29 D28 D27 D26 D25 D24 D23 D22 D21 D20 D19 D18 D17 D16
|---------------|---------------|---------------|---------------|
|Val|Mod|Unused |<------------Virtual Address-------------------

  D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02 D01 D00
|---------------|---------------|---------------|---------------|
 --VA-->|Wrt|Sup|Un-|<---CID--->|<----------Unused------------>|
  con't.| Prot. |used
```

8.1.3. The Cache Hit and Protection

Contexts may contain both Supervisor and User data, as indicated by the Supervisor protection bit. Within the MMU, access to common Supervisor code from two separate contexts is through two separate Page Map Entry Groups (PMEG's). Within the cache, however, common Supervisor code must be recognized regardless of the context.

The Sun-3 cache architecture therefore REQUIRES that all Supervisor code and data must have identical address mapping across all contexts. For protection consistency, the Sun-3 cache architecture further REQUIRES that if a page is marked as having Supervisor access within one context, then that page must be marked as having Supervisor access for all contexts.

Having stated this requirement, the cache hit and protection may be defined. Note that the cache protection checking must be a simple extension of MMU protection checking. No differences in results (except performance) should be discernable whether a Sun-3 system is run with its cache enabled or disabled.

8.1.4. Definition of a Cache Hit

The cache hit is defined as follows. The source for a cache request may be either a processor or DVMA device; in both cases, a full 28 bit virtual address (extended for DVMA, if necessary) is assumed. The source context is either the Context ID register or address bits A30:A28 for VME User DVMA. There are two requirements for a cache hit. First, the source address A27:A4 must match the cache virtual address tags plus the cache block address. For Sirius, the tags match A27:A14, and the cache block is addressed by A13:A4.

Second, either the source context must match the cache Context ID tags, or the cache Supervisor protection tag must be set. In the first case, the cache hit is within the same context, regardless of whether the cache request is a User or Supervisor request. In the second case, the cache hit definition allows a source request to access common Supervisor code within the cache, regardless of the source's context.

The concept of a cache hit has meaning for all read or write bus cycles to memory (Type 0 access). In addition, it also applies for the Block Copy (Read) and Block Copy (Write) Control Space operations.

8.1.5. Definition of Cache Protection

The cache protection is defined as follows. First, no cache protection violation can result unless there is a cache hit. Second, if the source access is a User request, a protection violation results on a hit if either the cache block has a Supervisor protection tag, or if the source attempts to write into a cache block whose Write protection tag is reset. Notice, as a result, that a User request from one context which matches a Supervisor cache block in another context will terminate with a protection violation. Third, if the source access is a Supervisor request, a protection violation results on a hit only if the source attempts to write into a cache block whose Write protection tag is reset.

A protection violation terminates the bus cycle with a bus error, while setting the Protection Error bit in the Bus Error register (on CPU bus cycles). Protection checking in inhibited on all Control Space operations.

8.1.6. Enabling the Cache

The "Enable External Cache" bit, D4 of the System Enable register, determines whether the cache is enabled for Read and Write accesses. In Boot state, the cache is disabled. If disabled, all cache accesses "miss" the cache, no cache blocks are written back to memory, and memory data are directly read from or written to main memory. The Control Space operations for the cache, however, remain unaffected by the Enable bit.

8.1.7. Cache Access and Block Replacement

Whenever a normal Device space data access is initiated by either the processor or through DVMA, the cache is accessed if it is enabled. If a cache "hit" occurs, data are directly read from (or written to) the cache, assuming a valid protection check. If a cache "miss" occurs, then an I/O transfer, control register access, or main memory access is initiated, depending on the Page Map. If a main

memory access is required, then the sequence of memory accesses that follow depend on the state of the cache block that is being replaced and whether the access is to a "Don't Cache" page.

If the access is to a "Don't Cache" page (see Cache Consistency, below) then data are directly read from or written to main memory without disturbing the cache. All Sun-3 cache systems operate with ECC memory, and the check bits for ECC are defined over 64 bits. So a "Don't Cache" write to memory (four bytes or less) requires a memory read before check bits can be generated on the write data. On Sirius, the 64 bits of read data are merged with the new write data on the memory board.

The remainder of this cache access description assumes that the data may be cached. The description is for a cache access on the Sirius system; some details may change for other Sun-3 cache implementations.

Upon detecting the cache miss, a memory Read for the new cache block is begun, regardless if the cache access were a Read or Write access. (This memory Read is aborted if the MMU translates the address to some non-memory device, if a protection check occurs, or if the access is a write to a "Don't Cache" page.) If the cache block being replaced has been modified, the virtual address of this block is translated through the MMU and compared with the real address for the memory Read request, which is held in the Sirius bus address register. For the Sirius system, only translated address bits A27:A13 are compared.

If these real addresses match, then the data from memory are stale; the cache contains updated data for this block. So when the block from memory returns, it must be discarded, while the cache tags are still updated to show the new virtual address and new protection. (See also the discussion on cache data consistency, below.)

Whether the real addresses match or not, the old modified block data from the cache, the translated address for this block, and controls are loaded into Sirius Write Data buffers before the memory access for the Read data completes. Eventually a block Write to memory returns this data to memory.

