DataGeneral

ECLIPSE® S/120
Computer System Hardware Reference

This document contains information describing both Class A computing devices which are compliant with FCC radio frequency interference Rules, and Class A computing devices which are not compliant with FCC Rules. Therefore, suitable warnings to cover both compliant and noncompliant devices are provided below.

Warning: For devices compliant with FCC Rules

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. It has been tested and found to comply with the limits for Class A computing devices pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Warning: For devices that do not comply with FCC Rules

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. As temporarily permitted by regulation it has not been tested for compliance with the limits for Class A computing devices pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Since October 1, 1983, all computing devices, both Class A and B, must have permanently attached in a conspicuous area, an FCC compliant label. The only exceptions to this rule are those devices covered by a grandfather clause or a waiver; these must bear the FCC noncompliant label. If a unit does not have either label, it is an indication that it was manufactured before October 1, 1983, and is probably noncompliant.

ECLIPSE® S/120 Computer Systems
Hardware Reference

DataGeneral

Notice

Ordering No. 014-000690

Printed in the United States of America
Rev. 03, April 1985

Preface

This book is intended to serve the needs of several categories of readers: engineers who want to incorporate ECLIPSE® S/120 equipment into their own systems; system operators who want to configure their own specially tailored systems; and field service engineers who want to know how to troubleshoot equipment.

This manual covers both FCC compliant systems, and those that were manufactured and sold before FCC compliance was mandatory. The devices detailed first in each case are compliant.

The book covers the system processing unit with RAM memory, as well as the chassis and power supply. A separate chapter on the virtual console is also included.

In this manual, central processing unit (CPU) refers to the microECLIPSE™ ME674 16-bit CPU integrated circuit and three external microcode controller (XMC) chips. The system processing unit (SPU) refers to the CPU, a multidevice chip, memory allocation and protection unit, error checking and correction unit, system read/write memory, and virtual console, all of which reside on one printed circuit board.

Manual Organization

This manual consists of an overview, seven chapters, four appendices, and an index. Following the index are several forms and lists to help you contact DGC for information and services.

The "System Overview" lists and briefly describes the ECLIPSE S/120 SPU and its components. It also provides lists of standard ECLIPSE S/120 model numbers and specifications.

Chapter 1, "System Processing Units," expands the description of components begun in the overview and summarizes the capabilities of the ECLIPSE S/120 and its instruction set. The chapter concludes with discussions of installation, jumpering, and interfacing.

Chapter 2, "Theory of Operation," begins with discussions of the ECLIPSE S/120 system and timing and provides an overview of the SPU. Two topics related to the CPU follow: "The CPU Section" and "CPU Support Elements." Discussions of system memory and the NOVA/ECLIPSE input/output interface end the chapter.

Chapters 3 and 4 are entitled "16-Slot Chassis" and "5-Slot Chassis," respectively. Each includes information about chassis architecture, backpanel jumpering, the front console switches and indicators, and cabling.

Chapters 5 and 6 describe the "16-Slot Power Supply" and "5-Slot Power Supply." Each provides a functional overview, discusses operating theory, and describes system interfacing.

Chapter 7, "Virtual Console," describes a program that allows the user to interact with the ECLIPSE S/120 system. It explains the use of all the commands that facilitate this interaction and contains sections on command format and errors.

Appendix A presents a "Summary of NOVA/ECLIPSE I/O Bus Signals."

Appendix B contains "Standard I/O Device Codes."

Appendix C pinpoints the "Locations of Microcode on the microECLIPSE Chip Set."

Appendix D lists SPU external signals.

The Index alphabetically lists the major concepts and terms in the manual.

Several lists and forms follows the index.

"DG Offices" lists all Data General facilities world-wide.

"How to Order Technical Publications" provides the addresses and telephone numbers of agencies from which order forms and manuals can be obtained.

"Technical Publications Comment Form" invites you to assist DGC in improving future publications by evaluating this manual.

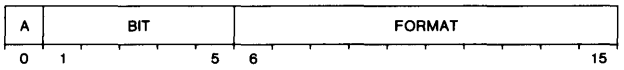
"Users' Group Membership Form" brings DGC software users together, in group meetings and through various publications, to exchange ideas, applications, problems, and solutions.

Typesetting Conventions

Throughout this manual we use the following conventions to show instruction formats.

