Arete Systems Corporation

System 3000 Hardware Description

4 May 1987

1. Arete System 3000 Hardware Description

1.1 General

This document is intended as a technical description of the Arete System 3000. It identifies the major hardware components of the system and describes their function. The first sections of this document is an overview of the system and later sections detail the function and operation of the major system components.

1.2 System 3000 Overview

The Arete System 3000 is a high performance general purpose computer system. The system is intended to enhance Arete's position as a vendor of high performance on-line multi-processor UNIX systems.

The system logic is composed of a computational subsystem and an I/O subsystem. The computational subsystem is newly designed specifically for the System 3000 while the I/O subsystem is retained from the Arete 1000 family.

The system is packaged in a double width cabinet. The first bay encloses the logic card racks and power supply. The second bay encloses system peripherals. Additional peripheral bays can be attached to the main cabinet extending the capacity of the system.

2. Computational Subsystem

The System 3000 Computational Subsystem (CSS) consists of a cluster of tightly coupled processors and memory modules on a high speed system bus. Also in the CSS are I/O modules which adapt the main system bus to I/O buses.

The major components of the CSS are the Processing Modules, Memory Modules, I/O Modules, Service Processor Module, CSS Arbiter, and CSS backplane.

2.1 CSS Bus

The CSS bus is the backbone of the System 3000. It is a 64 bit wide, high speed synchronous data transfer bus. It is designed to accommodate up to 14 modules; either Processor Modules, Memory Modules, or I/O Modules.

The CSS bus is implemented as an N-port time division multiplexed transmission switch. The bus supports read and write operations between all attached modules. Read operations transfer up to 64 bits and write operations transfer up to 32 bits. The bus supports concurrent operations to and from the modules on the bus. This allows overlapping cycles on individual modules and provides enough bus bandwidth to minimize contention for this system resource. This allows the support of multiple high performance processors with near linear incremental performance as each processor is added. The bus is used only for issuing a command for an operation to occur and for returning a response to that command.

The CSS bus is a synchronous bus that is designed to clock at 20 MHz creating a fundamental time division of 50 nS. This allows 20 million bus operations per second and provides throughput of 80 Megabytes per second.

A later section of this document details operation of the CSS bus.

2.2 CSS Bus Arbiter

The CSS Bus Arbiter (CBA) is a module that provides the control for the CSS bus. It is responsible for the arbitration of the CSS bus as well as system clock generation. The CBA collects requests for use of the bus from all elements and grants use on each bus clock tick.

The CBA is implemented as a separate printed circuit card that attaches to the CSS motherboard.

A later section of this document details operation of the CSS bus.

2.3 CSS Backplane

The CSS backplane is the system bus motherboard that distributes bus signals between modules of the CSS. It also provides DC power distribution to the cards.

The CSS backplane accommodates 14 CSS modules; either processors, memories, or I/O modules.

2.4 Processing Module

The Processing Module (PM) is the main computational element within the CSS. It is based on a 25 MHz Motorola 68020 32 bit microprocessor and 68881 floating point unit (FPU). The PM includes proprietary memory management unit (MMU) which is well suited for demand paged virtual memory implementation in a multiple processor environment. It also includes large on-board cache memory.

The functional details of the Processing Module appear in later sections of this document.

2.5 Memory Module

The Memory Module (MM) is the main system storage element of the CSS. It provides from 8 to 32 MB of storage by populating one, two, or four banks. Error detection and correction (EDAC) corrects any single bit error and detects all double and most multiple bit errors.

The MM supports single or burst mode 64 bit reads. Also supports 8, 16, 24, and 32 bit writes. The board provides high throughput by allowing two way interleaving between banks.

The MM is designed with 1 Mbit dynamic RAM but will also accommodate 4 Mbit devices as they become available. It uses VLSI error detection and correction devices for EDAC.

The functional details of the Processing Module appear in later sections of this document.

2.6 Service Processor Module

The Service Processor Module provides the central console interface. It contains the system console port, remote diagnostics port, time of day clock, internal and external power system interfaces, environmental status monitor, floppy disk controller interface, and front panel interface.

It also provides a system wide interrupt dispatcher that collects system interrupts and fairly

distributes them to Processing Modules within the CSS.

The Service Processor Module is further detailed in later sections of this document.

2.7 I/O Module

The I/O Module (IOM) is an adaptor between the CSS bus and an I/O bus. It provides DMA routing from the I/O Link to the CSS bus.

2.8 CSS Configurations

The CSS can be configured with mixes of the above boards to best meet particular requirements of a given system application. Below is a matrix showing minimum, optimal maximum, and physical maximum numbers of modules allowed in a configuration.

Minimum is the number of a given board that is required to run a minimal system configuration. Optimal maximum is what is considered to be a reasonable maximum configuration for commercial applications. The physical maximum is an absolute maximum number of modules that can physically be connected to the CSS.

	Min	OMax	PMax
Processor Module	1	8	11
Memory Module	1	4	11
Service Processing Module	1	1	1
I/O Module	1	2	4
I/O Adaptor	1	2	4

Other restrictions and guidelines apply to system configurations. For instance, for balanced processor and memory subsystem performance, one Memory Module is required for each two Processor Modules.

3. I/O Subsystem

An I/O Subsystem (IOSS) of the System 3000 consists of an I/O Link, an I/O adaptor, a I/O backplane, and a variety of I/O processors.

The I/O Link is a generic bus interface between the CSS bus and specific system I/O busses. The I/O adaptor is a card that adapts the I/O Link to the particular characteristics of the I/O bus. The I/O backplane carries bus signals between the I/O processors and the I/O bus adaptor. The I/O processors are intelligent device and peripheral controllers.

3.1 I/O Link

The I/O Link (IOL) is 32 bit time division multiplexed bus that provides an interconnect between I/O modules and I/O adaptors. It is intended to be generic and allow connectivity to different varieties of I/O adaptors.

The IOL is a synchronous bus which is designed to clock every 100 nS. By multiplexing addresses and data on the bus, a peak transfer rate of 20 MB per second is achieved.

The operation of the I/O link is further detailed in a later section.

3.2 A1000 I/O Adaptor

The A1000 I/O Adaptor (IOA) provides an adaptation between the generic I/O Link and the A1000 I/O buses (the ICB and the DTB). This is the only IOA that will initially be supported in the System 3000.

3.3 A1000 I/O Backplane

The A1000 I/O Backplane carries ICB and DTB signals between the A1000 I/O Adaptor and the A1000 I/O controllers. It also provides DC power distribution to those controllers.

The backplane accommodates 20 cards which allows a single I/O adaptor and up to 19 I/O controllers.

3.4 Industry Standard Bus Interface

It is intended that an industry standard bus interface (ie - VME32) be supported within the System 3000. This is possible by developing an additional I/O Adaptor that adapts the I/O Link with the specific industry standard bus.

Additional industry standard bus interfaces are not currently planned in the initial System

3000, however accommodations have been made to allow this in the future.

3.5 I/O Control Processors

The System 3000 I/O subsystem is based on controllers and peripherals from the A1000 family. These I/O controllers include disk, tape, serial communication, parallel communications controllers. Also included is a Multibus adaptor adapting A1000 busses to the IEEE-796 bus. The I/O control processors (IOCP) are intelligent devices allowing a high degree of distributed I/O processing.

An additional IOCP is to be developed for the A1000 and available on the System 3000. This is a high performance communications controller referred to as the GC16/IOCP.

It is anticipated that the HSDT/IOCP and GC/IOCP will be replaced by the EDT/IOCP and EGC/IOCP respectively. The HSDT/IOCP and GC/IOCP will not be supported in the System 3000.

3.5.1 Enhanced Disk Tape IOCP

The Enhanced Disk Tape IOCP (EDT) is a high performance Disk and Tape controller. It provides a 2.4 MB/S ESMD interface and supports tape interface to either QIC or Pertec devices.

The EDT itself is a 68000 based controller with 256 KB of local memory. It has a SMD disk data channel and a parallel tape interface that can be adapted to either cartridge tape drives or nine track drives. The EDT communicates with the system through the A1000 ICB and A1000 DTB.

The EDT interfaces to one of two interface cards. The Dual Port interface (DP/IF) allows connection to 4 ESMD devices and a single QIC interface. This allows support of high performance disks and streaming cartridge tape devices. The 9 Track interface (9T/IF) connects to 4 ESMD devices and a single Pertec interface. This allows support of high performance disks while supporting half inch (9 track) tape devices.

3.5.2 Enhanced General Communications IOCP

The Enhanced General Communications IOCP (EGC) is a medium performance serial and parallel communications controller. It provides interface to 8 serial channels, 2 of which can be synchronous, and a single parallel port.

The EGC is based on a 10 MHz 68000 processor and provides up to 1 MB of local RAM. The EGC attaches to the I/O system through the A1000 ICB only.

The EGC interfaces to the General Communications interface card (GC/IF). This card allows interfacing to 8 RS-232 ports (two of which have full complement of signals to support synchronous operation) and a Centronics compatible parallel printer port.

3.5.3 Multibus Adaptor

The Multibus Adaptor Card (MAC) provides an interface from the A1000 I/O system to IEEE-796 bus (Multibus). The MAC is a "dumb" interface device providing only a shared buffer memory area between the central processors and the Multibus card.

The MAC interfaces to the A1000 I/O system through the ICB only.

3.5.4 Sixteen Channel General Communications IOCP

The Sixteen Channel General Communications IOCP (GC16) is a high performance communications controller supporting serial and parallel interfaces. It supports up to 16 serial channels (eight of which can be synchronous) and a high performance parallel port.

Two interface cards are provided with the GC16/IOCP. One interface card provides an interface to 16 RS-232 ports and a single Centronics port. The second interface provides 4 RS-422 ports.

The GC16/IOCP is a 68020 based controller. It attached to the I/O system through the ICB only.

3.5.5 ESDI/SCSI IOCP

The ESDI/SCSI IOCP is a high performance peripheral controller supporting both the Enhanced Small Device Interface (ESDI) and the Small Computer Systems Interface (SCSI).

The ESDI/SCSI IOCP is a 68000 based controller with up to 1 MB of local memory. It will support the serial ESDI disk data channel at greater than 1.25 MB/S and a SCSI channel that can be adapted to either synchronous or asynchronous and differential or single-ended operation. The ESDI/SCSI IOCP communicates with the system through the A1000 ICB and A1000 DTB.

The ESDI/SCSI IOCP will interface to a selection of interface cards which will tailor the number of ESDI drive connections and SCSI options. The SCSI interface will be the one of choice for supporting tape drives, optical drives, and large arrays of 5 1/4 inch disks.

3.5.6 Remote Communications Processor

A Remote Communications Processor (RCP) is a communications device that will be used on the A1000 and will be used on the System 3000 as well. The RCP acts a remote intelligent front end to the GC16/IOCP. It provides increased serial ports connectivity to the system. It also provides additional system topology flexibility by allowing clusters of ports to be remoted via high speed short haul composite links or via medium speed long haul links.

The RCP is a standalone unit that multiplexes and demultiplexes from a composite link to upto 32 separate asynchronous ports. The asynchronous ports are RS-232 compatible. The composite link is either RS-232 for low speed operation or RS-422 for high speed operation.

4. Peripherals

Various on-line and archival storage devices are supported by the System 3000. High capacity, high performance ESMD disk drives provide on-line main system storage. Cartridge and half inch tape drives provide low to high performance and capacity archival storage. Optical disks provide very high capacity archival capability.

In general, the same peripherals offers in the A1000 system will be applicable to the System 3000. Peripherals devices that will be obsoleted in the A1000 within the next year will not be supported on the System 3000.

4.1 Magnetic Disk Storage

Hard magnetic disk storage is provided for on-line main system storage and a low cost, low performance floppy disk drive is provided for IPL and software update purposes.

4.1.1 Hard Disk Storage

The System 3000 will initially support ESMD interface magnetic disk devices. These include both 8 inch and 5 1/4 inch form factor drives. When the EDSI/SCSI IOCP becomes available a much larger number of 5 1/4 inch drives will be usable. The drives will be mounted in a self powered rack mountable module. For instance, one rack would contain up to 2 each 8 inch drives or up to 4 each 5 1/4 inch drives as well as fans, cabling, and a power supply.

4.1.2 Floppy Disk Storage

A low performance 5 1/4 inch floppy disk drive is provided for IPL and software updates. The drive is located in the processor cabinet not the peripheral cabinet.

4.2 Tape Storage

Magnetic tape storage is provided for archival purposes. An inexpensive, low performance cartridge drive and higher performance half inch drives are provided.

4.2.1 Cartridge Tape

Quarter inch cartridge tape is provided for low priced, low performance backup. Streaming tape drives using the QIC-02 interface support 45/60 MB or 125 MB of storage per cartridge dependent on tape drive model and tape length. Internally the drives utilize the QIC-36 device level interface. Data is written to tape in the QIC-24 format on the 45/60 MB model and in the QIC-120 format on the 125 MB model.

The cartridge tape drive is located in the processor cabinet, not in the peripheral cabinet.

4.2.2 Half Inch Tape

Half inch 9 track tape is provided for standard media interchange and high performance archiving. Two half inch drives are supported. The first is a medium performance front loading device suited to standard media interchange and moderately demanding back-up performance. The second is a high performance reel to reel device for demanding back-up performance.

4.3 Optical Disk Storage

Optical disk storage is provided for very high capacity archival and on-line storage. Each optical disk device stores 1 Gigabyte on each side of a removable cartridge. At this time the optical drives interace only through a MAC with an SCSI interface board installed. The ESDI/SCSI IOCP will replace the MAC / SCSI host adapter interface combination.

5. System Package

Consistent with the concept of system scalability and configurability, the system packaging is modular and easily configured. All major subsystems and peripherals are installed as modules. Although internal subsystems and peripherals are configured modularly, the entire system package retains the image of a single integrated system.

The modular approach is intended to provide greatest system configurability, ease of manufacture, and ease of service.

5.1 Cabinet

The cabinet is based on standard 19 inch NEMA rack width. It serves as the external shell for the internal modules.