If the cache block being replaced has not been modified, the cache controls simply await the return of the Read data from memory. The Read data from memory are always returned with the missing data included in the first 64 bit transfer. The type of cache access being made determines how the data are handled when the memory data are returned.

On a Read access, if the first 64 bit transfer is successful (i.e., no uncorrectable errors), the memory data are simultaneously passed to the processor and to the cache data RAM's to be written as new data. If the second 64 bit data transfer from memory is successful, then these data are written into the cache and the cache tags are validated. If either transfer is unsuccessful, the cache block is left invalid and the uncorrectable error is reported to the processor as an interrupt. (See also the appendix on ECC Memory.)

On a Write access, if the first 64 bit transfer is successful, the processor's write data are merged with the data from memory and written into the cache. Following a successful second 64 bit transfer from memory, the cache data update is completed and the cache tags are validated. Both the cache Modified tag and the MMU Modified bit are set active. If either transfer from memory contains an uncorrectable error, then the cache block remains invalid and the processor's Write data are lost. This error is reported to the processor as an interrupt.

Following the completion of this memory Read request, a memory Write cycle for the buffered modified cache data (if any) may begin. This transfer, which is called a "Write Back" bus cycle, must complete before any other main memory access can be made.

8.1.8. The MMU Accessed Bit

In Sun-3 systems with no cache, an MMU Accessed bit is updated on every bus cycle to memory. In Sun-3 systems with a cache, no MMU update is made if the memory Read or Write access "hits" the cache. If the operating system resets an MMU Accessed bit, this means that the bit will not be set again in cache systems until there is a cache miss for data contained in that page.

As a consequence, MMU Accessed bits may be somewhat inaccurate in reflecting page useage within Sun-3 systems with caches.

8.1.9. Modified Bits for the Cache and MMU

The cache and MMU Modified bits are coordinated as follows. If a Write access misses the cache, then the MMU Access and Modified bits plus the cache Valid and Modified bits are all set active when data are returned from main memory. For subsequent Writes to the same cache block, no tag updates are required.

If a Read access misses the cache, then the MMU Access bit and the cache Valid bits are set active when data are returned from main memory. If a subsequent Write access to this same block occurs, then special cache handling is required (for the first write only). The Write access is basically treated as a cache miss with no data transfer; both the cache and MMU Modified bits are set active at the conclusion of the bus cycle. Again, subsequent Writes do not affect the cache tags.

8.1.10. Control Space Operations for the Cache

The four cache control operations in Control Space (Function Code 3 for the Sirius system) are summarized below. These may only be issued by the processor. Address nibble A31:A28 encodes the operation. A complete description follows under "Sun-3 Cache System Controls".

Note that these operations are unaffected by whether the cache is enabled or not.

The first two operations are for cache diagnostics and initialization.

- A31:A28 = 0x8 Read/Write Cache Tags: At the addressed cache block, Read or Write the cache tags as data in a 32 bit format. Unused bits are undefined. Note that the cache address bits depend on the size of the Sun-3 cache; the Sirius cache block is addressed by A13:A4.

- A31:A28 = 0x9 Read/Write Cache Data: Read or Write the cache data at the addressed cache block, as longword (or smaller) data. This operation does not depend on the content of the cache tags. (Again, the cache block is addressed by A13:A4 for the Sirius cache.)

The third operation is a Flush, which is a single Write bus cycle performed over a set of 16 cache blocks. The Sun-3 hardware forces address nibble A7:A4 = 0x0 for the Flush command, and increments these bits during the Flush bus cycle.

The Flush operates as follows: any Modified cache block which satisfies the Flush Match criteria is written back to main memory, and all Valid matching cache blocks are invalidated.

Which Match criteria applies is determined by single bit encodings within the Write data nibble D3:D0 for the Flush operation.

- A31:A28 = 0xA, D3:D0 = 0x1 - Flush Cache Set (Context Match): Flush cache blocks in the User space whose Context ID field matches the Context ID register. (The cache block is in the User space if its Supervisor protection bit is reset.)

- A31:A28 = 0xA, D3:D0 = 0x2 - Flush Cache Set (Page Match): Flush cache blocks satisfying two Page Match criteria. The first criterion is that the cache block's virtual page address A27:A13 must match the processor's address A27:A13. For the general Sun-3 cache, this address match may be any combination of cache addressing and Virtual Address field comparison. For the Sirius system, A13 forms part of the cache address, and A27:A14 are the Virtual Address field. The second criterion is that either the cache block's Supervisor protection bit be active or that the cache block's Context ID field matches the Context ID register.

- A31:A28 = 0xA, D3:D0 = 0x4 - Flush Cache Set (Segment Match): Flush cache blocks satisfying two Segment Match criteria (similar to the Page Match criteria). The first criterion is that the cache block's virtual segment address A27:A17 from its Virtual Address field must match the processor's address A27:A17. The second criterion is that either the cache block's Supervisor protection bit be active or that the cache block's Context ID field matches the Context ID register.

The fourth Control Space operation is the Block Copy. Two Block Copy commands - a Read and a Write - are used to copy a block (16 bytes) of data between two memory locations while bypassing the cache, maintaining cache data consistency, and updating the MMU.

- A31:A28 = 0xC - Block Copy (Read): Check the cache for the Read block and write it back to memory if found; save the translated Read address in a Block Copy buffer; and update the MMU, if necessary.