COMMAND	Bold face and uppercase letters indicate the instruction mnemonics. You code them into your program exactly as they appear. This typeface is also used to indicate user input in examples of the use of commands.
argument	Italics and lower case letters indicate that a particular instruction takes an argument. In your program, you must replace this symbol with the exact code for the argument you need.
[optional]	Brackets denote an optional argument. (Command switches appear in this format as well.) If you decide to use this argument or switch, do not write the brackets into your code; they only set off the choice.

The diagram below shows the arrangement of 16 bits in an instruction. The diagram is always divided into 16 boxes, numbered 0 to 15.



Mnemonics for signal names appear in bold face capital letters. When the signal is active low, it is barred, e.g., $\overline{\textbf{READY}}$.

Related Manuals

- ECLIPSE® *S/120 Assembly Language Programmer's Reference* (DGC No. 014-000686) is the main source of information for the assembly language programmer. It describes the instruction set of the ECLIPSE S/120 computer in detail.
- ECLIPSE® *S/120 Diagnostic and Maintenance Manual* (DGC No. 015-000116) contains a description of all controls, fault indicators, and testpoints; descriptions of power-up and power-down procedures; installation guidelines; references to all available diagnostic routines; and a description of spares and replacement procedures.
- *Introduction to Real Time* ECLIPSE *Computers* (DGC No. 014-000687) describes in summary form the entire line of DG ECLIPSE hardware, both processors and peripherals. It also contains a complete bibliography of DG ECLIPSE documentation and references to introductory software books.
- *Interface Designer's Manual for* NOVA *and* ECLIPSE *Line Computers* (DGC No. 014-000629) is written for those who are interested in designing a custom interface that will operate with the NOVA/ECLIPSE I/O bus. It describes the NOVA/ECLIPSE I/O bus architecture, including the standard data channel facility; connections, connectors, and terminators; and interface boards.

Table of Contents

Preface

Manual Organization **i**
Typesetting Conventions **ii**
Related Manuals **ii**

System Overview

Chapter 1
System Processing Unit

Major Elements **1-1**
CPU and XMCs **1-1**
System Input/Output Controller **1-2**
System Memory **1-3**
Memory Allocation and Protection **1-3**
Error Checking and Correction **1-3**
Virtual Console **1-3**
Power Monitoring and Initialization **1-3**
NOVA/ECLIPSE Input/Output Interface **1-5**
Instruction Set **1-5**
Addressing **1-5**
Data Formats **1-6**
Operations **1-6**
Summary of Instructions **1-10**
Reserved Memory Locations **1-18**
Instruction Execution Times **1-18**
Power-up Response **1-21**
Power Fail/Autorestart **1-21**
Installation and Jumpering **1-23**
Power Requirements **1-23**
Jumpering **1-23**
Asynchronous Interface Device Cables **1-28**
Interfacing: A & B Connectors **1-28**

Chapter 2
Theory of Operation

S/120 System Architecture **2-1**
System Timing **2-1**
SPU Overview **2-2**
The CPU Section **2-3**
mE674 CPU **2-3**
Microcode Controller Chips **2-9**
Timing **2-9**
Control **2-13**
System Memory Read/Write Operations **2-15**
Extended Memory Cycles **2-15**
CPU Support Elements **2-18**
The mE676 SIO Chip **2-18**
SIO Configuration Jumpers **2-21**
Address Latch **2-22**
I/O Decoder **2-22**
Virtual Console **2-22**
Automatic Program Load **2-22**
Error Checking and Correction Unit **2-26**
Memory Allocation and Protection **2-29**
System Memory **2-33**
Initiating a Memory Operation **2-33**
System Memory Operations **2-36**
Refresh and Correction Cycle **2-44**
Critical Voltage Monitor **2-44**
NOVA/ECLIPSE Input/Output Interface **2-48**
Bus Interface **2-48**
State Machine **2-49**
Bus Control Logic **2-55**
Data-Channel Address Latch **2-55**
Data-Channel Data Latch **2-55**

Chapter 3
16-Slot Chassis

Chassis Architecture 3-1
Backpanel Jumpering 3-6
Front Console 3-6
Switches 3-6
Indicators 3-6
Additional Indicators 3-7
Cabling 3-7
Line Cord 3-7
System Console Cable 3-7
I/O Cables 3-8
Diagnostic Test Plug 3-8

Chapter 4
5-Slot Chassis

Chassis Architecture 4-1
Backpanel Jumpering 4-4
Front Console 4-4
Switches 4-5
Indicators 4-5
Cabling 4-5
Line Cord 4-5
System Console Cable 4-5
I/O Cables 4-5