The initial system configuration will be a double wide rack width enclosure. One bay will enclose CSS logic, IOSS logic, power supply, and system cooling. The second bay holds system peripherals, their power supplies, and cooling. The two cabinets will be shipped separately and fastened together and internally cabled at the installation site.

An alternate configuration is a single width rack that is an integrated system package of logic and peripherals (similar to the Arete 1600 approach). This configuration is anticipated, although currently not scheduled for development.

The following are the major system sub-assemblies within the system cabinet.

5.2 CSS Card Rack

The CSS Card Rack is a rack mountable unit that mounts within the system cabinet. It provide mechanical support for the CSS backplane, CSS Arbiter, and up to 14 CSS modules. Also installed in the CSS Card Rack are the floppy disk drive and the cartridge tape drive.

5.3 A1000 IOSS Card Rack

The A1000 IOSS Card Rack is a rack mountable unit that mounts within the system cabinet. It provide mechanical support for the A1000 I/O adaptor, the A1000 I/O controllers, and interface cards.

5.4 Power Supply Module

The Power Supply module contains system power supplies and primary AC components.

The Power Supply module also mounts in the system rack.

5.5 Peripherals

I/O peripherals are rack mounted in the peripheral cabinet. Peripherals are powered separately from the system logic. Both 8 and 5 1/4 inch drives will be supported. A tray, which mounts in the cabinet, may be configured either of two ways: first, 1 - 2 each 8 inch drives, cabling and a power supply or, second, 1 - 4 each 5 1/4 inch drives, cabling, and a power supply.

6. System Control

System control consists of an operators panel and a system console terminal.

6.1 Front Panel

The front panel is the operator access to power the system up and down. It also is used to manually reset the system. A light is present that indicates the system is up and running. A full system status is read from the system console.

6.1.1 Front Panel Control Switches

The front panel consists of a keyswitch and an indicator light. The keyswitch performs power on, power off, and system manual reset. This protects the system from unauthorized or inadvertent access as a key is required. The keyswitch positions and read and interpreted by the SPM which has direct control of the system power.

6.1.2 Front Panel Status Indicator

The only visible indicator provided shows that the system is up and operating normally. This indicator is controlled by the service processor. Any abnormal conditions are indicated on the system console.

6.2 Control Parameters

Various system control parameters such as whether the system "auto boot" (automatic boot after returning from system power failure), timing parameters related to "auto boot", or commands to margin system DC supply levels are controlled by the SPM. Any programmable control parameter is soft configured on the service processor avoiding the need for configuration switches. The SPM has 50 bytes of battery backed up CMOS RAM which may contain configuration parameters and system control passwords.

6.3 Console Terminal

A serial port is provided to act as the main system console and local diagnostics port. All system diagnostics can be performed from this port. System power ON/OFF and system RESET functions can be performed from the console by appropriate commands and passwords. This port also acts as the system console for the operating system.

6.4 Hardcopy Terminal

A serial port is provided for attachment to a hardcopy output device.

6.5 Remote Console Port

A serial port intended for connection to a modem for remote system access and diagnostics is provided. All functions performed at the main system console are supported at this port.

6.6 Auxiliary Port

An auxiliary serial port will be provided. This port can be used to communicate with external devices such as a UPS.

6.7 Environmental Monitors

The SPM provides facilities for monitoring several environmental parameters. These include: air temperatures in several locations, power supply output voltages, power supply status, UPS status, AC line voltage. Abnormal conditions will be reported to the system console.

7. System Hardware Diagnostics

System diagnostics consist of four levels of test, each level built on the previous. The first level is the system power-up test; the next is the board-level diagnostic; the third is the system confidence test and the last is the system exerciser. The diagnostic strategy implements the concept of incremental testing from a minimum hardware kernel. The various test levels depend on the prior level for assurance of a minimal operational capability.

Source of the diagnostic test levels are firmware, floppy disk, hard disk and remote access (modem or direct connect) and download. Firmware controls the initial power-up testing. Continued testing is possible through any of the remaining media including remote link. Power-up testing of devices not covered by the power-up firmware will be executed from hard disk with floppy and remote available as backup.

7.1 Power-up Tests

The power-up tests are a series of firmware-based tests run on each of the intelligent in the system. The tests on each board begin by checking the microprocessor chip, the checksum of the firmware, addressability, data line I/O, local memory and bus interface hardware. Actual access to the bus is denied until the power-up controller (in this case, the Service Module) enables each.

The Service module will test itself, then proceed to test the bus, memory, the I/O subsystem and then each of the processor modules. Each processor module will be woke-up, queried for status and allowed to complete its own power-up sequence.

When the board power-up sequence is complete, a short exerciser will be run. That test will begin tasks on a single processor, then incrementally added processors and tasks until the full system is active. Power-up will be complete after successful execution and system boot will begin.

Should a new board type be added to the system that cannot be handled by the Service Module firmware, additional code will reside on both hard disk and floppy to support that phase of power-up. This code can be added to the system by floppy or by remote transmission.

7.2 Board Diagnostics

Each board has diagnostic support required to assist in development and manufacturing of individual circuit boards. These diagnostics exercise all major functions within each of the PCBs.

The board diagnostics exist in both floppy and remote download form. They are intended

to serve three separate levels of user and so are provided in those forms. The first level is the end user / customer confidence mode. This is a version that has all sequences predefined and tests each board in the system in an end to end order. It requires minimal user expertise and interaction.

The next level is field service mode. These tests provide the field engineer with the capability of defining test sequences and formats and isolates failures to the Field Replaceable Unit (FRU). This level resides on the customer floppy in transparent mode or can be executed through the remote diagnostic system.

The most detailed level is the Manufacturing mode. This level includes all of the loops, data and address manipulation and detail reporting to support debug of each individual board. Diagnostics can be run from a test bed or within a system and can be executed from floppy or by direct link. Fault isolation is to the subfunction level with loop and trace (verbose) mode available to support isolation to the component level.

7.3 System Level Confidence Test

The System Level Confidence Test is an extension of the board level diagnostic. At this level, the boards are judged working. Each function within the system is then executed on a one by one basic to prove that the system functions as a unit. Each known combination is run and then compared to expected results. This is a single process test procedure. Any fault at this level should be isolatable to a single FRU 80% of the time and within two FRUs 90% of the time.

7.4 System Exerciser

The System Exerciser is a worst case test mode. It consists of a series of multitasking UNIX like kernels on each processor module and on the service module. Tasks are initiated and monitored by the Service Module. In the canned mode, tasks are first run only on one processor, then two and so on until the entire system is running full out doing diverse jobs. These include HD reads (& writes when authorized), port I/O, memory tests and contention, tape writes and reads as well as processor functions. If the system can pass this test, it will run UNIX.

In a manual mode, separate functions or devices can be selected in order to stress a particular portion. Multiple tasks can be queued to again present worst case. This series of tests can be run floppy based or remotely through direct link or modem.

The intended usage is manufacturing burn-in in automatic mode as well as on site wring out of a new or suspect system. The exerciser would be available to the customer in a nondestructive mode only.

7.5 Remote Diagnostics

All system hardware diagnostics can be run via the remote console port. This allows service personnel to remotely isolate failed system components. This can be via direct RS-232 link or by modem from an off site location. Downloads are guaranteed accurate by use of CRC algorithms. The system can also act as a remote monitor for floppy based execution.

8. System Power

8.1 Logic Power Supply

The main system logic power supply provides DC power for the CSS card cage and the IOSS card cage within the first system rack. The logic power is provided by 1-3 each +5 VDC at 200 A supplies which are modular in construction, with plug in replacement as a design goal, and used in a current sharing mode. Only one supply would be required to operate a minimally configured system, two supplies for most systems. With three supplies installed N+1 redundancy is achieved, meaning there is one more supply than is actually required to operate the system. Should one supply fail the others can maintain the system until a replacement is available. The minimum number of supplies required in a system will be determined at configuration time. The redundancy feature could be available as a customer option. Hot (power on) replacement of supplies is not supported.

The system DC power requirement based on estimates for individual PCBs is as follows:

Minimum System	585 watts
Maximum System	2690 watts

Peripherals in the second cabinet are separately powered.

8.2 Communications Power Supply

A separate +/- 12 VDC supply is provided for operation of the communication ports on the IOCPs. This supply will also be used to power the cooling fans. Redundancy is provided by the SPM power supply.

8.3 SPM Power Supply

The SPM is powered by a separate supply. This supply is on if the system is connected to a live circuit and the main circuit breaker is ON. This supply also provides redundancy for the communications power supply.

8.4 Power Supply Control

The system power supply is controlled by the service processor. The service processor can turn system power on and off, margin DC voltages, and sense DC levels, and sense line voltage levels.

8.5 Uninterruptible Power Supply

The system supports connection to an uninterruptible AC power system.

9. System IARM

The System 3000 is intended to meet certain installability, availability, reliability, and maintainability (IARM) objectives.

9.1 Installability

The System 3000 will be installed, configured, and made operational in one day or less.

It is anticipated that communications circuits, local terminal and printer wiring, and AC power wiring be prepared at the site before installation time.

9.2 Availability

Several features of the System 3000 address system availability. The system minimizes common points of failure, provides protected main memory storage, provides error checking on system bus transmissions, and provides for uninterruptible power systems.

The system design minimizes common points of failure which would be fatal to the system. The common points of failure in the system are such that their failures do not dominate the overall failure rates of the system.

Main system memory storage is protected by error detection and correction which corrects any single bit error, detects any double bit error, and detects some multiple bit errors.

All transmissions on System 3000 busses are parity checked for data integrity. System bus protocols allow modules to retry bus cycles to attempt recovering from a failed cycle.

9.3 Reliability

The reliability of the System 3000 is high for a machine in its class. The range of system reliability is as great as the range of configurations. System MTBF is expected to range between 500 and 4000 hours depending on actual configuration.

Also note that many system hardware failures will not be fatal to the entire system. Many system failures are detected and resolved by deconfiguring a failed component before they become catastrophic to the system.

The level of system wide monitoring supported through the service processor will enable detailed evaluation of the system environment and health. An increase in average system temperature over time can be used to signal air filter replacement. A change in the inlet to outlet air temperature delta will indicate a system problem such as a fan failure. Drift in the DC power voltage levels is monitored and corrective action scheduled to prevent

failure. Inlet air temperature is monitored for compliance with the environmental specifications. Input AC voltage is monitored, and extended operation at brown out voltages prevented.

9.4 Maintainability

The System 3000 addresses maintainability in several areas. Local and remote diagnostic access and high degree of serviceability.

System diagnostics can be run to isolate 90% of all hard failures to a field replaceable unit (FRU). These system diagnostics can be run either locally or remotely by trained service personnel. System level diagnostics must be run off line.

Also, any module that detects itself as being bad while the system is on-line will deconfigure itself and flag the failure to the system operator.

Any failed FRU can be replaced by trained service personnel in fifteen minutes or less.

All major system components with the exception of system backplanes and system chassis are considered to be field replaceable. Included as FRUs are all major logic assemblies, peripherals, and power supplies.

Hot (powered up) service is not supported.

10. Environmental and Safety

The System 3000 is compliant with various environmental, safety, and regulatory agency requirements.

10.1 Environmental

The System 3000 is intended to operate in copy room type environment. It does not require special cooling, raised flooring, etc.

The following environmental specifications are met by the system.

- Temperature (operating)

  10 to 35 degrees centigrade at sea level, with maximum change of 10 degrees centigrade per hour. Maximum operating environment temperature is dependent on altitude, and a graph will be provided to indicate maximum operating temperature from 0 to 10,000 feet.

- Temperature (non-operating)

  minus 20 to 60 degrees centigrade with maximum change of 20 degrees centigrade per hour.

- Humidity (operating)

  20 to 80 percent relative humidity (non-condensing) with maximum change of 10 percent per hour.

- Humidity (non-operating)

  5 to 95 percent relative humidity (non-condensing).

- Altitude (operating)

  5,000 feet (3000 meters).

- Altitude (non-operating)

  40,000 feet (12,000 meters).

- Acoustic Emissions

  50-55 dBA.

- Power Requirements

  220 VAC 50/60 Hz. Estimated up to 18 Amps for logic cabinet. Estimated up to 20

Amps for peripheral cabinet.

10.2 Safety and Regulatory

The System 3000 is compliant with the following safety and regulatory agencies: Underwriters Laboratories, Federal Communications Commission, Canadian Standards Association, and international safety and regulatory (ie - German TUV and IEC).

Specifically the system will comply with the following domestic and international safety and emissions standards.

- UL 478

  Underwriter Laboratories safety certification for "Electronic Data Processing Units and Systems". The system will in general be designed to meet the 5th edition, with the exception of section 9A.

- FCC Class A

  Federal Communications Commissions standards for acceptable radio frequency emissions.

- CSA C22.2-220

  Canadian Standards Associations safety standards for "Data Processing Equipment".

- VDE 871 Level B

  German regulatory approval of radio interference suppression of radio frequency equipment for industrial, scientific, medical, and similar purposes.

- IEC 950 (TUV)

  International Electrotechnical Commission standards on "Safety of Data Processing Equipment".

11. System Cost

11.1 Cost Breakdown

Estimated costs are presented here for each major system element. Any special consideration is noted. Also estimates for introduction time (I=1Q88) and introduction plus 12 months (I+12=1Q89) are included.

Cost of system logic is estimated based strictly on anticipated complexity factor and chip count. Only parts cost is included here.

Estimates at introduction (I) are plus or minus 25 percent. Estimates at introduction plus 12 months (I+12) are plus or minus 50 percent. The cost estimates are expected to firm as actual logic implementation is determined and as actual component pricing is forecasted.

- Compute Module

  Cost at I=($1800). Cost at I+12=($1000). Cost estimate is based on 325 ICs. Cost is greatly affected by 68020 and 68881 pricing.

- Memory Module

  Cost estimate based on 200 ICs and $120 per Megabyte for RAM devices at "I" and $80 per Megabyte for RAM at "I+12". Total cost of this system element is dominated by DRAM pricing.