- A31:A28 = 0xC - Block Copy (Write): Check the cache for the Write block and invalidate it if found; read a block from memory into the Sirius Write Data buffers using the address in the Block Copy buffer; write this block to memory at the Write address; and update the MMU.

8.1.11. Write Back Cycles, Control Space Operations, and the MMU

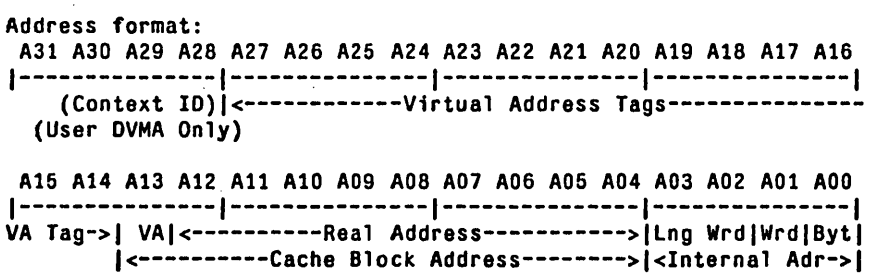
Write Back cycles and Control Space operations require special mention regarding their use of the MMU. MMU translations are required during Write Back cycles, Flushes, and Block Copy operations. During these translations, NO protection checking is performed.

The MMU is never updated on Write Back cycles or Flushes. On Block Copy (Read) operations, the Accessed bit of the block read is updated if the block is not in the cache. On Block Copy (Write) operations, the Accessed and Modified bits are both updated..

8.1.12. Application of Control Space Operations on Sun-3 Caches

The Sun-3 Cache Architecture provides for varying sizes of caches, ranging from the 1K block Sirius cache up to an 8K block cache. In this section, two tables are provided. One shows how virtual addresses access the 16K byte Sirius cache, and the second shows how flush commands compare with two Sun-3 caches, one with 1K cache blocks and the second with 8K cache blocks.

Virtual address format for the 1K block (16K byte) Sun-3 cache:



Comparison of Flush commands for two Sun-3 caches:

	16 KB Cache	128 KB Cache
Size:	1K x 16B	8K x 16B
Cache Address:	A13:A4	A16:A4
Cache Virt Addr tags:	A27:A14	A27:A17
Flush: Hardware Auto-increment	A7:A4	A7:A4
Page Flush: Command Incr.	A12:A8	A12:A8
Context Flush: Command Incr.	A13:A8	A16:A8
Segment Flush: Command Incr.	A13:A8	A16:A8

Note: As an example, a Page Flush command must increment address bits A12:A8 to flush a page from the cache.

8.1.13. Cache Error Conditions

There are two types of errors unique to cache based systems. These are Sirius Bus Time Out errors and Write Back errors.

In Sun-3 systems with caches, a time out error may be reported differently, depending on when the condition is detected. If the condition is detected while either a CPU or DVMA bus cycle is in process, then the time out is reported as a bus error to the CPU or DVMA master. On CPU bus cycles, the Time Out error bit (D5) in the Bus Error register records the cause of the bus error. See the Bus Error register description for more information.

If the time out is detected asynchronously with respect to CPU or DVMA bus cycles, then the error is reported, if enabled, as an interrupt to the processor. A time out detected during a Write Back cycle is an example. The Sirius Bus Time Out bit (D3) in the Memory Error register indicates that this error caused the interrupt.

A Write Back error results whenever a translation exception is detected for the address of a modified cache block which must be written back to memory. This asynchronous error, if enabled, is also reported to the processor as an interrupt. The Write Back Error bit (D2) in the Memory Error register records the cause of the interrupt. Note that no protection check is performed during the

Write Back address translation.

A description of the Memory Error register and more general information on reporting errors through interrupts are contained in the appendix on ECC Memory.

8.2. Sun-3 Cache Data Consistency

8.2.1. Data Consistency: Overview

The Sun-3 cache architecture defines cache operations for a series of cache based workstations, beginning with Sirius. Sun-3 caches are virtual address caches; implicit with the use of a virtual address cache is a set of system programming restrictions which MUST be adhered to in order to guarantee system data consistency.

Virtual addressing allows aliasing: the possibility of multiple virtual addresses mapping to the same real address. If a Sun-3 cache is used without architectural restrictions, any two arbitrary virtual addresses could occupy any two arbitrary cache locations and still map to the same real address. When cache blocks are modified, Sun-3 cache hardware will provide NO data consistency checking between different cache blocks. Data can become inconsistent when changes at one cache location are not seen at another cache location. Ultimately, the data at the common real address in main memory are going to include only part of the Write modifications from the several cache locations.

The Sun-3 cache architecture solves this data consistency problem by providing two distinct mechanisms. Both mechanisms require the interaction of software with special cache hardware to ensure consistent data. Briefly, the first mechanism requires that all alias addresses which map to the same data must match in their low order 17 bits (modulo 128KB) IF these data are to be cached. The second mechanism restricts data from being cached through the use of a "Don't Cache" bit which is defined for each page in the Page Map.

Both of these mechanisms are explained in more detail below.