Chapter 5
16-Slot Power Supply

Functional Overview 5-1
Theory of Operation 5-3
Compliant VNR Unit 5-3
Noncompliant VNR Unit 5-4
Power Supply Board 5-5
Battery Backup 5-7
Interconnection with the System 5-8

Chapter 6
5-Slot Power Supply

Compliant 5-slot Power Supply 6-1
Functional Overview 6-4
Main Power Supply 6-4
Battery-backed Supply 6-7
Fault Detection 6-8
System Clocks 6-8
Status Circuits 6-9
Test Points 6-10
Fault Indications 6-10
Interconnections 6-11
Noncompliant 5-Slot Power Supply 6-12
Functional Overview 6-12
Theory of Operation 6-12
Battery Backup 6-15
Interconnection with the System 6-15

Chapter 7
Virtual Console

Entering the Virtual Console 7-1
Correcting Entry Errors 7-2
Rubout/Delete 7-2
The K Command 7-2
Virtual Console Errors 7-2
Commands 7-2
Cell Commands 7-2
Function Commands 7-4
Miscellaneous Commands 7-6

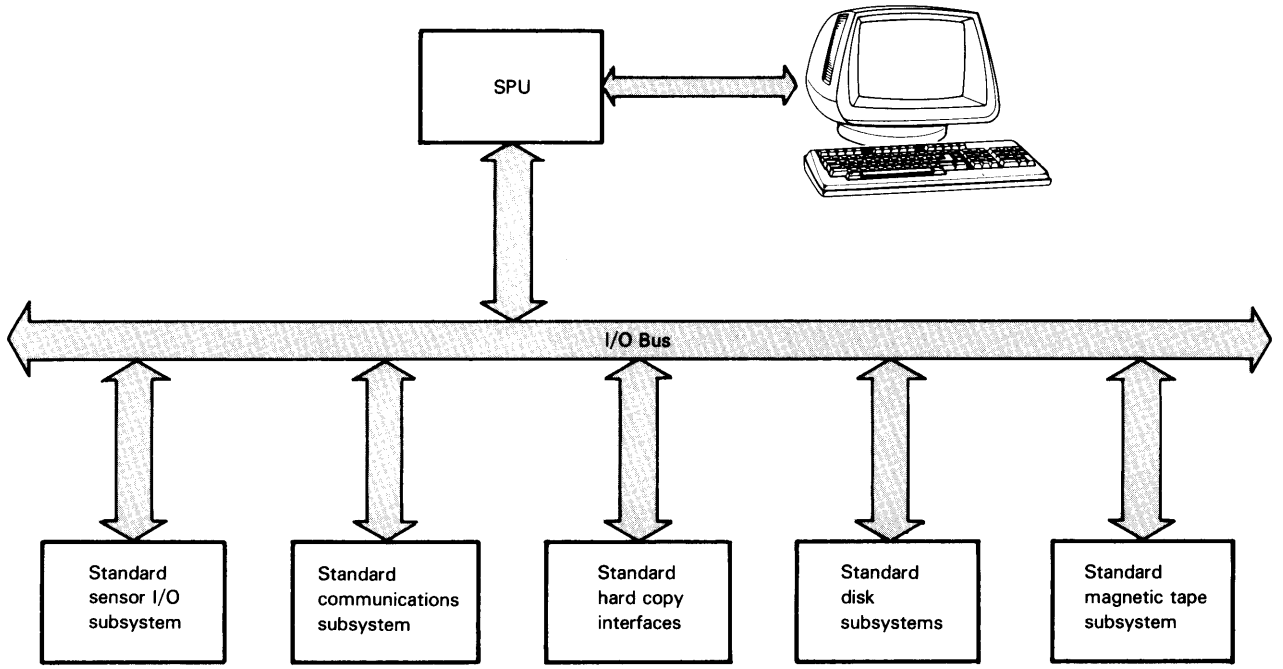
Appendix A
Summary of NOVA/ECLIPSE I/O Bus Signals

Summary of I/O Bus Signals A-4
Data A-4
Programmed I/O A-4
Program Interrupt A-5
Data Channel A-5
System Control A-15

Appendix B
Standard I/O Device Codes

Appendix C
Locations of Microcode on the microECLIPSE® Chip Set

Appendix D
SPU External Signals



DG-09035

A typical ECLIPSE S/120 computer system

System Overview

An ECLIPSE S/120 system processor unit (SPU) combines several computer elements on a single printed circuit board (PCB). These elements include

- A microprogrammed central processing unit (CPU)
- Dynamic random-access memory (RAM) of 128, 256, or 512 kilobytes
- Memory error detection and correction unit
- Real time clock
- Programmable interval timer
- Asynchronous console interface
- Input/output interface
- Virtual console
- Power status monitoring

Combined with standard input/output (I/O) controllers, the SPU PCB provides the foundation for a powerful high-speed minicomputer system.