- Service Processor Module

  Cost at I=($800). Cost at I+12=($600). Cost estimate is based on 275 ICs.

- I/O Module

  Cost at I=($900). Cost at I+12=($750). Cost estimated on 350 ICs.

- CSS Arbiter

  Cost at I=($300). Cost at I+12=($200). Cost estimated on 100 IC complexity.

- I/O Adaptor

  Cost at I=($900). Cost at I+12=($750). Cost estimated on 3505 ICs.

- I/O Backplane Cost at I=($320). Cost at I+12=($250). Cost based on 20 slots.

- CSS Backplane

  Cost at I=($320). Cost at I+12=($250). Cost based on 14 slots.

- EDT

  Cost at I=($900). Cost at I+12=($700).

- GC16/IOCP

  Cost at I=($750). Cost at I+12=($600).

- RCP

  Cost at I=($850). Cost at I+12=($700).

- Package

  Package cost includes system chassis, card cages, and external panels. This cost is estimated as I=($1800) and I+12=($1800).

- Power Supply

  Includes power supply, primary AC components, and internal system power distribution. Cost is estimated on $1.00 per watt for the above. Cost at I=($2700). Cost I+12=($2700).

- Peripherals

  Magnetic disk includes the peripheral itself, power supply, and mounting hardware is estimated at $9.00 per Megabyte at "I" and $8.00 per Megabyte at "I+12".

  Streaming tape unit cost is I=($600) and I+12=($500).

  Floppy disk unit cost is I=($100) and I+12=($75).

11.2 Cost for Specific Configurations

Consider costs of three configurations.

11.2.1 Configuration A

Dual wide system cabinet, power supply, streaming tape unit, floppy disk unit, dual card cages, 1 Processing Module, Service Processor, 8 Mb Memory Module, I/O Module, A1000 I/O Adaptor, 1 EDT, 2 GC16/IOCP, 2 RCP, 0.6 Gigabyte disk storage.

Cost at I=($xxxxx). Cost at I+12=($xxxxx).

11.2.2 Configuration B

Dual wide system cabinet, power supply, streaming tape unit, floppy disk unit, dual card cages, 4 Processing Module, Service Processor, 32 Mb Memory Module, I/O Module, A1000 I/O Adaptor, 2 EDT, 4 GC16/IOCP, 6 RCP, 1.2 Gigabyte disk storage.

Cost at I=($xxxxx). Cost at I+12=($xxxxx).

11.2.3 Configuration C

Dual wide system cabinet, power supply, streaming tape unit, floppy disk unit, dual card cages, 6 Processing Module, Service Processor, 128 Mb Memory Module, I/O Module, A1000 I/O Adaptor, 4 EDT, 8 GC16/IOCP, 12 RCP, 4.8 Gigabyte disk storage.

Cost at I=($xxxxx). Cost at I+12=($xxxxx).

12. CSS Bus

12.1 CSS Bus Overview

The system bus is an N-port time-division multiplexed transmission switch. Any of the N ports (backplane positions/slots) can send (receive) transmissions to (from) any of the N ports, including itself. Each transmission consists of a source port address, a destination port address, a transmission type and 8 bytes of "data". The transmission type determines what the "data" field contains. Error detection is provided for the source slot address, destination slot address and transmission type fields and optionally provided for the "data" field. The design value of N is 14 (ports); the design clock rate is 20 MHz.

Modules installed in bus ports interact with each other by exchanging transmissions over the bus. There are two types of transmissions, COMMANDS and RESPONSES. A module on the bus begins an interaction with another module by sending a COMMAND. The source of the COMMAND is the MASTER for that interaction; the destination of the COMMAND is the SLAVE. The SLAVE sends a RESPONSE back to the MASTER if required to complete the interaction.

12.2 CSS Bus Operation

The CSS bus arbiter controls access to the bus. To transmit on the bus, a module issues a request, a request modifier and a destination port address to the arbiter. If the request is to send a COMMAND, the arbiter checks that the destination port has a COMMAND input buffer available. A port with a COMMAND input buffer available is said to be READY. If the request is to send a RESPONSE, the destination port is required to have to have enough RESPONSE buffer space available for the size of the RESPONSE it requested. Ports wanting to send COMMANDS to destinations that are READY and ports wanting to send RESPONSES arbitrate for time slots on the bus. Arbitration occurs for each time slot.

Arbitration priority is determined by the port number of requesting port. Port N has the highest priority, and port 0 the lowest. Computational modules as a group are assigned the lowest priorities. A bus bandwidth spreading scheme insures that all computational modules get about the same amount of access to the bus.

A module on the bus is READY to receive a COMMAND when it has at least one COMMAND buffer free. Each module indicates to the arbiter how many COMMAND buffers it has free. The arbiter maintains a count of free COMMAND buffers for each module and decrements the count for the destination module as permission to send each COMMAND is GRANTED. Each module in turn signals the arbiter to increment its free buffer (READY) count whenever one of the module's COMMAND buffers becomes free.

The bus arbiter also supports interlocked sequences of operations. These sequences are required to to support the TAS, CAS and CAS2 instructions of the Motorola 68020 and are a generalization of the READ/MODIFY/WRITE operation. Interlocked sequences are atomic to each other and are composed of any number of READ and WRITE commands. The signal LOCK is asserted by the arbiter when an interlocked sequence is in progress. A interlocked sequence begins when LOCK is not asserted and a port asserting BUS REQUEST and MODIFY to the arbiter is granted the bus. The sequence ends when the port executing the sequence deasserts MODIFY to the arbiter. To minimize performance loss, an interlocked sequence does not lock out commands from ports not asserting MODIFY to the arbiter.

The arbiter grants access to the bus by asserting GRANT to each module that wins an arbitration. GRANT is asserted to a module for one time slot (clock cycle). A module receiving a GRANT may transmit on the bus during the time slot immediately following the time slot in which that GRANT was received.

When a module is GRANTED the bus to send a RESPONSE, the module may use the bus for up to four consecutive time slots by asserting BURST to the arbiter during all but the last time slot. The assertion of BURST during a time slot prevents the arbiter from issuing a GRANT to any module for use of the bus during the next time slot. Only RESPONSES can be sent in BURST mode. BURST mode RESPONSES can be to one or more destination ports.

When a module transmits on the bus, it asserts the signal BUS_ACTIVE to indicate the presence of a transmission.

Each module monitors all bus transmissions. Each transmission is first checked for destination field errors. A module recognizes a transmission as addressed to it if the destination field contains no detected errors and matches the module's port (slot) number. The transmission is then checked for source and type field errors and optionally, parity errors in the data field. If the transmission is a RESPONSE, it must be expected and the source field must be that of the expected source.

Each transmission received without detected error is indicated by asserting ACK on the bus during the second time slot after the transmission. Transmissions received with one or more detected errors are indicated by asserting NACK on the bus during the second time slot. The sole exception to this are transmissions of type CONTROL WRITE. These transmissions are ACKed or NACKed based on the detection of errors in the source and type fields. Any detected errors in the data filed are ignored.

Only the destination port asserts ACK or NACK for a transmission. Transmissions with destination field errors or destination fields not matching the port (slot) number of any installed module are neither ACKed nor NACKed.

When a system bus transmission fails, retry is permitted, but not required. If retry is attempted, the module that issued the COMMAND resulting in the failed transmission

restarts the transaction by reissuing the COMMAND.

12.3 Transmission Format

Each transmission has a destination field, a source field, a type field and a data field. The destination field contains the destination port number and, for error detection, the compliment of the port number. The source field contains the source port number and its compliment. The type field indicates the type of COMMAND or RESPONSE and the format of the data field. The type field has a parity bit for single-bit error detection. A parity bit is also defined for each byte of the data field for single-bit error detection, but implementation is optional. The source of a bus transmission indicates whether data parity bits have been sent.

The data field is organized as 8 bytes of 8 bits each. The bytes are numbered 0 through 7 with byte 0 the most significant and of lowest address. Bits within a byte are numbered 7 through 0 with bit 7 the most significant. Each bit in the data field has a name of the form BUS_DATA[B,b] where B is the byte number and b is the bit number within the byte. The optional parity bit for byte B is BUS_PARITY[B]. Data field parity is even. The bit BUS_PARITY_ENBL indicates whether data field parity is implemented.

System addressing is by 4 bits of physical port number and 32 bits of offset. The address of an operand is the address of its first (most significant and lowest address) byte.

The bus COMMANDS are as follows.

1. READ [size = 1, 2, 3, 4, 8, 16 or 32 bytes]

   The operand of a READ command for 4 bytes or less must not cross a long word (4 byte) boundary. The operand of a READ command for 8, 16 or 32 bytes must be aligned on an 8, 16 or 32 byte boundary, respectively, and must not cross a 4 kilobyte page boundary. READs of 8, 16 and 32 bytes are not supported by all module types. The offset address of the operand is sent in bytes 4 through 7 of the data field. Bytes 0 through 3 of the data field are undefined.

   The operand is returned in the data field of one or more RESPONSE transmissions. The operand is aligned in the data field for an 8 byte wide port. Bytes in the data field that are not part of the requested operand are undefined.

   Operands longer than 8 bytes require multiple 8 byte RESPONSE transmissions. Bytes are returned in order of increasing byte address with the bytes of lowest address returned first. Multiple RESPONSE transmissions may be sent one at a time or in one or more bursts.

2. WRITE [size = 1, 2, 3 or 4 bytes]

The operand of a WRITE command must not cross a long word boundary. The offset address of the operand is sent in bytes 4 through 7 of the data field.

The WRITE data is sent in bytes 0 through 3 of the data field and must be aligned for a 4 byte wide port.

3. CONTROL WRITE [signal = 0, 1, 2 or 3] [value = 0 or 1]

   The CONTROL command allows one of several control signals in a module to be asserted or deasserted even in the presence of errors in the "data" field. The specified signal is set to the specified value. The data field is undefined.

The bus RESPONSES are as follows.

RESPONSE [response type = DATA, ERROR DATA 0, ERROR DATA 1 or ERROR]

1. DATA is the normal response to a READ command. A READ command for more than 8 bytes requires more than one RESPONSE transmission.

2. ERROR DATA is the response to a READ command encountering a detected but uncorrectable data error. It contains the requested data as read or after correction has been attempted.

3. ERROR DATA 0 indicates that the error was detected by device responding to the READ.
4. ERROR DATA 1 indicates that the error was a transmission error detected by the IO MODULE.

5. ERROR is the response to a READ command that is somehow recognized as having failed to read anything. The data field of the response is undefined.

The format of the transmission TYPE field is as follows.

BUSTYPE [5:0] = [3 bit type field],[3 bit modifier field]

BUSTYPE [5:3]	type
0	RESPONSE
1	Not Used (Reserved)
2	Not Used (Reserved)
3	READ
4	Not Used (Reserved)

5	WRITE
6	CONTROL WRITE
7	Not Used (Reserved)

BUSTYPE [2:0]	size	response type
0	4 bytes	ERROR
1	1 byte	--
2	2 bytes	ERROR DATA 0
3	3 bytes	ERROR DATA 1
4	8 bytes	DATA
5	16 bytes	--
6	32 bytes	--
7	--	--

BUSTYPE [2:1]	control signal
0	Not Used (Reserved)
1	module enable
2	module interface enable
3	Not Used (Reserved)

12.4 Transmission Conventions

With the exception of CONTROL WRITE, the target of a READ or WRITE command is specified by its system address. Within a module, the range of defined offset addresses varies from 256 bytes to 4 gigabytes.

The offset address range 0xFFFF FF00 through 0xFFFF FFFF of all modules contains control and status registers. Some modules also have higher level message buffers in this address range. Some elements of this address range are found in more than one module and have the same address in each module in which they appear. For instance, the ID of the module installed in a bus port can be read at offset 0xFFFF FFFF.

For ease of use and testing, any bit that can be written in a control register can be read at the same byte and bit address and with the same sense. By definition, status registers are read-only.

The number of offset address bits decoded depends on the module receiving a command. Memory modules and i/o modules which support 4 GB offset address spaces decode all offset address bits. Computational modules and Service modules which support 256 byte address spaces decode only the low order 8 offset address bits.

Use of the 0xFFFF FF00 through 0xFFFF FFFF is as follows.

0xFFFF FF00-1F	Higher level message buffers (Computational, Service & Test)
0xFFFF FF00-03	Command buffer
0xFFFF FF20-2F	Reserved
0xFFFF FF30-3F	Reserved
0xFFFF FF40-4F	Reserved
0xFFFF FF50-5F	Reserved
0xFFFF FF60-63	Interrupt request (Service & Test)
0xFFFF FF80-BF	Interrupt acknowledge (Service & Test)
0xFFFF FFC4-C7	Interrupt vector (All)
0xFFFF FFC0-FF	Control and Status registers (All)
0xFFFF FFE4-E7	Memory check bits and control (memory)
0xFFFF FFEC-EF	Memory error information (memory)
0xFFFF FFF4-F7	Memory error address (memory)
0xFFFF FFFC-FE	Memory status register (memory)
0xFFFF FFFC	68020 Interrupt request level (Computational)
0xFFFF FFFF	Module ID (All)

12.5 Interface Specification

12.5.1 Bus Signals

The bus signals are as follows.

1. BUS_DEST[3:0]
2. BUS_DEST[3:0]*
3. BUS_SRC[3:0]
4. BUS_SRC[3:0]*
5. BUS_TYPE[5:0]
6. BUS_TYPE_PARITY
7. BUS_DATA[0:77]
8. BUS_PARITY[0:7]
9. BUS_PARITY_ENBL
10. BUS_ACTIVE
11. BUS_ACK
12. BUS_NACK

The signals from the arbiter or backplane to each port are as follows.

1. ARB_CLOCK*

   Bus Clock.

2. ARB_DCLOCK*

   A delayed version of CLOCK*.

3. ARB_GRANT*

   When asserted, the port may transmit on the bus during the next time slot and must deassert ARB_REQUEST*.