8.2.2. Data Consistency through Modulo 128K Addressing

The first, and prefered, method of guaranteeing data consistency is through restricting the use of alias addressing for cache data. To guarantee that all alias virtual addresses map to a common cache location, it is REQUIRED that any two alias virtual addresses must match in their low order 17 bits (i.e., modulo 128K). This applies to alias addresses within the same context as well as aliases between contexts.

Note that to guarantee data consistency, it is only necessary that aliases to Write data adhere to this restriction. No requirement is placed on alias addressing to Read Only pages.

Also note that other means are available to guarantee limited data consistency, for example, in alias addressing between User contexts. If the operating system issues a series of "Flush Cache Set (Context Match)" control commands at the time of User context switches, then data consistency between User contexts is assured. The problem is that the Flush operation is relatively slow and does nothing for alias addresses within the context. Further, Supervisor cache blocks are unaffected by this Flush command.

8.2.3. Modulo 128K Addressing: Sun-3 Cache Hardware Controls

The modulo 128K rule forces all alias addresses to map to a common cache location. Within that location, the Sun-3 cache control hardware guarantees data consistency for all alias addresses. As outlined in the section (above) on "Cache Access and Block Replacement", the cache consistency control operates as follows.

Upon detecting a cache miss, a memory Read for the new cache block is begun. If the cache block being replaced has been modified, the virtual address of this block is translated through the MMU and compared with the real address for the memory Read request, which is held in the Sirius bus address register. For the Sirius system, only translated address bits A27:A13 are compared.

If these real addresses match, then the data from memory are stale; the cache contains updated data for this line. So when the block from memory returns, it is discarded, while the cache tags are still updated to show the new virtual address and new protection.

Finally, the modified cache data are written back to main memory.

8.2.4. Data Consistency through Don't Cache Pages

The second mechanism to ensure data consistency is through the use of a "Don't Cache Page" bit in the MMU Page Map. If this control bit is set for a page, then all data accesses to this page are made directly to and from main memory. In bypassing the cache, the virtual cache data consistency problem is avoided.

The primary problem in using the "Don't Cache Page" option is system performance. Direct memory data accesses are much slower than cache accesses. Consequenly, this consistency mechanism should be avoided if possible.

8.3. Sun-3 Cache System Controls

8.3.1. Sun-3 Cache System Controls: Overview

There are four cache control operations in the Sun-3 cache architecture. All are Control Space operations (Function Code 3 for the Sirius system) and may only be issued by the processor.

Two of these operations are used to directly read or write the tags or data at a particular cache location. The third operation is the Flush Set operation. This is a single Write bus cycle performed over 16 consecutive cache blocks. The Sun-3 hardware forces address nibble A7:A4 = 0x0 for the Flush command, and increments these bits during the Flush bus cycle.

During the Flush Set operation, the Tags for each of the 16 cache blocks are checked using the flush match criteria. Which match criteria applies is determined by single bit encodings within the Write data nibble D3:D0 for the Flush operation. Any modified cache block that satisfies the flush match criteria is written back to main memory, and all valid matching cache blocks are invalidated.

The fourth cache control operation is used to perform a "block copy" operation external to the cache. Two commands, a Block Copy (Read) and a Block Copy (Write), move a block (16 bytes) of

data from one memory location to another, bypassing the cache, while maintaining cache data consistency and updating the MMU.

Note that protection checking is inhibited for all cache Control Space operations.

Detailed descriptions of the cache control operations are given below.

The Sun-3 cache control operations involving reading and writing cache tags and data assume the 68020 data addressing convention. All sizes of data transfers are supported: 8, 16, or 32 bit.

An address bit value of "d" indicates "Don't Care"; "W" indicates Word address; and "B" indicates byte address.

8.3.2. Read/Write Cache Tags

```
A Read or Write command in Control Space.
At the addressed cache block, Read or Write the cache tags as
  data in a 32 bit format.
Command address format, assuming the Sirius cache:
  (Note: the cache address expands to A16:A4 for a 128KB cache)
 A31 A30 A29 A28 A27 A26 A25 A24 A23 A22 A21 A20 A19 A18 A17 A16
|---------------|---------------|---------------|---------------|
  1   0   0   0   d   d   d   d   d   d   d   d   d   d   d   d

 A15 A14 A13 A12 A11 A10 A09 A08 A07 A06 A05 A04 A03 A02 A01 A00
|---------------|---------------|---------------|---------------|
  d   d |<--------Cache Address---------------->| d   d   W   B

Data fields:
 D31 D30 D29 D28 D27 D26 D25 D24 D23 D22 D21 D20 D19 D18 D17 D16
|---------------|---------------|---------------|---------------|
|Val|Mod|Unused |<------------Virtual Address-------------------
  D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02 D01 D00
|---------------|---------------|---------------|---------------|
 --VA-->|Wrt|Sup|Un-|<---CID--->|<----------Unused------------>|
  con't.| Prot. |used

Data format:
  32 bit (logical) data, D31:D00.
    Unused bits are undefined.
  8, 16, or 32 bit transfers are supported.
  Note: The cache may be initialized by writing all cache
    Valid bits to 0.
Bus Error Conditions: None.
Interrupts Generated: None.
```

8.3.3. Read/Write Cache Data

```
A Read or Write command in Control Space.
Read or Write the cache data at the addressed cache block.
  Cache tags are not checked for this operation.