The SPU PCB measures 15 in. by 15 in. and is intended for use in a standard DGC chassis (PCB cage) with power supply.

The adjacent figure shows a typical ECLIPSE S/120 system. The following tables summarize the standard S/120 computer system's features and model numbers, respectively.

Model	Description
8731-K	Standard 5-slot chassis, including power supply with battery backup, and SPU with 128 kbytes of MOS RAM.
8731-N	Standard 5-slot chassis, including power supply with battery backup, and SPU with 256 kbytes of MOS RAM.
8731-R	Standard 5-slot chassis, including power supply with battery backup, and SPU with 512 kbytes of MOS RAM.
8732-K	Standard 16-slot chassis, including power supply with battery backup, and SPU with 128 kbytes of MOS RAM.
8732-N	Standard 16-slot chassis, including power supply with battery backup, and SPU with 256 kbytes of MOS RAM.
8732-R	Standard 16-slot chassis, including power supply with battery backup, and SPU with 512 kbytes of MOS RAM.

Standard ECLIPSE S/120 model numbers

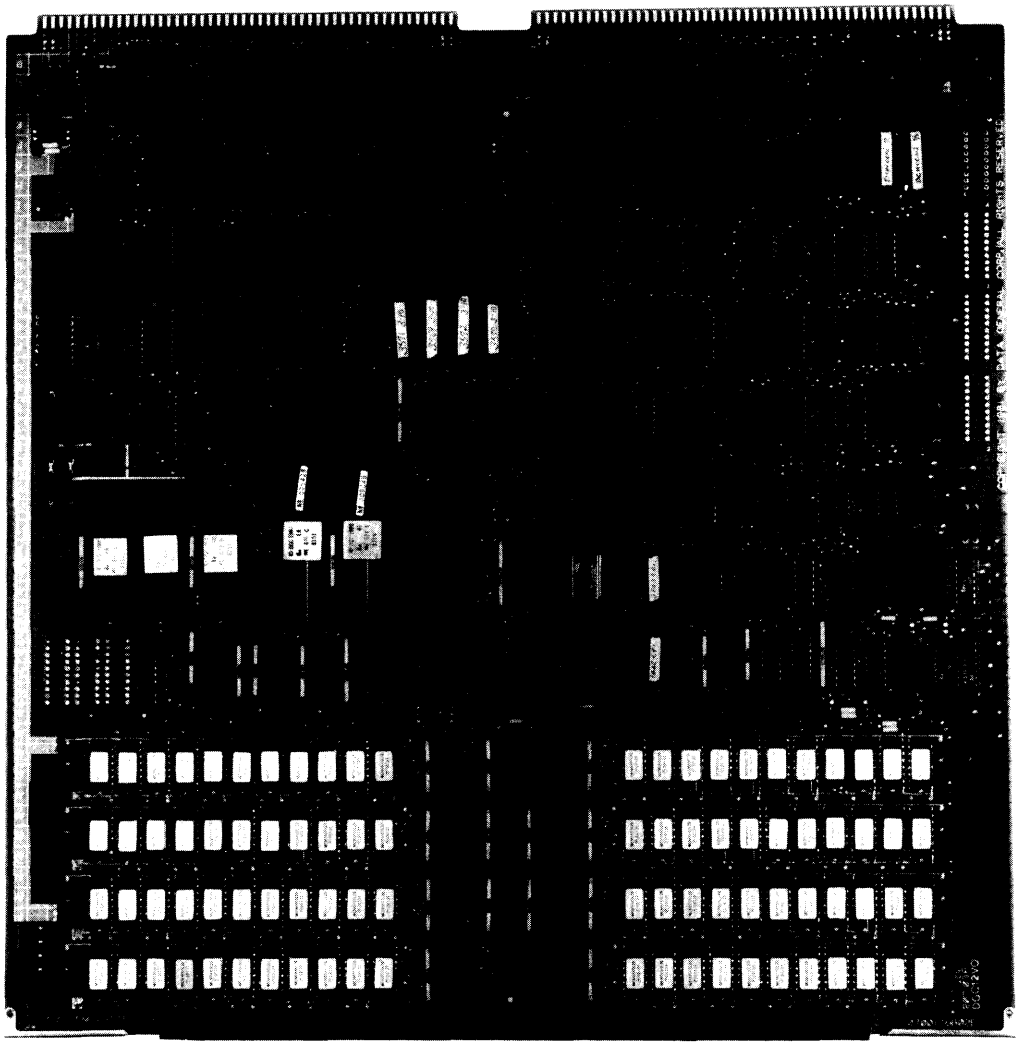
Feature	Description
Central Processor	
Type	mE674 microprocessor
Word length	16 bits
Instruction length	16 bits, 32 bits
Number of accumulators	
Fixed point	4 (2 may be used as index registers.)
Floating point	4
Accumulator length	
Fixed point	16 bits
Floating point	64 bits
Hardware stack facility	Overflow protected; stack pointer, frame pointer, stack fault address, and stack limit control
Priority interrupt levels	16
Cycle time	500 nanoseconds
Addressing modes	Absolute, relative, indexed, and deferred
Data types	Bits, bytes, words, floating-point numbers, blocks, and character strings
Data channel rates (max)	
Input	2.0 megabytes/second
Output	1.3 megabytes/second
Latency	30 microseconds
Address translation	Standard ECLIPSE map
User maps	4
Data channel maps	4
Protection	Validity, write, indirect, and I/O
Memory	
Type	Semiconductor (dynamic MOS)
Capacity	128, 256, or 512 kilobytes
Word length	16 data bits plus 6-bit error code
Error detection/correction	Double-bit and most triple-bit errors detected; single-bit errors corrected
Cycle time	500 nanoseconds
Powerfail protection	Battery backup