4. ARB_LOCK*

   ARB_LOCK is asserted when an interlocked sequence of operations is in progress. The signal is provided for test purposes and is not required for normal operation of the bus.

5. ARB_GRANTERR*

   When asserted, ARB_GRANTERR* indicates that two or more GRANT's were issued for one bus cycle. ARB_GRANTERR* is asserted during the third cycle after the cycle in which the multiple GRANT fault occurred. The signal is for fault isolation only.

6. ARB_SLOT[3:0]

   Indicates the geographical bus slot number.

The signals from each port to the arbiter are:

1. ARB_RS_READY*

   When asserted, the READY counter is reset to ZERO (NOT READY).

2. ARB_INC_READY*

   When asserted, the READY counter is incremented by one on each falling edge of ARB_CLOCK*. The maximum READY count is 7. Excess assertion of

ARB_INC_READY will not cause the count to exceed 7 or roll-over to 0.

3. ARB_CPU*

   When asserted, indicates that the port is occupied by a computational module.

4. ARB_DEST[3:0]

   Slot address of destination.

5. ARB_MODIFY*

   When asserted in conjunction with ARB_REQUEST*, indicates that the port wants to transmit a READ or WRITE command that is part of an interlocked sequence of operations.

   Once a port is granted the bus to begin an interlocked sequence, no other port can begin an interlocked sequence until the locking port deasserts ARB_MODIFY*.

6. ARB_RESP*

   When asserted in conjunction with ARB_REQUEST*, indicates that the port wants to transmit a RESPONSE.

7. ARB_REQUEST* When asserted, indicates that the port wants transmit a COMMAND or RESPONSE on the bus.

8. ARB_BURST*

   When asserted, prevents a GRANT from being issued for the next time slot.

12.5.2 Bus Timing

The timing of signals passing between a bus module and the arbiter is specified at the point where the backplane connector and pc board of the module or arbiter join. 2 ns is allowed for a signal to pass through the two backplane connectors and the backplane.

Signals from a bus module to the arbiter must be valid 18 ns after the falling edge of ARB_CLOCK*.

Signals from the arbiter to a bus module must be valid 4 ns before the falling edge of ARB_CLOCK*.

Signals driven by a module onto the bus must be enabled and valid or disabled by 27 ns

after the falling edge of ARB_CLOCK*.

12.5.3 Drivers

With the exception of BUS_ACTIVE, BUS_ACK and BUS_NACK, all bus signals are driven with 74F244's.

The signals BUS_ACTIVE, BUS_ACK and BUS_NACK are driven with 100 mA discrete open emitter drivers.

12.5.4 Receivers

All signals are received with 74F374's.

12.5.5 Bus Pinouts

12.5.5.1 Connector P1 Pinouts

Pin	Row A	Row B	Row C
1	COMMON	BUS_PARITY[0]	BUS_DATA[07]
2	COMMON	BUS_DATA[06]	BUS_DATA[05]
3	COMMON	BUS_DATA[04]	BUS_DATA[03]
4	COMMON	BUS_DATA[02]	BUS_DATA[01]
5	COMMON	BUS_DATA[00]	BUS_PARITY[1]
6	COMMON	BUS_DATA[17]	BUS_DATA[16]
7	COMMON	BUS_DATA[15]	BUS_DATA[14]
8	COMMON	BUS_DATA[13]	BUS_DATA[12]
9	COMMON	BUS_DATA[11]	BUS_DATA[10]
10	COMMON	BUS_PARITY[2]	BUS_DATA[27]
11	COMMON	BUS_DATA[26]	BUS_DATA[25]
12	COMMON	BUS_DATA[24]	BUS_DATA[23]
13	COMMON	BUS_DATA[22]	BUS_DATA[21]
14	COMMON	BUS_DATA[20]	BUS_PARITY[3]
15	COMMON	BUS_DATA[37]	BUS_DATA[36]
16	COMMON	BUS_DATA[35]	BUS_DATA[34]
17	COMMON	BUS_DATA[33]	BUS_DATA[32]
18	COMMON	BUS_DATA[31]	BUS_DATA[30]
19	COMMON	BUS_PARITY[4]	BUS_DATA[47]
20	COMMON	BUS_DATA[46]	BUS_DATA[45]
21	COMMON	BUS_DATA[44]	BUS_DATA[43]
22	COMMON	BUS_DATA[42]	BUS_DATA[41]
23	COMMON	BUS_DATA[40]	BUS_PARITY[5]
24	COMMON	BUS_DATA[57]	BUS_DATA[56]
25	COMMON	BUS_DATA[55]	BUS_DATA[54]
26	COMMON	BUS_DATA[53]	BUS_DATA[52]
27	COMMON	BUS_DATA[51]	BUS_DATA[50]
28	COMMON	BUS_PARITY[6]	BUS_DATA[67]
29	COMMON	BUS_DATA[66]	BUS_DATA[65]
30	COMMON	BUS_DATA[64]	BUS_DATA[63]
31	COMMON	BUS_DATA[62]	BUS_DATA[61]
32	COMMON	BUS_DATA[60]	BUS_PARITY[7]
33	COMMON	BUS_DATA[77]	BUS_DATA[76]
34	COMMON	BUS_DATA[75]	BUS_DATA[74]
35	COMMON	BUS_DATA[73]	BUS_DATA[72]
36	COMMON	BUS_DATA[71]	BUS_DATA[70]
37	COMMON	BUS_SRC[3]	BUS_SRC[3]*
38	COMMON	BUS_SRC[2]	BUS_SRC[2]*
39	COMMON	BUS_SRC[1]	BUS_SRC[1]*
40	COMMON	BUS_SRC[0]	BUS_SRC[0]*
41	COMMON	BUS_DEST[3]	BUS_DEST[3]*
42	COMMON	BUS_DEST[2]	BUS_DEST[2]*
43	COMMON	BUS_DEST[1]	BUS_DEST[1]*
44	COMMON	BUS_DEST[0]	BUS_DEST[0]*
45	COMMON	BUS_TYPE_PARITY	BUS_TYPE[5]
46	COMMON	BUS_TYPE[4]	BUS_TYPE[3]
47	COMMON	BUS_TYPE[2]	BUS_TYPE[1]
48	COMMON	BUS_TYPE[0]	BUS_PARITY_ENBL
49	COMMON	RESERVED	BUS_ACTIVE
50	COMMON	BUS_ACK	BUS_NACK

12.5.5.2 P1 Signals - Bus Termination

BUS_DATA[], BUS_PARITY[], BUS_PARITY_ENBL, BUS_SRC[], BUS_SRC[]*, BUS_DEST[], BUS_DEST[]*, BUS_TYPE[], BUS_TYPE_PARITY and RESERVED are terminated at each with 150 Ohms +/- 2%, 100 mW to +5V and 100 Ohms +/-2%, 100 mW to COMMON (equivalent to 60.0 Ohms in series with 2.00 V, nominal).

BUS_ACTIVE, BUS_ACK and BUS_NACK are terminated at each end with 820 Ohms +/- 5%, 100 mW to +5V and 43 Ohms +/- 5%, 100 mW to COMMON (equivalent to 40.9 Ohms in series with 0.25 V, nominal).

12.5.5.3 Connector P2 Pinouts

Pin	Row A	Row B	Row C
2		+5V	
4	+12V	N/C	+12V
5	ARB_CLOCK*	COMMON	ARB_DCLOCK*
6	ARB_CPU*	COMMON	BUS_SLOT[3]
7	ARB_DEST[3]	COMMON	BUS_SLOT[2]
8	ARB_DEST[2]	COMMON	BUS_SLOT[1]
9	ARB_DEST[1]	COMMON	BUS_SLOT[0]
10	ARB_DEST[0]	COMMON	BUS_C10
11	ARB_GRANT*	COMMON	BUS_C11
12	ARB_BURST*	COMMON	BUS_C12
13	ARB_REQUEST*	COMMON	BUS_C13
14	ARB_RESP*	COMMON	BUS_C14
15	ARB_MODIFY*	COMMON	BUS_C15
16	ARB_LOCK*	COMMON	COMMON
17	ARB_INCREADY*	COMMON	BUS_PWRFAILWRN*
18	ARB_RSREADY*	COMMON	COMMON
19	ARB_GRANTERR*	COMMON	BUS_RESET*
20	BUS_A20	COMMON	COMMON
21	BUS_A21	COMMON	BUS_C21
22	BUS_A22	COMMON	BUS_C22
23	BUS_A23	COMMON	BUS_C23
24	BUS_A24	COMMON	BUS_C24
25	+12VAUX	+12VAUX	+12VAUX
26	BUS_A26	COMMON	BUS_C26
27	-12VAUX	-12VAUX	-12VAUX
28	BUS_A28	COMMON	BUS_C28
29	-12V	N/C	-12V
31		+5VAUX	

12.5.5.4 P1 Signals - Bus Termination

BUS_RESET*, BUS_PWRFAILWARN*, BUS_AXX and BUS_CXX are terminated at each end with 150 Ohms +/- 5% to +5V and 220 Ohms +/- 5% to COMMON

(equivalent to 89.2 Ohms in series with 2.97 V, nominal).

BUS_SLOT[3:0] are not bussed. BUS_SLOT[3;0] for each slot are selectively connected to COMMON to encode the slot number in positive logic. The low value is provided by the connection to COMMON on the backplane; the high value is provided by a pull-up resistor per signal on each module.

13. Processing Module

13.1 Processor

13.1.1 Type

The PM board is built around a Motorola 68020. A 68881 floating point coprocessor (FPU) is included.

13.1.2 Performance

The processor's clock rate is 25MHz. The coprocessor's clock rate is variable, with options of 12 MHz and 20 MHz.

Cache and Memory Management Unit (MMU) are organized to allow no wait state memory cycles for reads which hit cache and all writes. (One wait state may be inserted in any memory cycle which immediately follows a write hit. This allows sufficient time to reliably check rights and write in the cache's data store.) At least six wait states are imposed on a cache miss.

13.1.3 Address Spaces

Three spaces are decoded from the virtual address: User memory, Supervisor memory, and I/O.

13.1.3.1 Privilege Level

The supervisor bit (in the 68020's processor status word) determines whether the CPU is at "user" or "kernel" privilege level. In the "user" state, user memory is the only legal decode, and any other decode will cause a bus error exception.

To clear the supervisor bit, the kernel creates the desired stack structure, and executes

```
MOVE An,USP        ; if we are switching to a new
                   ; user or USP was changed

RTE
```

These are, of course, privileged instructions. To set the supervisor bit, the user may:

1. Violate the access rights
2. Refer to a non-resident page (as shown in the descriptor)
3. Refer to a page whose page table or page pointer table is not resident and whose descriptor is not loaded in the TLB already
4. Do anything that causes an exception in the 68020.

Detectable hardware faults such as garbled bus transaction will also generate bus error exceptions. (See the MMU description for the details of the memory management and the fault register.)

13.1.3.2 User Memory

User memory is virtual, subject to management. A user virtual space is 2 gigabytes (2 GB). (The MMU allows many user spaces to reside in memory simultaneously.) The page size is 4 kB.

13.1.3.3 Supervisor Memory

Supervisor space includes the current user space, a 1 GB kernel space, and a 1 GB input/output space. When the supervisor accesses its image of user space, the user's root pointer and page tables are used, and the permissions and attributes in the user's page descriptors apply.

13.1.3.4 Off-board I/O

A lookup table is used to map the I/O space into CSS bus physical addresses. The I/O space is mapped in 1 MB sections to individual bus addresses.

13.1.3.5 Local I/O and Test

The local I/O includes test paths for all storage, some control and status functions, and the detachable diagnostic board. The decodes are fixed and restricted to the supervisor.

13.1.4 Instruction Set

The system determines the behavior of certain 68020 instructions.

13.1.4.1 Semaphore Instructions: TAS, CAS, and CAS2

These instructions generate a special read-modify-write (R-M-W) memory cycle. The PM

and the CSS arbiter arrange for these cycles to be atomic to one another; that is, the read part of one R-M-W cycle cannot be separated in time from its write part by either part of another R-M-W cycle. Thus these instructions may be used to arbitrate critical resources.

Warning: **a R-M-W cycle in progress (that is, read done and write not started yet) does NOT block another processor from writing on that same location. Such a write would be lost when the R-M-W completed. Warning #2: The R-M-W cycle generates much more system traffic than normal reads and writes, so use it sparingly.**

13.1.4.2 RESET

The RESET instruction generates a delay of more than 64 microseconds (us) and less than 150 us. During the delay the processor ignores interrupt requests. The state of the system is not affected. (Reset functions for various PM features are provided in the local I/O space.)

13.1.4.3 BKPT

The breakpoint instruction generates an illegal instruction exception, with the four word normal exception stack frame. The stack may be examined by the handler to determine the trapped PC, which may be used to read the three bit breakpoint select value.

13.1.4.4 F-line Instructions

Floating point instructions are executed jointly by the 68020 and FPU. If the FPU is missing from the board, the exception logic detects an invalid decode and the instruction generates the f-line exception (vector 11). An invalid instruction encoding generates the same exception.

13.1.4.5 Other Coprocessors

An instruction for any coprocessor except the FPU generates the f-line exception. In particular, the 68020 cannot communicate (as described in the MC68020 User's Manual "Coprocessor Interface" chapter 8) with coprocessors which may be mounted on another board. These user's coprocessors may, however, be mapped into virtual memory and the coprocessor interface emulated by the user's software. For efficiency, those users should use subroutine calls, not f-line traps, to access their special hardware.

13.1.5 Read Ahead

The 68020 cpu chip will generate spurious memory read cycles under two conditions:

1. Prefetching. The chip reads instruction words from memory before the PC value is actually calculated. It guesses the instructions in progress do not alter program flow, and it reads from memory at a location obtained by incrementing the current PC value.
2. MOVML instruction. In the execution of a MOVML if the destination is the cpu, one more operand will be read than the instruction called for. The value which was read is discarded. All of the reads generated by the instruction are from consecutive locations.