Command address format, assuming the Sirius cache:
  (Note: the cache address expands to A16:A2 for a 128KB cache)
 A31 A30 A29 A28 A27 A26 A25 A24 A23 A22 A21 A20 A19 A18 A17 A16
|---------------|---------------|---------------|---------------|
  1   0   0   1   d   d   d   d   d   d   d   d   d   d   d   d

 A15 A14 A13 A12 A11 A10 A09 A08 A07 A06 A05 A04 A03 A02 A01 A00
|---------------|---------------|---------------|---------------|
  d   d |<--------Cache Address------------------------>| W   B

Address note: A03:A02 address a Longword within a cache block.
Data format:
  32 bit  data, D31:D00.
  8, 16, or 32 bit transfers are supported.
Bus Error Conditions: None.
Interrupts Generated: None.
```

8.3.4. Flush Cache Set [Context Match]

A Write command in Control Space.
Flush any Valid cache block, from a Set of 16 blocks, which satifies the Flush match criteria. (See the Overview for a definition of Flush.)
Flush match criteria: The Supervisor protection tag must be reset (User space), and the Context ID tags must = the Context ID register (3 bits).
Command address format, assuming the Sirius cache:
(Note: the cache address expands to A16:A8 for a 128KB cache)

A31	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16
1	0	1	0	d	d	d	d	d	d	d	d	d	d	d	d

A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
d	d	<----	Cache	Address	---->	0	0	0	0	d	d	d	d		

Address note: A7:A4 are forced to 0x0 by the cache hardware before incrementing.
Data format:
(D3:D0 = 0x1) identifies this flush command as a Context Match flush. (Hardware decodes D0 = 1.)
MMU notes: No MMU protection check is done, and no update is performed on MMU Accessed or Modified bits.
Error interrupt condition:
Write Back error if the address translation for a cache block with modified data is invalid.
Notes:
A Context Flush is used to ensure cache addressing consistency whenever a new active context replaces an old context in the MMU. The Context Flush must be performed before the old context references are removed from the MMU, since the MMU is required to translate the cache blocks' virtual addresses.
Note that cache blocks in the Context with Supervisor protection are not flushed.

8.3.5. Flush Cache Set [Page Match]

A Write command in Control Space.
Flush any Valid cache block, from a Set of 16 blocks, which satifies the Flush match criteria. (See the Overview for a definition of Flush.)
Flush match criteria: Two criteria must be satisfied.
First, the cache block's virtual page address A27:A13 must match the processor's address A27:A13. For the general Sun-3 cache, this address match may be any combination of cache addressing and Virtual Address field comparison. For the Sirius system, A13 forms part of the cache address, and A27:A14 are the Virtual Address field.
The second criterion is that either the cache block's Supervisor protection tag be active or that the cache block's Context ID field matches the Context ID register.
Command address format, assuming the Sirius cache:
(Note: For a 128KB cache, the cache address expands to A16:A8, and the tag compare contracts to A27:A17)

```
 A31 A30 A29 A28 A27 A26 A25 A24 A23 A22 A21 A20 A19 A18 A17 A16
|---------------|---------------|---------------|---------------|
  1   0   1   0 |<--------Virtual Address Tag Compare-----------

 A15 A14 A13 A12 A11 A10 A09 A08 A07 A06 A05 A04 A03 A02 A01 A00
|---------------|---------------|---------------|---------------|
Tag Cmp>|<----Cache Address---->| 0   0   0   0   d   d   d   d
```

Address note: A7:A4 are forced to 0x0 by the cache hardware before incrementing.
Data format:
(D3:D0 = 0x2) identifies this flush command as a Page Match flush. (Hardware decodes D1 = 1.)
MMU notes: No MMU protection check is done, and no update is performed on MMU Accessed or Modified bits.
Error interrupt condition:
Write Back error if the translation of the address for a cache block with modified data is invalid.
Notes:
The Page Flush is used during page management to purge all references to a virtual page from the cache. It must be performed before the MMU is updated to remove the page, since the MMU is required to translate the cache blocks' virtual addresses.

8.3.6. Flush Cache Set [Segment Match]

A Write command in Control Space.
Flush any Valid cache block, from a Set of 16 blocks, which satifies the Flush match criteria. (See the Overview for a definition of Flush.)
Flush match criteria: Two criteria must be satisfied.
First, the cache block's virtual segment address A27:A17 from its Virtual Address field must match the processor's address A27:A17.
The second criterion is that either the cache block's Supervisor protection tag be active or that the cache block's Context ID field matches the Context ID register.
Command address format, assuming the Sirius cache:
(Note: the cache address expands to A16:A8 for a 128KB cache)

A31	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16
1	0	1	0	<------Virtual Address Tag Compare-------->											d

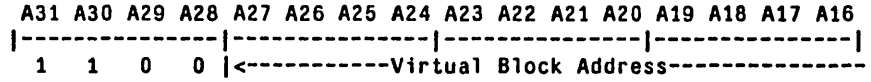
A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
d	d	<----Cache Address---->						0	0	0	0	d	d	d	d

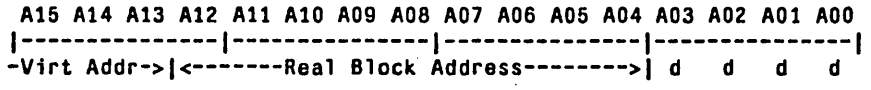
Address note: A7:A4 are forced to 0x0 by the cache hardware before incrementing.