Features of an ECLIPSE S/120

Feature	Description
Power Supply	
5-slot (compliant)	
Line voltage	90 - 132 V ac (187 - 264 V ac)
Line frequency	47 - 63 Hz
Line current draw	8 A (4 A)
Current delivered	
+5V dc	40 A
−5 V dc	2.0 A
+12V dc	5 A
−12V dc	0.025 A
+5V MEM dc	5 A
+12V MEM dc	2.3 A
−5V MEM dc	0.05 A
Current delivered during Battery Backup	
+5V MEM dc	5.0 A
+12V MEM dc	0.3 A
−5V MEM dc	.05 A
Battery Backup duration using DG batteries fully charged	
Full load	10 minutes
Half load	30 minutes
DG Battery	118-1818
5-slot (noncompliant)	
Line voltage	85 - 132 V ac (187 - 264 V ac)
Line frequency	47 - 63 Hz
Line current draw	6 A (3.5 A)
Current delivered	
+5V dc	35 A
−5V dc	1.5 A
+12V dc	5 A
+15V dc	5 A
−11V dc	0.025 A
+5V dc MEM	1 A
+12V dc MEM	3 A
−5V dc MEM	0.05 A
16-slot*	
Line voltage	102 - 132 V ac (187 - 264 V ac)
Line frequency	47 - 63 Hz
Line current draw	12 A (7 A)
Current delivered	
+5V dc	100 A
−5V dc	3 A
+12V dc	12.5 A
+15V dc	1.5 A
−11V dc	0.02 A
+5V dc MEM	9.5 A
+12V dc MEM	6 A
−5V dc MEM	0.3 A

Features of an ECLIPSE S/120

**Power requirements are the same in both compliant and noncompliant devices.*



PH-0578

Figure 1.1 ECLIPSE S/120 SPU board

Chapter 1

System Processing Unit

Designed as the basic module in Data General's ECLIPSE S/120 computer system, the S/120 system processing unit (SPU), shown in Figure 1.1, consists of the following elements:

- Central processing unit (CPU) based on
  - a microECLIPSE mE674 16-bit CPU integrated circuit (IC)
  - three external microcode controller chips (XMCs)
- System input/output controller based on a microECLIPSE mE676 16-bit SIO integrated circuit (IC). The controller generates NOVA/ECLIPSE input/output bus timing and protocol signals, and it also contains
  - a real time clock
  - programmable interval timer
  - power monitor
  - full-duplex asynchronous communications interface
  - CPU status register
- System memory providing 128, 256, or 512 Kbytes of read-write memory
- Memory allocation and protection (MAP) unit
- Memory error checking and correction unit
- Virtual console, residing in 2 Kbytes of read-only memory (ROM) with 512 bytes of static read-write memory (RAM)
- NOVA/ECLIPSE input/output interface

The SPU receives power and communicates with other printed circuit boards in the card cage/chassis via its A and B connectors, which plug into the chassis backpanel. It communicates with a full-duplex serial, asynchronous device via its A connector and a device cable which slips over designated pins located on the rear of the backpanel.