If the PC can reach the last word of its last code page, or the program will MOVML from its last long word in any of its data pages, then the program may generate these spurious reads at addresses which are outside its translation tables. (See the WARNING in the fault register description.)

13.1.6 EPROM

A 64 kB EPROM is always available to the supervisor in the local I/O space. Following a reset to the processor, repeating images of the EPROM appear throughout all supervisor space. A local I/O write is available which causes the EPROM to assume its normal location.

13.2 Interrupt functions

13.2.1 Interrupting devices

The 68020 receives interrupt request signals from three sources.

1. The 8030 device on the **Diagnostic Attachment** generates a level 7 non-maskable interrupt request.
2. The **command register** also generates a level 7 interrupt request.
3. Seven one-bit **Interrupt request registers** may be loaded from the bus at any time. In normal operation, the interrupt dispatcher sets and clears these registers.

13.2.2 Autovectors

Interrupts are acknowledged on-board by the appropriate autovector.

13.2.3 Interrupt Service

The service routines must turn off the request because the acknowledge cycle leaves the request alone. For the command register interrupt, the service must also increment the

free-buffer count (touch an on-board address) to allow a new command. Other boards attempting to write a command will be suspended in wait-states and may time out.

13.2.4 View From Other Boards

The current value of the IPL processor inputs is available in the status value readable from the bus.

13.3 Cache

The cache has one 64 kB direct mapped set.

13.3.1 Data Store

This block of 64 kB of fast memory retains local copies of main memory locations which have recently been read.

Each "cache entry" or "line" contains 16 bytes of data, and 46 bits of tag information. There are 4096 lines in the cache. The cache is addressed logically, but the tags process both logical and physical addresses.

13.3.2 Processor's Tag

This block of 4K by 21 bits of very fast memory retains a tag word associated with each entry in the data store. It also keeps a valid bit for each entry. If the valid bit is cleared, the entry is forced to miss.

13.3.3 Controls

The following controls are provided for the supervisor to manage cache operation.

13.3.3.1 Enables

Five bits in a single register enable:

1. Read-hits: This bit allows memory reads to hit cache.
2. Write-hits: This bit allows memory writes to hit cache. It does not affect the operation of the write buffer.
3. Fill: This bit allows any reads which miss cache to "fill" cache, that is replace the current entry. Finer control over cache fill is provided in the page descriptors of the MMU.

4. Spyin: This bit allows the spy to capture hits from the CSS Bus.
5. Spyout: This bit allows the spy to unload any hits into the cache's tag's valid bits.

The enables are broken out separately so faults can be isolated. In normal operation all five enables would be switched simultaneously.

13.3.3.2 Read-Modify-Write

The TAS, CAS, and CAS2 instructions generate strings of memory cycles which must lock the locations they are using for the duration of the string. To guarantee the locked copy of data is the most up-to-date copy, these cycles bypass the cache and operate only on main memory. The appropriate line in cache is marked invalid in both tags. (The read-modify-write cycle is intended only to arbitrate for critical resources. Its execution can take the time of ten or more cache hits. It also generates bus and memory traffic. For most purposes, the arbitration procedure should spin reading the location, and attempt the R-M-W cycle only when the location changes. By this method the cache can be used to eliminate unnecessary memory load.)

13.3.4 Spy: CSS Bus's Tag

13.3.4.1 Purpose

The spy maintains the appearance of one pool of global memory shared by all the processors on the CSS bus, by performing two functions.

1. Flushing stale entries from the cache. These entries become stale when some other board writes into the corresponding main memory entry.
2. Ensuring no duplicate entries get loaded into the cache. (Since the cache holds 16 pages worth of data, the same physical location may appear in 16 different users' virtual spaces in different virtual pages.) As the cache fills an entry, the spy looks up any older entry (which was loaded according to some other page table) and invalidates it.

This allows processes (including DMA transfers) to communicate through shared memory.

13.3.4.2 Description

This subsystem contains another tag on the cache's data store. It captures write addresses from other modules on the backplane bus, and read-miss addresses from our own module, and compares them for hits. If it gets a match, it seizes the cache and clears the valid bit for that entry.

13.3.4.3 Performance

A write through (no waiting) to virtual memory by the '020 on one PM could take up to 2 microseconds (us) to propagate into the cpu's tag on all the other ones. Two us is enough time for any one '020 to execute ten or more typical instructions. Arbitration for semaphores should use the synchronized bus operation available with the TAS, CAS, and CAS2 instructions.

13.3.4.4 Contents

Each spy entry contains the following.

1. 24 bits selecting a physical page number
2. 4 bits representing bits 15:12 of the cache address containing the associated entry.
3. One bit indicating the entry is valid.

There are 4096 entries.

13.3.4.5 Write Function

Each write request which appears on the CSS Bus is captured in the Bus Receive register. These requests contain the slot, physical address, and data to be written. The spy processes each write request by using bits 15:4 of the write request to look up the tag word (physical address) of our cache entry. If the looked up tag word matches bits 35:16 in the write request, then we have a hit, that is, some other module has written new data into a location we have cached.

When there is a hit, the spy generates a cache address to invalidate. The cache address has bits 15:12 from the logical address (these were stored when the cache filled) and 11:4 from the write request. The spy puts the cache address into a first-in-first-out (FIFO) memory, and requests cache. The cache arbiter recognizes the request and grabs the cache away from the 68020 as soon as possible (up to 3 cpu clocks delay). Once the spy owns the cache, it unloads the FIFO, using each cache address to clear a valid bit. It gives the cache back when the FIFO is empty.

13.3.4.6 Read function

The spy's read function allows the logically addressed cache to be larger than the page size. It resolves the ambiguity introduced by the possibility that several users may share a single physical page, but use it from different logical addresses. (This function is known as BQFP.)

When the cache fills, it stores logical address bits 15:12 in both tags. Later, when it misses and requests a replacement from main memory, the read request address is processed by the spy, almost like some other module's write request. If the spy finds a valid-bit true for that location, it means some other process had cached that physical location in some other place in cache.

When the spy generates that hit, it looks up the logical address bits 15:11 of the old cache entry. Those logical bits are fed through a fast register and used to invalidate the old entry. The invalidate takes place while the 68020 is waiting for the memory to respond.

13.4 Memory Management Unit

The MMU uses a standard technique of "Translation Look-aside Buffering" to manage the virtual memory spaces. In this scheme, the operating system maintains descriptor tables, in main memory, for each process. The lookup table on the PM is simply a direct mapped, write through cache for the most recently used page descriptors.

13.4.1 Tables

13.4.1.1 Where the MMU Finds Page Descriptors

The virtual address calculated by software combines with the supervisor bit to determine where the PM seeks a page descriptor. When a user process is executing and refers to a page whose descriptor is not currently buffered (in the TLB), the PM performs a **table walk** to fill the TLB with the required descriptor. To "walk tables" the PM suspends the 68020's memory cycle to read two words from main memory. The first word is a **page pointer table** entry, and the second is a **page table entry**. The page table entry goes into the TLB where it can be used to address memory. Finally the 68020 regains control and uses the new descriptor.

The PM finds the page pointer table entry in its table in main memory as follows: A **Root Pointer Register contains bits 5 through 11 of the physical address of the base of the page pointer table. The offset into the page pointer table (the LEAST significant part of this physical address, bits 11 through 2) comes from the most significant part of the software's logical address: bits 30 through 22.**

Once the pointer table entry is read, the PM uses it and the software's logical address bits 21 through 12 to read a page descriptor from main memory. The logical address bits become an offset into the 1024 entry table.

13.4.1.2 Supervisor Root Pointer

This register contains the physical address of the first entry in the supervisor's Page Pointer Table.

13.4.1.3 User Root Pointer

This register contains the physical address of the first entry in the user's Page Pointer Table.

13.4.1.4 Page Pointer Tables

The kernel and each user process has exactly one of these 4 kB data structures in main memory. They contain the base addresses of the individual page tables.

13.4.1.5 Page Tables

Each page table contains 512 page descriptors.

The page descriptors are four byte words. Each descriptor gives the physical address, rights, and statistics for one page of one process's virtual space.

13.4.2 Address Translation

The MMU performs address translation according to the information it reads from the Page Tables. It uses the virtual address from the 68020 to look up the correct page descriptor. That descriptor contains the physical address for the memory cycle. A physical address contains a four bit slot ID, a twenty bit page address, and a twelve bit offset within the page.

13.4.2.1 Slots

There are four bits of slot address. These comprise the highest order bits of the physical address.

13.4.2.2 Physical Address

The twenty bits in this field may be considered to be the less significant than the slot address.

13.4.2.3 Low Order Address

The least significant twelve bits of the logical address from the 68020 bypass the MMU. They specify a byte within a 4 kilobyte (4 KB) page.

13.4.3 Rights Checker

The rights of access for each page are specified in each descriptor. The MMU checks these rights during each memory reference. It generates a bus error exception when a process exceeds its rights. Bits in the fault register indicates which permissions were violated. The MMU protects the memory image from attempted write without permission.

13.4.3.1 Read Permission

This bit enables the user to read data from the page.

13.4.3.2 Write Permission

This bit enables the user to write data on the page.

13.4.3.3 Execute Permission

This bit enables the user to execute the page.

13.4.4 Fault Registers

Warning: An unused prefetch with a violation will update the violation bits in the FR without generating a bus error exception. This syndrome is called a "spurious read violation" in this document.

The fault register always reflects the most recent information. Some bits accumulate until cleared; in others the information from the previous error is destroyed.

Most information for fault recovery is available on the stack following a bus error exception. The fault register captures the following additional information.

13.4.4.0.0.1 Accumulating Bits

These bits are all cleared by a certain local I/O write, Each is set whenever a bus error occurs, if its particular condition is true.

1. A bit which says "this is not the first bus error since the register was last cleared." This bit may be used to determine whether a spurious read violation has sent a bus error which was ignored.

2. A bit showing uncorrectable memory error encountered
3. A bit showing bus transmission error (That is, the response from the bus was garbled or the command was garbled.)
4. A bit showing page-not-resident
5. A bit showing request timed out

13.4.4.0.0.2 The Bits Which Change With Each Bus Error

1. Three bits show the permissions at the location which bus errored.
2. Wrong coprocessor type indicates an f-line instruction with coprocessor id not equal to 001 was attempted. This gives the f-line instruction exception, not the bus error exception.
3. Wrong Space indicates a MOVES instruction to space 0, 3, or 4 was attempted.

The fault registers may be read repeatedly at any time because reading does not change the state of the register or its ability to capture data.

13.4.5 Statistics

These bits are maintained by the MMU for use by the supervisor in deciding which pages to swap in or out.

13.4.5.1 Accessed

This bit indicates a valid read or write has been made to the page. Its state is not changed if there is no permission for the attempted access.

13.4.5.2 Written

This bit indicates a valid write has been made to the page. Its state is not changed if there is no permission for the attempted write.

13.4.6 Attributes

13.4.6.1 Page Resident

The Page Resident (PR) bit in a page descriptor indicates to the MMU that the desired logical page is currently in memory and may be accessed by the user to whose table the descriptor belongs. If the user attempts to access a non-resident page, the MMU generates the Bus Error Exception and sets the Page Fault bit in the fault register.

Note that the MMU must load in the new descriptor to test whether it refers to a resident page. Therefore the old descriptor is discarded from the buffer. However, it does not set the accessed or written bits, in main memory or TLB. Since the second table walk read is an atomic read-modify-write, the MMU writes exactly what it read back to main memory.

13.4.6.2 Cacheable

The Cacheable attribute bit in a page descriptor indicates to the cache controller that lines from this page may be filled in the cache. This bit lets the supervisor control cache fill activity. There is no per page control on cache hits; that is, once a line has filled, it will hit if cache hits are enabled.

Overriding The Cacheable Attribute

The cacheable attribute is ignored when the processor executes a read-modify-write cycle. R-M-W cycles never fill cache. Also, a bit in the cache control register can defeat all cache filling. (See processor and cache descriptions.)

13.4.7 Translation Lookaside Buffer (TLB)

13.4.7.1 Store

The TLB store is a small memory which contains recently used page descriptors. The MMU automatically loads new descriptors into it on demand.

13.4.7.2 Tag

The TLB Tag contains three fields:

1. Corresponding logical address bits 31:22
2. A bit showing the entry is valid for the current user. This bit must be flushed when switching context from one user's page table to another.
3. A bit showing the entry belongs to supervisor.

A flush function is available to the supervisor. This function marks all entries in the TLB "invalid" so that the new user's page table will be used. A second image of the entire translation store (page descriptors, root pointers, io section descriptors, and waste) appears in the local io area. Writes to this second image flush the TLB's tag. The flush function executes in less time than a main memory read.

13.4.7.3 Error Handling

The MMU generates bus error exceptions due to permit violations, page-not-resident, or descriptor acquisition failure. There are three failure modes available in acquiring a descriptor:

1. NAK: the memory board rejected a read or write command because the parity it received was wrong.
2. ERROR DATA: the memory board had an uncorrectable error reading the PPTE or the descriptor.
3. TIMEOUT: the table-walk sequence took more than a millisecond for some reason and the processor module gave up.

When a non-resident page causes a user bus error exception, or when a table-walk fails, the operating system must invalidate the appropriate entry in the TLB's tag.

13.5 Input/output mapper

13.5.1 Purpose

The CSS bus requires 36 bits of address. The off-board I/O is assigned 1 gigabyte (GB) or 30 bits of address. The 1 GB space is mapped in 1 MB sections onto the backplane address space, by a lookup table.

Only the supervisor may access I/O space. Users must access I/O devices off-board by having them mapped into user virtual memory, not I/O space. (This gives user I/O to devices such as frame buffers the performance advantage of cache. Of course, user devices which require polling of status or shared memory must be marked non-cacheable.)

13.5.2 Contents

There are 1016 usable entries in the lookup table. Each entry has two fields.

13.5.2.1 Backplane Address

16 bits of slot and high order address.

13.5.2.2 Valid Bit

This bit indicates the I/O page is mapped to some device on the backplane. An attempt to access an invalid I/O page generates a bus error exception.