Data format:
(D3:D0 = 0x4) identifies this flush command as a Segment Match flush. (Hardware decodes D2 = 1.)
MMU notes: No MMU protection check is done, and no update is performed on MMU Accessed or Modified bits.
Error interrupt condition:
Write Back error if the translation of the address for a cache block with modified data is invalid.
Notes:
The Segment Flush is used during page management to purge all references to a virtual segment from the cache. It is required whenever an active Page Map Entry Group (PMEG) must be replaced. It must be performed before the MMU is updated to remove the PMEG, since the MMU is required to translate the cache blocks' virtual addresses.

8.3.7. Block Copy [Read]

A Read command in Control Space.
Check the cache for the addressed Read block and write it back to memory if it is found. The cache check uses the same cache hit criteria as a memory access, including a comparison of the translated cache address with the translated address for the Read command. The translated address for the Read command is saved in a Block Copy buffer.

Command address format:





Data format:
Full 16 byte block.

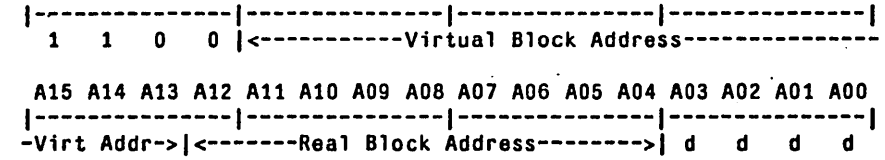
MMU notes: No MMU protection check is done. The MMU Accessed bit is updated if the Read block is not in the cache.

Error interrupt condition: None.

Notes:
If the block of Read data is modified by any instruction after the Block Copy (Read) and prior to the Block Copy (Write), then these modifications may not appear in the Write block.

8.3.8. Block Copy [Write]

A Write command in Control Space.
Check the cache for the addressed Write block and invalidate it if it is found. The cache check uses the same cache hit criteria as a memory access, including a comparison of the translated cache address with the translated address for the Write command. Read a block from memory into the Sirius Write Data buffers using the address in the Block Copy buffer, and then write this block to memory at the Write address.
Command address format:



Data format:
Full 16 byte block.
MMU notes: No MMU protection check is done. The MMU Accessed and Modified bits are updated.
Error interrupt condition: None.
Notes:
If the block of Read data is modified by any instruction after the Block Copy (Read) and prior to the Block Copy (Write), then these modifications may not appear in the Write block.

9. The Sun-3 ECC Memory Architecture

9.1. The Sun-3 ECC Memory: Overview

ECC memory is the standard memory for Sun-3 systems with caches. However, the ECC memory architecture, as specified below, is not restricted to cache based systems alone.

ECC memory requires three control registers on each memory board mapped into the Device Space. One of these registers initializes the base address of the memory board; the base address of each board is restricted to be a multiple of the size of the board. (For example, the base address of an 8 MB board must be located on an 8 MB address boundary.) Other registers capture the failing address and syndrome on single bit errors, and define the mode and operation of the ECC chips. In addition, the Memory Error Register on the processor board controls ECC error interrupts to the processor.

The contents of these registers and how they are accessed are discussed in the following sections.

9.1.1. ECC Memory Operations

The following operations are supported by ECC Memory:

- Memory Access Cycles: Read and write main memory in response to CPU, DVMA, or cache related bus cycles.
- Register Access Cycles: Read and write memory control registers. Registers are addressed in Control Space (Type 1 acccess) by a four bit board number field (A7:A4) and a four bit Register Address field (A3:A0); see below also.
- Refresh Scrub Cycles: On some implementations, an optional data scrub may be performed during memory refresh.

9.1.2. Memory Error Conditions

There are two error conditions which may be detected in ECC memory: Correctable Errors (CE's) and Uncorrectable Errors (UE's). These may be detected during either memory access cycles or refresh scrub cycles.

How these errors are reported to the processor (or DVMA master) is discussed below.

9.2. Error Reporting for ECC Memory Systems

The following sections summarize first the controls to enable error reporting and second how the error reporting differs according to the type of memory cycle and error.

9.2.1. Enabling Error Checking and Reporting

A table listing the names and functions of enable bits for errors is given below.

Enable Bit Name/Reg/Bd	Applies on Memory Cycles	Function of Enable Bit
ENABLE ECC/ Mem Enab/Mem	All Memory access, Scrub	Enables Check Bit Generation and ECC Check for memory
"	All Memory access	Enables reporting of CE's and UE's to CPU Bd
"	Scrub cycles only	Enables ONLY CE reporting to CPU Bd (UE's are inhibited)
(None)	All Memory access cycles	UE bit in Mem Error Reg always enabled and is set on UE reported to CPU Bd
ENABLE CE/ Mem Error/CPU	All Memory access, Scrub	Enables CE bit in Mem Error Reg; CE is set if the Error Status bit (D0) in the Correctable Error Reg on ANY Mem Bd is active; CE bit is reset only when ALL Error Status bits are reset
ENABLE INT/ Mem Error/CPU	All Memory access, Scrub	Enables interrupt to CPU if any memory error bit in the Memory Error Reg is active. The error bits, D0:D3, include the CE, UE, WBACK, and Sirius Time Out bits. (See Sun-3 Cache Appendix for WBACK, Time Out bits.)

9.2.2. Reporting ECC Errors

Uncorretable and Correctable errors are reported to the CPU or DVMA Master as follows.

- On DVMA cycles, the ONLY error that is reported to the DVMA Master is an Uncorrectable error on the data requested during a DVMA Read cycle. This UE is reported to the Master as a bus error. Errors resulting from DVMA Write cycles or cache related operations are not reported to the DVMA Master.

- ALL Correctable and Uncorrectable errors, if enabled, are reported to the processor as interrupts for all memory access or refresh scrub cycles, with one exception. Uncorrectable errors detected during refresh scrub cycles are NOT reported, since the virtual address for this asynchronous cycle cannot be obtained.

- Note that the Memory Address reg on the processor board does NOT capture the failing address of Correctable errors. The Correctable Error Reg on each memory board saves this address. The Memory Address Reg saves the virtual addresses for Uncorrectable errors plus other cache related errors.

9.3. Device Space Registers for ECC Memory

In this section, the registers defined in the Device Space (Type 1 access) for ECC control are examined in detail. First the use of the Memory Error and Memory Address Registers for ECC error reporting are discussed, and then the memory board Device Space registers are covered.

9.3.1. The Memory Error and Address Registers

The Device Space (Type 1) page for ECC errors includes two registers, the Memory Error Reg and the Memory Address Reg. These registers are at the same address as the corresponding registers for Parity memory.

The Memory Error register enables and records both ECC memory related and cache related errors. Its address is the same as that of the Parity Error register for Parity memory and has the same format. The low order nibble, D3:D0, records the error condition, and bit D7 indicates an interrupt. Bits D6:D4 are enable (or unused) bits.

Memory Error Register

Address	Register	Type	Data Width
0	MEMORY ERROR	Read/Write	Byte

Bit	Name	Type	Meaning
D0	CE	Read Only	Correctable Error
D1	UE	Read Only	Uncorrectable Error
D2	WBACKERR	Read Only	Write Back Error
D3	TIMEOUT	Read Only	Sirius Bus Time Out
D4	ENABLE CE	Read/Write	Enable Correctable Error Recording
D5	Unused	Read Only	
D6	ENABLE INT	Read/Write	Enable Memory Error Interrupts
D7	ERROR INT	Read Only	Memory Error Interrupt

- The CE bit records a correctable error. It functions somewhat differently from other error bits (D3:D1). If enabled by the Enable CE bit (D4), the CE bit is set if the Error Status bit (D0) of the Correctable Error Reg on ANY memory board is active (indicating a CE on that board). It resets only if ALL Error Status bits are reset. In particular, resetting the Memory Error Reg by writing to the Memory Address Reg does NOT reset the CE bit.

- The UE bit records an Uncorrectable error.

- WBACKERR and TIMEOUT error conditions are discussed in the Sun-3 Cache chapter.

- ENABLE CE enables the recording of a Correctable error as bit D0.

- ENABLE INT enables an interrupt to the processor (bit D7) for any of the error conditions D3:D0.

- The ERROR INT bit signals a level 7 interrupt to the processor.

The Memory Address Register acts in conjunction with the Memory Error Reg to freeze the failing virtual address for all error conditions recorded in Error register bits D3:D1 (not CE's). The real addresses for CE's are captured in the Correctable Error Reg on each memory board.

Memory Error bits D3:D1 and D7 are reset and the Memory Address register is unfrozen by a write to the high order byte of the Memory Address Reg.

Memory Address Register Description

Address Register	Type	Data Width
4 MEMORY ADDRESS	Read Only	Long

Bit	Name	Type	Meaning
D27:D0	VA(27:0)	Read Only	Virtual Addr. (28 bit)
D30:D28	CX(2:0)	Read Only	Context Number (3 bit)
D31	DVMA-BIT	Read Only	Set if error occured on DVMA bus cycle

9.4. Addressing Registers on ECC Memory Boards

Each memory board on the Sirius bus contains three different Device Space registers. All of these registers are addressable through a single page in the Device Space, as shown in the partial map below.

Physical Address Assignment for ECC Memory Registers

Type	Address	Device
1	[0x001E0000] (21 bit)	ECC Memory Registers

Within this page, address bits A7:A4 address the memory board number (set by jumper or backplane slot number, depending on the implementation). Address bits A3:A0 address each of the three registers, as shown below.

9.4.1. The ECC Memory Enable Register

The contents of the 16 bit ECC Memory Enable Register are shown below.