13.5.3 Errors

If no response comes back from an I/O request after TBD microseconds (us), or if an error response comes back, a bus error exception is generated.

13.6 On-board I/O, interrupt, and diagnostic features

13.6.1 Local I/O

The on-board, or local I/O uses up 4 MB, therefore the last 4 sections in the mapped I/O are unusable. One MB is assigned to the diagnostics module. One MB is assigned to the following on-board features. Two MB are reserved.

13.6.1.1 Resets

Write-only registers which reset the following PM features are provided in the local I/O space:

1. 68881 FPU chip
2. Clear the "somebody's selected the FPU" bit
3. Clear the "there's been ... since ..." bit in the bus error exception register
4. TLB valid-bit
5. FIFO
6. unlock the command register (input buffer)

13.6.1.2 Enables

13.6.1.2.1 Cache Enable Register

1. Fill: allows the cache to replace a stale (tag word doesn't match) or invalid entry when it misses and the page is cacheable.
2. Read-hit: allows the no-wait memory read cycle to occur. If hit enable is off, memory reads always go to the bus. This bit does not affect the MMU or the fill mechanism.
3. Write-hit: allows write cycles which hit cache to write in the cache. Write cycles always go through to the system bus. This bit does not affect the MMU or the fill mechanism. It also does not affect the write buffer.

4. Spyin: allows the bus' tag to load hit locations into its FIFO.
5. Spyout: allows the bus' tag to unload its FIFO into the cpu's tag

13.6.1.2.2 One Bit Enables

These one bit registers respond only when read or written as bytes.

1. Write buffer enable: When true, allows the cpu to proceed after writing without waiting for the system's bus cycle to complete. When false, makes the cpu wait for main memory to acknowledge each write.

2. FIFO test enable: Causes the FIFO control to interpret all transmissions by this module to be write commands which hit cache (regardless of their type or the contents of the bus' tag). Bits in the transmission which would be address bits <15:4> of a write command are loaded into the FIFO.

3. PROM decode When true, the on-board PROM occupies its normal address range. When false, the PROM occupies kernel virtual space as well.

4. Read-violation enable Allows the MMU to enforce the read permit bit in the page descriptor.

5. Write-violation enable

6. Execute-violation enable

7. Interrupt enable (from command register)

8. Command register mode bit. When set, the command register may be overwritten by any bus module at any time. When clear, the command register locks after each write, and NAKs further writing. The NAK causes the sender of the command to take a bus error exception.

9. LEDs 1 and 2

13.6.1.3 Status and Fault Registers

13.6.1.3.1 Bus Error Exception Register

Here are the bits in the bus error exception register.

1. Read-violation
2. Write-violation
3. Execute-violation
4. Page table not resident
5. Page not resident
6. User out-of-range
7. No coprocessor
8. Timeout
9. NAK from CSS Bus (the destination rejected the command)
10. READ BAD DATA from CSS Bus
11. There has been a bus error since software last cleared this bit

The register may be read any time and repeatedly. It always shows the cause of the most recent bus error sent into the processor chip, even if it was a "spurious read cycle" faulting. The "bus error since" bit is for detecting the spurious read fault.

13.6.1.4 Status Register

Here are the bit fields which are available in the general status register:

1. Pending interrupt level (3 bits)
2. Spy not empty
3. FPU is installed
4. Bus input buffer (command register) not empty
5. Clock option installed (2 bits)
6. FPU touched since this bit was last cleared
7. Slot number (4 bits)

13.6.1.5 Command Register

This register retains the eight bytes of the last write command sent into the board from the

bus. When a module writes into the board, this input buffer becomes locked so any other modules attempting to overwrite will enter wait states. A level 6 interrupt is generated when the command register is written.

13.6.2 Test and Initialization Features

13.6.2.1 Cache as RAM

The 64KB cache data store may be used as RAM when accessed at this location.

13.6.2.2 TLB Test

The 4096 32 bit registers of the page descriptor store may be accessed at this location. These registers respond only when addressed as aligned long words. In operation only the first 1026 registers are used.

13.6.2.3 CPU's Tag Test

The 4096 25 bit registers in the cpu's tag may be accessed here. These registers respond only when addressed as aligned long words.

13.6.2.4 Bus's Tag Test

The 4096 28 bit registers in the bus' tag may be accessed here. These registers respond only when addressed as aligned long words.

13.6.2.5 TLB's Tag Test

The 1024 11 bit registers in the the TLB's tag may be accessed here. These registers respond only when addressed as aligned long words. The active 11 bits are left justified in the long word. The other bits read unpredictable data.

13.6.2.6 Bus Monitor

This 10 byte register contains the module's most recent transmission on the CSS Bus. It can be used to determine if the bus drivers and receivers are working. If the bus monitor is enabled, the command register may be overwritten whenever the module transmits anything.

13.6.2.7 Direct Bus Interface

These features allow the supervisor to control the bus interface directly. They may not be useful when executing from main memory. They are intended for the PROM code to generate responses for any commands which are not taken care of by the bus interface hardware.

13.6.2.7.1 Transmit Registers

These 4-byte write-only registers contain the next transmission the pm will send on the CSS Bus. They are overwritten by the automatic functions main memory read, write, modify, cache fill. The registers are: data, address, and type.

13.6.2.7.2 Send Button

This write-only register causes the contents of the transmit register to be sent on the CSS Bus.

13.7 Write Buffering

13.7.1 Buffer

The write buffer is a temporary holding register. Its function is to hold the address and data of a pending write cycle. This allows the 68020 to continue reading cache and using the coprocessor while a write is pending.

13.7.2 Control

An enable bit appears in the (supervisor's) I/O space which controls the action of the write buffer. If the buffer is disabled, then the 68020 must wait for the write to complete before beginning its next cycle. (This feature is for diagnostics.)

13.8 Reset Functions

13.8.1 Power-up Clear

An on-board sensor generates a power-up-clear signal when power is first applied. A bus command is available which has the same effect. This signal resets certain features as follows:

1. 68020 CPU chip
2. Memory buffer full flag
3. PROM decoder disabled (PROM is everywhere)

PROM code is responsible for resetting anything else. Specifically, it should reset the FPU.

13.8.2 68020's RESET Instruction and Pin

The 68020 executes its RESET instruction by driving its reset pin (normally an input) low for at least 20 microseconds (us) and up to 200 us (clocks 2.5 to 25 MHz). A bit in the board status register lets other CSS modules observe the state of the 68020's reset pin.

13.9 Diagnostic Attachment

This board plugs onto a connector on the PM's accessible edge.

13.9.1 PROM

The board has four locations for 27512 or 27010 EPROM. These devices hold 64 KB or 128 KB each. They are organized 1 byte wide so no word splitting is required. This makes them run slowly unless the on-chip cache is on.

13.9.2 RAM

The board contains 64KB of static RAM. There is no parity check. They are organized 1 byte wide.

13.9.3 RESET Button

The board has a momentary switch which asserts RESET to the 68020. No other state of the PM is affected. A read/write bit is provided on the diagnostics board. It's cleared by power up, and set by the RESET button, as well as accepting writes.

13.9.4 Serial Port

A Zilog (AMD, SGS) 8031 Asynchronous Communications Chip is provided for use as a diagnostic console and download port. EIA line driver and receiver are provided for use with ASCII terminals, but the full RS-232-C spec is not met. There are two ports, both wired like terminals.

13.9.5 Signature Probe

A polynomial generator chip is provided, with movable probes for clock and data. This instrument will be used by board techs to test nodes on the PM at speed. Diagnostic software will direct the probing and compare expected signatures.

14. Memory Module

The memory module is the high speed system memory for the System 3000. It has on-board 32 bit error detection and correction (EDAC), refresh logic, and partial write logic. There are two interleaved arrays which can read, write, partial write, and refresh independently. This is accomplished by interleaving on address bit 2 so that even and odd four byte addresses go to opposite arrays. Because read operations return 8 bytes of data, they require both arrays to work together. Each array contains four 39 bit wide banks. The 39 bits include 32 bits of data and 7 check bits. Depopulated options of the module are possible by stuffing one, two, or all four of the banks in each array. This yields an 8, 16, or 32 megabyte memory module.

The memory chips used are 1 megabit dynamic rams (1 Mb DRAMs). When 4 megabit DRAMs become available, they may be used in the module with only jumper changes. The stuffing options will then be 32, 64, or 128 megabytes per module. The access time of the DRAMs used must be no more than 100 nanoseconds (nS). Nibble mode access and CAS before RAS refresh are the other requirements of the DRAM chips. The ZIP (zig-zag in-line package) style is used to save module space and make the arrays more compact so that the control lines will be shorter.

The size of the data used by the memory module depends on the type of access. Write cycles use a 32 bit data word. Partial write cycles must be generated by the memory module's control logic if it detects a write of less than four bytes. Read cycles return 64 bits to the requester. Burst mode is available on a read cycle so that the memory module returns 16 or 32 bytes of data for one read request. Write and 8 byte read cycles last five 50 nsec system clock periods. Partial write cycles are nine clocks long. Refresh cycles take six clock periods. The 16 and 32 byte reads take eight and 14 clock periods each, respectively.

Parity is checked on input to the module and parity is generated on output. Status registers are readable by other modules in the system to determine the memory size of the module, find the address of an error, see if the memory module is in diagnostic mode, or various other module characteristics. Control registers are available so that other modules may enable or disable the memory array, put the module into diagnostic mode, inhibit error correction, or other functions.

14.1 Memory Module Description

14.1.1 Memory Module System Bus Interface

Command, address and data if applicable get clocked into the bus input registers every 50 nsec by the 20 MHz system clock. The slot identifier field is compared with the memory module's slot number to see if the command is meant for this memory module. If there is a match, the top five bits of the address are checked for equality to all ones or all zeroes. If the bits are all ones, the status or control registers are to be accessed. If the bits are all zeroes, the access is to the memory array. If the bits are neither all zeroes or all ones, a NACK will be issued. If the transaction is not meant for this memory module, no action will be taken and new a new command, address, and data will be clocked into the bus

input register on the next rising edge of the system clock. Array accesses when the input pipeline register for the arrays is full and accesses to nonexistent status or control registers will get a NACK response from the memory module. The input register may not be used as a holding register for the memory module because it always gets new data clocked into it on every system clock rising edge.

14.1.2 Status And Control Registers

There are several status and control registers on the memory module. Each bit that can be written can also be read. The registers are all byte writable using pseudo read-modify-writes. The registers are as follows: Status Control Base Address Error Address Error Board I.D. Error Syndrome Read Error Check Bits Last Check Bits Written Diagnostic Check Bits

14.1.3 Memory Array Accesses

When a valid memory access gets clocked into the input registers, the address, and data if applicable will get clocked into the input pipeline register on the next system clock's rising edge. The input pipeline register holds commands until the arrays are ready to process them.

14.1.4 Input Pipeline Register

The pipeline register is two units wide and two units deep. The units here are 32 bits of address, 32 bits of data if applicable, and a few appropriate control bits. This register is implemented with nine 520 pipeline registers. Each 520 has two halves, A and B, one for the odd array and one for the even array. Depending on whether address bit 2 is even or odd, the address, data, and control will be clocked into the A or B half. Clocking data into the 520 is done on the rising edge of the 20 MHz system clock, using the I1 and I0 control signals as shown below:

I1	I0	ACTION
0	0	LOAD A
0	1	LOAD B
1	0	HOLD
1	1	NOT USED

The A and B halves each have two registers, A1 and A2, and B1 and B2. Each half is set up like a FIFO with new data pushing the previous A1 or B1 into A2 or B2 and the data in A2 or B2 is over written. The outputs of these registers feed a four to one multiplexer which is controlled by the signals S1 and S0. One of the four registers is always being selected from the multiplexer. The encodings are as follows:

S1	S0	OUTPUT
0	0	B2
0	1	B1
1	0	A2
1	1	A1

The default, when no outputs are being used is 00. To differentiate the default code from the READ.B2 code, the control signal LATCH.B must be tested. The output enable of the pipeline register is always active so that it always drives the bus.

14.1.5 Input Pipeline Register Control

The input pipeline register is controlled by four PALs.

The LOAD PAL generates the I0 and I1 signals used to steer input data to one, the other, or neither halves of the register. Four signals, one for each register, are also generated to tell if the register has any data presently in it. These signals are A1.FULL, A2.FULL, B1.FULL, and B2.FULL. The I0 and I1 signals can also be called LD.A1* and LD.B1*, respectively. The two signals are mutually exclusive and when neither is active the register is in the HOLD state.

The PRIO PAL continuously keeps a record of which register has the oldest data in it and therefore has the highest priority.

The QRD PAL keeps track of which registers contain read commands.

The READ PAL generates the S0 and S1 signals used to select the correct register for the pipeline register to output. Also generated are LATCH.A and LATCH.B, which latch the input registers to the A and B arrays. The A register is enabled when the pipeline register is outputting the A1 or A2 register and the B register is enabled when the pipeline register is selecting the B1 or B2 register. Both A and B registers are enabled on a read, so that the two arrays are synchronized for the read cycle.

14.1.6 Array Input Latches

The output of the pipeline registers feed the 74F373 array address and data latches. The outputs of the data latches go directly to the DRAM chips data inputs. The outputs of the array input address latches get multiplexed from twenty row and column address lines to ten multiplexed address lines. Multiplexing is done with 74F241s, by connecting a row and a corresponding column address to each input having complementary output enables with their tri-state outputs tied together.

14.1.7 Array Organization

Each array contains four banks of dynamic RAM. The banks are 39 bits wide and 1 megabit deep. The 39 bits consist of 32 bits of data and 7 check bits. Depopulation of the memory module is achieved by removing one, two, or three banks from each array. In

this way, two way interleaving is preserved with any stuffing option.

14.1.8 Array Control

Each of the two interleaved arrays has five control PALs for generating DRAM control signals. The two arrays can therefore operate independently on all cycles except a read.

The RASPAL generates RAS and CAS for each of the four banks in the array.