```
ECC Memory Enable Register Description:
Address     Register          Type         Data Width
-------------------------------------------------------
A3:A0=0     ECC MEM ENABLE    Read/Write   Word
A7:A4=Bd#
-------------------------------------------------------
-------------------------------------------------------
Bit     Name            Type         Meaning
-------------------------------------------------------
D4:D0   BASE ADDRESS    Read/Write   Base Address; compare
                                      with A26:A22
D5      BOARD ENABLE    Read/Write   Overall memory board
                                      enable
D6      ECC ENABLE      Read/Write   ECC Check/Gen enable
D7      SCRUB ENABLE    Read/Write   Background Scrub
                                      enable
D9:8    BOARD SIZE      Read Only    Board size encoding:
                                      00 = 4 MB
                                      01 = 8 MB
                                      10 = 16 MB
                                      11 = 32 MB
D15:D10 Reserved        Read Only    Board type indentifier
```

The register contents are used as follows:

- The Base Address is compared with Sirius bus address bits A26:A22 during each Sirius bus Memory cycle. (Note that the Sun-3 architecture requires that all real address bits A31:A0 be decoded.) If the Board Enable bit (D4) is active, the memory access request is addressed to this board if A26:A22 is greater than or equal to the Base Address and less than the Base plus the board size (4 MB to 32 MB, encoded in D9:D8).

- The Board Enable bit is initialized to zero and must be active to read or write memory.

- The ECC Enable bit enables check bit generation and all single bit and uncorrectable error reporting to the processor board.

- The Scrub Enable bit enables a background data scrub (an optional memory feature) during refresh cycles. Correctable errors detected during the scrub are reported.

- The Board Size field is a Read Only field encoding the board size, 4 MB to 16 MB.

- The Reserved field can be used to identify particular versions of the Memory board. (This field is set to 0's for the initial Sirius Memory board.)

9.4.2. The ECC Memory Enable Register: Initialization

All memory boards in the system MUST have their Base Address registers initialized on address boundaries that are multiples of the board size. As examples, an 8 MB board must be initialized with bit D0 = 0; a 16 MB board with D1:D0 = 00; and a 32 MB board with D2:D0 = 000.

9.4.3. The Correctable Error Register

The 32 bit Correctable Error Register captures the syndrome and real address of the first single bit error on a memory board. It is reset on a write to the high order byte of the CE register.

Correctable Error Register Description:

Address	Register	Type	Data Width
A3:A0=4 A7:A4=Bd#	CORRECTABLE ERR	Read Only	32 bits

Bit	Name	Type	Meaning
D0	ERROR STATUS	Read Only	Status: 1 = CE Error; 0 = No CE error
D23:D1	CE ADDRESS	Read Only	Real Address bits A25:A3 for first CE
D31:D24	SYNDROME	Read Only	Syndrome for first CE

The register contents are used as follows:

- The Error Status bit is set active when the first correctable error is detected and reset by a Write to the (read only) CE register. The address and syndrome are frozen while the bit is set.
- The CE Address field holds real address bits A25:A3 of the first CE.
- The Syndrome field holds the syndrome of the first CE.

9.4.4. The ECC Chip Diagnostic Register

Any implementation of the Sun-3 ECC memory must include control registers to initialize and test the ECC generation and check logic. In the first implementation of ECC memory on Sirius, the AMD 2960A ECC chips have been selected for this logic. Their initialization and use is described below as an example of an ECC Diagnostic Register.

It is not intended, by including this description, to restrict all implementations of Sun-3 ECC memory to use these AMD chips.

The Sirius ECC memory utilizes four AMD 2960A ECC chips per memory board. A 16 bit register internal to each of these chips controls its initialization and testing.

The ECC Chip Diagnostic Register is a logical 64 bit register consisting of the four 16 bit registers inside the 2960A chip. This register must be read or written as 16 bit words (or 32 bit longwords) on word (longword) boundaries.

See the AMD Data Book for a complete description.

ECC Chip Diagnostic Register Description:

Address	Register	Type	Data Width
A3:A0=8 A7:A4=Bd#	ECC CHIP DIAG	Read/Write	64 bits (16 or 32 bit access)

10. Appendix: Physical Address Map Example

This is an example for a physical address map of a implementation with VMEbus interface.

10.1. Physical Address Assignments

Type	Address	Device
0	32-bit	Memory Bus
	[0x00000000]	Physical Memory
	[0xFF000000]	Video Memory
1	17-bit	I/O Bus
	[0x00000000]	EPROM
	[0x00002000]	EEPROM
	[0x00004000]	Interrupt Register
	[0x00006000]	Memory Error Register
	[0x00008000]	Clock
	[0x0000A000]	Keyboard/Mouse Port
	[0x0000C000]	Serial Port
	[0x0000E000]	Ethernet Control Register
	[0x00010000]	Encryption Processor
2	32-bit	VMEbus 16-bit data
	[0x00000000]	VMEbus 32-bit address space
	[0xFF000000]	VMEbus 24-bit address space
	[0xFFFF0000]	VMEbus 16-bit address space
3	32-bit	VMEbus 32-bit data
	[0x00000000]	VMEbus 32-bit address space
	[0xFF000000]	VMEbus 24-bit address space
	[0xFFFF0000]	VMEbus 16-bit address space

11. References

AMD 7990 Ethernet Interface: AMD Order # 03378D

AMD 9513 Timer Technical Manual: AMD Order # 03402C.

AMD 8068 DCP Technical Manual: AMD Order # 04862A.

Intel 82586 EDLC Technical Manual: Intel LAN Manual

Intersil 7170 Data Sheet: Intersil 1985 Data Book

Motorola 68020 CPU Manual: Motorola MC68020UM(ADI).

Zilog 8530 SCC Technical Manual: Zilog Order # 00-2057-02.