The MRGPAL generates the output enables for the array data input registers and the 632 data latches.

The EDACPAL generates the control signals for the 632 EDAC chip and the check bit registers.

The WEPAL generates the write enable signals for each bank of the memory array as well as the array ready signal.

The CTLPAL contains a state counter and outputs signals to tell whether the cycle is a read, write, refresh, or partial write.

14.1.9 Array Outputs

The output of the DRAMs in the array feed a 74F244 whose outputs are shared with the bidirectional data and check bit pins of a 74AS632 EDAC chip, the input of another pipeline register used for output read data, the output of the array input data register, and the input to the DRAM array.

14.1.10 Output Pipeline Register

The pipeline register at the output of each array is organized as 32 bits wide and four deep. It is always used simultaneously with the output pipeline register of the other array. This is because the smallest read unit is 64 bits.

Control for this 64 bit wide register is handled by the OUTCTL PAL. Because of the simple organization of this pipeline register only three signals are needed to control it. The LOAD* signal outputs from the PAL and connects to both the I0 and I1 inputs of the 520. When active, the registers are loaded like a FIFO, when inactive, the register is in the hold state. The multiplexor signals, S0 and S1, are used as in the input pipeline register, except that they always start at 00 and increment to output the number of 8 byte words in the register.

14.1.11 Read Output

The output of the pipeline register goes into a 74F374 register for synchronization with the system clock before being output through a 74F244 and onto the system bus. On a read cycle, the memory module makes a request one clock before it loads the last 8 bytes of read data into the pipeline register. The GRANT from the arbiter goes to the J input of a 74F109 J-K flip-flop whose output is the output enable for the 74F244 output drivers. The K input of this 74F109, RD.DONE*, comes from the OUTCTL PAL, and becomes

active after the last read data bytes are clocked into the output register. In this way, the memory module drives the bus for 1, 2, or 4 bus clocks, depending on whether it is an 8, 16, or 32 byte read. The clear of the 74F109 is connected to the BOARD.DISABLE line. With the 74F109 always cleared, the memory module never drives the system bus. If the GRANT from the arbiter comes immediately, the data can be clocked into the data output register during the same clock period that it is being driven onto the bus.

14.2 Cycle Descriptions

The memory module supports six types of memory operations, including three different types of read cycles. All the cycles consist of a certain number of states, numbered S0, S1, S2, and so on. Each state is one system clock period, or 50 nsec. The cycles can be in any sequence and may start immediately after the preceding cycle's last state. The types of cycles with brief descriptions are as follows.

14.2.1 Write Cycle

The write cycle is the late write type. This means that while writing, the data output of the DRAM could be driving. This is acceptable because the tri-state drivers on the outputs of the DRAM array are turned off. The write cycle lasts five clock periods and will occur only if the LENGTH field does not equal one, two, or three bytes.

The write cycle begins the assertion of RAS in state S1. CAS is asserted in S2 and WE in S3. All three signals are de-asserted in S4. Write data and check bits are driven from S1-1/2 until S4-1/2. The signals EDAC.S1 and EDAC.S0 are both inactive for the duration of the cycle, putting the 74AS632 EDAC chip in the generate mode. The check bits generated are valid about halfway through S2.

14.2.2 Read Cycle

The normal read cycle returns 8 bytes of data to the requester. The LENGTH field is checked and must equal five (8 bytes) for this this type of read. Read cycles are different from the other cycles in that S5, the last state in the cycle, can be the S0 of a back to back operation. This is because only the 74AS632 EDAC chip and the output pipeline register are busy during S5, and the array has already been precharged. Thus, even though the 8 byte read cycle uses five clock periods, it uses states S0 through S5 with S5 as a possible overlap to the next cycle's S0. To simplify burst mode reads, the read data is corrected whether or not an error has been detected. Therefore, most of the time correct data is being automatically corrected again. This procedure adds and extra clock period to the read cycle but it is made up for in control logic simplification.

The read cycle begins with RAS asserted in state S1. CAS is asserted in S2 and both signals are de-asserted in S4. Read data is valid just before S4. The EDAC chip control signals start with EDAC.S1 and EDAC.S0 equal to zero. In S2, EDAC.S1 is activated to put the EDAC chip into the detection mode, and EDAC.S0 is activated in S4 for the correction mode. The rising edge of EDAC.S0 latches the read data from the DRAMs. Both signals are deactivated after S5. The corrected data is driven by the 74AS632 EDAC chip starting in S5 and is valid about halfway through S5. The output pipeline register

latches the corrected data on the next rising edge after S5.

14.2.3 16 Byte Read Cycle

The 16 byte read begins the same as a normal read cycle, but it does a nibble mode access when the 8 byte read would have completed. This type of read lasts 8 clock periods. Like the 8 byte read, the 16 byte read shares its last state with the following cycle, if it is back to back. Again, even though the 16 byte read cycle uses eight clock periods, it uses states S0 through S8, with S8 as a possible overlap to the next cycle's S0. When the cycle is a read and the LENGTH field is six (16 bytes), a 16 byte read will be done.

The read cycle begins with RAS asserted in state S1. CAS is asserted in S2 and only CAS is de-asserted in S4. Read data is valid just before S4. The EDAC chip control signals start with EDAC.S1 and EDAC.S0 equal to zero. In S2, EDAC.S1 is activated to put the EDAC chip into the detection mode, and EDAC.S0 is activated in S4 for the correction mode. The rising edge of EDAC.S0 latches the read data from the DRAMs. EDAC.S0 is deactivated after the rising edge of S6. The corrected data is driven by the 74AS632 EDAC chip starting in S5 and is valid about halfway through S5. The output pipeline register latches the corrected data on the rising edge of S6. The nibble mode access begins with CAS being asserted again in S5. Read data is valid just before S7. The EDAC chip control signal EDAC.S0 is activated in S7 and its rising edge latches the read data from the DRAMs. RAS and CAS are deactivated in S7. EDAC.S0 and EDAC.S1 are deactivated on the next rising edge after S8. The corrected data is driven by the 74AS632 EDAC chip starting in S8 and is valid about halfway through S8. The output pipeline register latches the corrected data on the next rising edge after S8.

14.2.4 32 Byte Read Cycle

The 32 byte read begins the same as a normal read cycle, but it does three nibble mode accesses when the 8 byte read would have completed. This read cycle takes 14 clock cycles to complete. Like the 8 byte read, the 32 byte read shares its last state with the following cycle, if it is back to back. Again, even though the 32 byte read cycle uses 14 clock periods, it uses states S0 through S14 with S14 as a possible overlap to the next cycle's S0. When the cycle is a read and the LENGTH field is seven (32 bytes), the 32 byte read will occur.

The read cycle begins with RAS asserted in state S1. CAS is asserted in S2 and only CAS is de-asserted in S4. Read data is valid just before S4. The EDAC chip control signals start with EDAC.S1 and EDAC.S0 equal to zero. In S2, EDAC.S1 is activated to put the EDAC chip into the detection mode, and EDAC.S0 is activated in S4 for the correction mode. The rising edge of EDAC.S0 latches the read data from the DRAMs. EDAC.S0 is deactivated after the rising edge of S6. The corrected data is driven by the 74AS632 EDAC chip starting in S5 and is valid about halfway through S5. The output pipeline register latches the corrected data on the rising edge of S6. The first nibble mode access begins with CAS being asserted again in S5. Read data is valid just before S7. The EDAC chip control signal EDAC.S0 is activated in S7 and its rising edge latches the read data from the DRAMs. CAS is deactivated in S7 and EDAC.S0 is deactivated on the

rising edge of S9. The corrected data is driven by the 74AS632 EDAC chip starting in S8 and is valid about halfway through S8. The output pipeline register latches the corrected data on the rising edge of S9. The second nibble mode access begins with CAS being asserted again in S8. Read data is valid just before S10. The EDAC chip control signal EDAC.S0 is activated in S10 and its rising edge latches the read data from the DRAMs. CAS is deactivated in S10 and EDAC.S0 is deactivated on the rising edge of S12. The corrected data is driven by the 74AS632 EDAC chip starting in S11 and is valid about halfway through S8. The output pipeline register latches the corrected data on the rising edge of S9. The third nibble mode access begins with CAS being asserted again in S11. Read data is valid just before S13. The EDAC chip control signal EDAC.S0 is activated in S13 and its rising edge latches the read data from the DRAMs. RAS and CAS are deactivated in S13. EDAC.S0 and EDAC.S1 are deactivated on the next rising edge after S14. The corrected data is driven by the 74AS632 EDAC chip starting in S14 and is valid about halfway through S8. The output pipeline register latches the corrected data on the next rising edge after S14.

14.2.5 Partial Write Cycle

The partial write cycle is like a four byte read followed by a write. When a write command is clocked onto the memory module, the LENGTH field is checked to determine if a partial write will be needed. If the LENGTH is 1,2, or 3, the lower two address bits and the LENGTH field are used to generate byte selects for the bytes that will be changed in the partial write cycle. The new data bytes are merged with corrected data read from the location to form the new 4 byte data word which is then written. The data that is read is corrected whether or not there is an error, just like a read cycle. If there is an uncorrectable error, the cycle continues, and the possibly incorrect data bytes are written back into the location. The next read to this location will produce an uncorrectable error if one of the bad bytes was rewritten. The partial write cycle lasts nine clock periods.

The partial write cycle begins with RAS asserted in state S1. CAS is asserted in S2 and both signals are de-asserted in S8. Read data is valid just before S4. The EDAC chip control signals start with EDAC.S1 and EDAC.S0 equal to zero. In S2, EDAC.S1 is activated to put the EDAC chip into the detection mode, and EDAC.S0 is activated in S4 for the correction mode. The rising edge of EDAC.S0 latches the read data from the DRAMs. EDAC.S0 and EDAC.S1 are deactivated on the rising edge of S6. The corrected data is latched in the output latches of the EDAC chip by the rising edge of LEDBO, which occurs after the rising edge of S6. The data bytes that are not being changed are then driven by the 74AS632 in S6, and the new bytes to be written are driven by the array input latches. Check bits are generated on the merged data and WE strobes for one clock period during S7. The various data output enables and LEDBO are turned off in S8, preparing the array for another cycle.

14.2.6 Refresh Cycle

The memory module uses CAS before RAS refresh to save address counter logic and extra address bus drivers. In the CAS before RAS refresh mode, an internal counter in the

DRAM increments with each refresh operation, insuring that all 512 rows will be refreshed automatically without having to externally drive the address lines. The 512 rows need to be refreshed each 8 milliseconds (mS), so refresh cycles must occur every 15.625 microseconds (uS). A counter clocked by the 20 MHz system clock counts to 312, producing a refresh request every 15.6 uS.

The refresh cycle is initiated by the refresh request, which is the latched carry output of a counter. If the array is idle, the refresh cycle will start on the next rising edge of the system clock. If there is another cycle in progress, the refresh cycle will not start until the present cycle has completed. If there is a refresh request active while another cycle is going, the array will not activate the READY signal until after the refresh cycle to avoid read synchronization problems. The refresh request gets cleared by S3 of the refresh cycle.

The refresh cycle is six clock periods long. It begins with the activation of CAS in S1, and then RAS is asserted in S2. CAS is then de-asserted at S3, and RAS is deactivated during S5. The array is ready to begin a new cycle after S6.

14.3 Error Handling

Single bit or uncorrectable errors may be detected on a read or a partial write. Single bit errors are corrected and an interrupt is generated so that a CPU may keep track of corrected errors. The CPU that handles the interrupt may read the syndrome, which points out the bit in error, the address that the error occurred at, and the slot i.d. of the module that was performing the read. Double bit errors will always be detected but can not be corrected. Multiple bit errors may be detected but will not be corrected. Reading the error information kept for an error clears the registers that hold this information. Information on additional errors which occur before these registers are cleared is lost, that is, no new error information will be latched until the old information is read.

Single bit errors do not affect the operation of a read or partial write cycle, because data correction takes place automatically. On a read cycle, an uncorrectable error forces a special message to be returned with the bad data. Although there is no interrupt generated, the same error information is available as with a single bit error, except that the syndrome now only states that a multiple bit error has occurred. On a partial write, the data bytes are written even if there is an uncorrectable error. It is possible to write bad data back into the location. This will not be detected until the next read to this location, when an uncorrectable error message will be generated if the bits in error were written back. Single bit or uncorrectable errors will not cause a 16 byte or 32 byte read to abort before completion and only the error information for the first error encountered will be kept.

14.4 Diagnostics

Diagnostics will do write and read tests with various patterns to find stuck data bits, shorted data bits, stuck address bits, shorted address bits, and data bits shorted to address bits. The address and the bit in error will be provided to the diagnostic user. Partial writes will also be used in place of writes to test the byte merging logic. The EDAC operation will be tested to see if the 74AS632 EDAC chip can generate correct check bits on a write and partial write, detect single bit and multiple bit errors, correct single bit

errors, generate an interrupt on a single bit error, and return an uncorrectable error message on a multiple bit error. Read, write, and partial write loops will also be available to the diagnostic user. All the control and status registers will be accessible and testable by the user.

15. Service Processor Module

15.1 Service Processor Module Description

The Service Processor Module (SPM) attaches to the system bus and provides several service functions to the system. These functions include a service processor, IPL ROM, local diagnostic RAM, time of day clock, clock interrupt source, front panel interface, power supply and power system interfaces, system temperature monitoring, system console interfaces, floppy disc interface, CSS bus interface, and a system wide interrupt dispatcher.

15.2 Service Processor

The service processor provides the system control function. The processor includes a 68020 running at half the system bus clock rate, EPROM, RAM, SCC's for the console, remote console, system log printer and ups, a clock/ calendar chip, mapping registers for the system bus and a system bus interface.

Part of the service processor is powered separately from the rest of the system. Its power supply is ON if the system is plugged into a live circuit and the and the system's circuit breaker is ON. The main power supply is turned ON and OFF and margined by the service processor.

Inputs from the front panel are available to the service processor. The RESET button on the front panel resets the service processor.

Inputs from sensors around the system allow the service process to monitor the status of the system. Sensors monitor the temperature of air entering the system, leaving the logic chassis and around peripherals. Input line voltage is monitored as is the condition of any backup power sources. Each output of the main power supply is monitored.

The clock/calendar chip maintains the time of day and date for the system. It also provides the time base for system clock interrupts.

With the support of the mapping registers, the processor can address any part of the system.

The service processor runs the diagnostic monitor and controls the execution of diagnostics, standalone code and system boot or recovery.

15.3 Physical

Physically, the service processor consists of two boards, the main board, which plugs into the CSS backplane, and the real world interface board, which contains all connectors for

interfacing with the various sensors and communications devices supported by the service processor. The main board and the real world interface board are connected with a ribbon cable.

15.4 Floppy Interface

The Service Processor Module provides an interface to a 5-1/4" floppy disk drive. This device is used for system level diagnostics.

15.5 Interrupt Dispatcher

The interrupt dispatcher provides a centralized distribution point for assigning I/O interrupts to the Processing Modules. The interrupt dispatcher collects interrupts from throughout the system and redistributes them to Processing Modules. The intent of the interrupt dispatcher is to closely emulate interrupt handling characteristics of the single processor model in a peer processing system.

The interrupt dispatcher queues interrupts by priority level. The queue entry is the 32 bit interrupt information vector associated with each CSS interrupt. Interrupts are assigned in the order of priority to processing modules in a round robin manner.

16. I/O Module

16.1 I/O Module Description

The I/O module (IOM) attaches to the system bus and provides the communication path between the system bus and the link bus to an I/O subsystem.

A system may have up to four I/O Modules each connecting through a separate I/O Link (IOL) to a separate I/O Adaptor.

16.2 I/O Communications

The I/O Module provides the communication path between the system bus and the link bus to an I/O subsystem. A separate I/O Module and I/O Link is required for each I/O subsystem.

The IOM provides buffering for commands and responses at the connection to both the system and the IOL. The buffering is required to smooth the flow over the buses and for interfacing the 50 nS 64 bit wide system bus to the 100 nS 32 bit wide link bus.

17. I/O Link Bus

17.1 I/O Link Bus Description

The link bus is the point to point communication link between between an I/O module and an I/O adapter. The bus is synchronous with a clock rate half that of the system bus. The design clock rate is 10 MHz. The "data" path is 32 bits wide. Maximum bandwidth is 20 MB per second when writing to the system bus and 35.6 MB per second when reading from the system bus.

The operation of the link bus is somewhat similar to that of the system bus. It supports the same commands, responses and data sizes. Each port can transmit to the other port or itself. Each transmission consists of a source port address, a destination port address, a transmission type and 4 bytes of "data". The transmission type determines what "data" contains.

Each port (I/O module, I/O adapter) tells the other how many buffers it has available for READ or MODIFY commands and how buffers it has available for WRITE commands. When a command is sent from one port to the other, the appropriate available buffer count maintained at the sending port for the receiving port is decremented. When an input buffer in one port becomes available, the other port is told to increment the appropriate free buffer count. These counts provide flow control on the bus. Separate counts for READ or MODIFY and WRITE commands permit optimizing the interleaving of commands for maximum data rate with a minimum number of buffers.

An arbiter located on the I/O module controls access to the link bus. The I/O module has the higher priority for use of the bus. The arbiter grants the I/O module a time slot on the bus when ever the module wants to send a response or send a command to an available command buffer. The arbiter grants the I/O adapter all time slots not granted to the I/O module.

Each transmission is checked for errors. The source field, destination field, type field and each byte of data is protected with a parity check bit. Transmissions that are received with correct parity and valid type field are indicated by asserting ACK on the bus during the second time slot after the transmission. Transmissions that are received with one or more detected errors are indicated by asserting NACK on the bus during the second time slot. Unreceived transmissions are indicated by the lack of either ACK or NACK being asserted. Only the destination port asserts ACK or NACK for a transmission.

When a link bus transmission fails, retry is permitted, but not required. If retry is attempted, the port that issued the COMMAND resulting in the failed transmission restarts the COMMAND.

17.2 IOL Commands

The I/O Link Commands are as follows.

1. READ [1, 2, 3, 4, 8, 16 or 32 bytes]

   The operand of a READ command for 4 bytes or less must not cross a long word boundary. The operand of a READ command for more than 4 bytes must be quad word aligned and must not cross a 4 kilobyte page boundary.

2. WRITE [1, 2, 3 or 4 bytes]

   The operand of a WRITE command must not cross a long word boundary. The WRITE data must be aligned for a 4 byte memory port.

3. MODIFY [1, 2, 3 or 4 bytes]

   The MODIFY command is the first portion of a read/modify/write operation. It reads the operand and locks the slot to other read/modify/write operations. The operation is completed with a regular WRITE command that writes the operand and unlocks the slot.

   The operand of a MODIFY command must not cross a long word boundary.

17.3 IOL Responses

The I/O Link Responses are as follows.

1. READ DATA

   The response to a READ command is the requested data aligned for an 8 byte wide memory. A request for more than 8 bytes requires more than one RESPONSE.

2. ERROR DATA

   The response to a READ command encountering an uncorrectable read error is the requested data as read or after correction has been attempted and aligned for a 8 byte wide memory.

   ERROR [size = 0]

   A READ command that is somehow recognized as having failed to read anything is acknowledged by returning the ERROR response. The data field of the response is undefined.

17.4 Transmission Format

The format of the TRANSMISSION TYPE is as follows.

[3 bit type field]		[3 bit size field]	
0	Illegal	0	0 bytes
1	READ	1	1 byte
2	WRITE	2	2 bytes
3	MODIFY	3	3 bytes
4	READ DATA	4	4 bytes
5	ERROR DATA	5	8 bytes
6	ERROR	6	16 bytes
7	Illegal	7	32 bytes

17.5 I/O Link Signals

The I/O Link signals are as follows.

1. Destination Slot Address [CSS bus slot 3:0][I/O bus slot 4:0]
2. Destination Parity
3. Source Slot Address [CSS bus slot 3:0][I/O bus slot 4:0]
4. Source Parity
5. Transmission Type [5:0]
6. Type Parity
7. Transmission Data [0:31]
8. Data Parity [0:3]
9. ACTIVE
10. ACK

11. NACK

12. CSS Bus Clock

13. I/O Link Bus Clock

14. Free Buffer Count Control [6 signals]

15. Reset

16. Module Alive

17. Adapter Alive

18. Adapter Burst

19. Adapter Grant

20. System Power Fail

18. A1000 I/O Adaptor

18.1 A1000 IOA Description

The A1000 I/O adaptor connects an I/O subsystem of A1000 I/O cards in an I/O backplane to an I/O module plugged into the system bus. The I/O adaptor connects to its I/O module via two ribbon cables called the I/O link for transferring data to the I/O module and subsequently to the system bus. The A1000 I/O controllers connect to the I/O adaptor through the ICB and DTB busses for control and character data transfers and block data transfers respectively.

Functionally the I/O adaptor provides three major capabilities:

1. Individual data transfers between the system bus and the I/O cards through the ICB bus.

2. Interrupt generation to system bus processors on request from the I/O cards via the ICB bus.

3. Block DMA transfers between system bus memory and I/O cards through the DTB for four independent DMA channels.

The IOA logic consists of three major sections; the link interface, ICB interface, and the DMA interface.

18.1.1 IOA Link Interface Overview

The link interface connects the I/O adapter to the I/O module monitoring the link for commands and data bound for the I/O adapter, and arbitrating access for on board resources to send commands and data to the I/O module.

18.1.2 IOA ICB Interface Overview

The ICB interface receives commands from the link interface to generate read/write cycles on the ICB bus, and monitors the interrupt requests from the I/O cards sending interrupt request messages to the link interface.

18.1.3 IOA DMA Interface Overview

The four DMA channels are operated by the I/O cards through the DTB bus to perform block data transfers between the DTB bus and the link interface.

18.1.4 I/O Link Bus Interface

The I/O link bus interface controls I/O adaptor transfers to the I/O module. Access to the link bus by block transfer control, and the ICB interface is arbitrated by the link bus interface. The link bus interface hardware consists of registers to buffer commands from the I/O adapter to the I/O module, and from the I/O module to the I/O adapter, steering and control logic to monitor the link bus for commands and responses directed to the I/O adapter from the I/O module, and to control I/O adapter acquisition of the link bus and issuing commands to the I/O module, and pipeline, synchronization registers and bus drivers.

18.2 IOA ICB Interface

The ICB interface controls data transfers between the ICB and the link for subsequent transfer to a processor module. The ICB interface monitors the link bus for cycles addressing I/O and generates cycle(s) on the ICB. For reads from the ICB a response message is sent back on the link bus containing the data read from the I/O controller. The ICB interface hardware consists of buffers to drive the request information on the ICB, registers to store the received data for read cycles and steering and control logic to control link bus data transfers and ICB cycle timing generation.

18.2.1 ICB Error Handling

For read errors from the ICB an error or error data response is returned. For reads or writes an interrupt request is sent to the I/O module with a slot number of 19. A status register mapped into local I/O space for the I/O adaptor indicates whether the error was a ICB bus timeout, or ICB bus error. An ICB error latch mapped into local I/O space stores the ICB address that generated the error, the link cycle type and the system bus slot that requested the operation in error. An error overrun bit is set if subsequent requests are issued to the ICB interface from the I/O module. The cycles requested will be performed, but information on any subsequent error is lost.

18.3 IOA DMA Channels

The DMA channels control block data transfers between the DTB and the link bus for subsequent transfer to system bus memory modules. The four major subsections of the DMA channels are DTB interface, block transfer control, and FIFO data buffer.

18.3.1 DTB Interface

The DTB interface controls I/O controller data transfers to the I/O adaptor. Access to the DTB by the I/O controllers and the block transfer control is arbitrated by the DTB

interface. The DTB interface hardware consists of two sets of register files and latches one for inbound and one for outbound DTB data, drivers and receivers, channel control logic that maintains state information for each channel, and steering and control logic to direct traffic from the DTB to block transfer control and the FIFO buffer and to control DTB data and control transfers.

18.3.2 Block Transfer Control

Block transfer control sequences block data transfers providing address generation, transfer length control, and channel data transfer control between the FIFO and the DTB and link busses. Block transfer control hardware consists of register file and update paths for system bus address, DTB and link transfer counts, and steering and control logic for channel transfer control.

18.3.3 FIFO Data Buffer

Because of the intermittent nature of instantaneous transfer rates on the system bus, and the lack of buffering on the A1000 EDT controller, the link bus interface and the DTB interface transfer data into a first-in-first-out (FIFO) data buffer for each DMA channel. FIFO data buffer hardware consists of an 8K by 32 bit RAM array which is divided into 4 partitions for the DMA channels, 2 register files and update paths to maintain pointers and counts for DMA FIFO channels.

18.3.4 DMA Channel Setup

A DMA channel is programmed by an I/O controller to perform a block transfer into physical address space. The I/O controller sends two 32 bit words on the DTB to the DMA channel specifying the following:

1. Bits 00-31 of start address of data transfer in physical address space
2. transfer length, write, controller DTB ID, system bus memory module slot address

18.3.5 DMA Channel Error Handling

When an error is encountered by a DMA channel as a result of a main memory uncorrectable error or a I/O module error on read transfer the current block transfer for that channel is aborted, and an interrupt is sent to the I/O controller over the DTB. The controller determines the error reason from the DMA status word. In the event of a NACK from the I/O module all DMA channels are reset. The I/O controller must provide a time out on DMA transfers and reset the DMA channel, in the event of a I/O module, or memory module failure to respond to a command from a DMA channel.

18.3.6 DMA Channel Reset

A DMA channel is reset by a I/O adapter board reset, or a channel reset from an I/O controller. The channel active status, its FIFO channel and channel data registers are cleared.

19. Glossary

- CBA

  CSS Bus Arbiter

- Centronics

  An industry standard parallel printer interface

- CSS

  Computational Subsystem

- DP/IF

  Dual Port interface board, an interface to 4 SMD devices and a QIC interface

- DTB

  Data Transfer Bus (an Arete 1000 I/O bus)

- EDAC

  Error Detection and Correction

- EDT

  Enhanced Disk/Tape controller

- EGC

  Enhanced General Purpose Communications controller

- ESMD

  Enhanced SMD, a performance enhanced version of SMD capable of 2.4 Mb per Second transfer rates.

- FPU

  Floating Point Unit

- FRU

  Field Replaceable Unit

- GB

  Gigabyte (2**30 bytes)

- GC

  General Purpose Communications controller

- GC/IF

  General Purpose Communication interface board, an interface to 8 serial ports and 1 parallel port.

- HSDT

  High Speed Disk Tape controller

- ICB

  Interprocessor Communications Bus (an A1000 I/O bus)

- IEC

  International Electrotechnical Commission (an international safety agency)

- I/O

  Input/Output

- IOA

  I/O Adaptor

- IOCP

  Input/Output Control Processor

- IOL

  I/O Link, an internal System 3000 I/O bus

- IOM

  I/O Module

- IOSS

  Input Output Subsystem

- MAC

  Multibus Adaptor Card, an Arete 1000 I/O card that adapts the ICB to the IEEE-796 bus.

- MB

Megabyte (2**20 bytes)

- MHz

MegaHertz (million cycles per second)

- MMU

Memory Management Unit

- Multibus

An industry standard 16 bit microcomputer bus (also referred to as IEEE-796)

- 9T/IF Nine track interface board, an interface to 2 SMD devices and a Pertec interface.
- nS

nanosecond (one billionth of a second)

- Pertec

An industry standard interface to half inch 9 track tape devices

- QIC

An industry standard interface to quarter inch tape devices

- SMD

Storage Module Device, an industry standard interface to fixed disk magnetic storage devices

- TUV

German safety and emissions testing organization

- UL

Underwriters Labs (safety agency)

- uS

microsecond (one millionth of a second)

- VDE

German safety and emissions standards organization