Major Elements

This section describes the functional characteristics of each major element on the S/120 SPU board.

CPU and XMCs

The S/120 CPU is a microprogrammed processor that incorporates the full ECLIPSE 16-bit architecture, including four 16-bit and four 64-bit (floating-point) accumulators, a floating-point status register, and a MAP status register. It executes a standard ECLIPSE instruction set. A kernel of the microinstructions for the ECLIPSE instruction set is executed by the mE674 CPU IC; the remainder of the microinstructions are executed by the XMCs. The XMCs communicate with the mE674 through a dedicated 8-bit time-multiplexed bus.

An S/120 SPU board contains three XMCs. One holds microcode for ECLIPSE instructions that are not resident on the mE674, such as character, decimal, bit, byte instructions, as well as microcode for several floating-point instructions. The other XMCs hold microcode for all remaining floating-point instructions.

The mE674 CPU uses a 16-bit wide memory address/data bus and various buffered busses to communicate with its support elements which include

- system input/output controller
- system memory
- virtual console
- MAP unit
- error checking and correction unit
- input/output interface

During operations in memory mapped mode, the memory address width expands to 20 bits, 18 of which are used in the maximum-sized S/120 system (512 Kbytes). A NOVA/ECLIPSE input/output bus interface, connected to three of the buffered busses, enables the CPU to communicate with standard NOVA/ECLIPSE I/O device controllers.

Memory access time for the S/120 is 500 nanoseconds. Instruction execution times range from 500 nanoseconds to 21.5 microseconds for fixed-point operations. Floating-point operations controlled by the floating-point XMCs require 1.0 to 1144.0 microseconds. A complete listing of nominal instruction execution times for the S/120 CPU appears later in this chapter. Maximum interrupt latency is 110 microseconds (in the absence of any data channel operations).

The S/120 can contain up to 512 Kbytes of memory, with validity, I/O and write protection, and error checking and correction. In addition, the CPU supports

Direct memory access (DMA) via data channel
Two distinct program interrupt facilities
16 levels of programmed interrupt priorities

The CPU's data channel facility allows devices to transfer data to and from system memory over the ECLIPSE I/O bus at speeds of 1.3 megabytes per second for output (memory to device) and 2.0 megabytes per second for input (device to memory).

The CPU has two types of interrupt facilities: *maskable* (can be disabled); *nonmaskable* (cannot be disabled). The maskable interrupt facility services interrupt requests based on the following order of priorities:

(1) power fail
(2) stack overflow
(3) programmed interval timer
(4) real time clock
(5) TTI (SPU asynchronous interface receiver)
(6) TTO (transmitter)
(7) error checking and correction
(8) external I/O

The SPU reserves the nonmaskable interrupt facility for user entry into the virtual console.

Sixteen levels of programmed interrupt priority are associated with the maskable interrupt facility. They allow the program to establish interrupt priorities among I/O interfaces. A special vectored interrupt instruction updates the priority mask while saving return information and transferring control.

System Input/Output Controller

The system input/output controller is an mE676 SIO integrated circuit designed to support the mE674 CPU. It contains three internal I/O devices plus a power monitor and CPU status register. The three internal I/O devices—an asynchronous line controller, a real time clock, and a programmable interval timer—are programmed as external I/O devices using standard I/O format instructions. The SIO chip also provides the timing and protocol signals that allow the microECLIPSE CPU to communicate with the NOVA/ECLIPSE I/O bus.

Asynchronous Line Controller

The asynchronous line controller (ALC) is the programmed input/output interface with the primary console of the S/120 system. It can transmit and receive serial asynchronous information at jumper selectable rates from 50 to 38,400 baud (refer to Table 1.25), and can use either a 20-mA current loop or an EIA RS-232-C communications line.

NOTE: *The S/120 asychronous communications interface receives and transmits eight-bit data characters without parity. If the system console device being used with the S/120 operates with a data character length of seven bits, you should configure the device to operate with mark parity. If the system console operates with a data character length of eight bits, you should configure the device to operate with no parity. When receiving data characters from a seven bit system console device, software should mask out the parity bit after the character has been loaded into an accumulator. The parity bit is the most significant bit of the character and is contained in bit 8 of the specified accumulator.*

Real Time Clock

The real time clock generates low-frequency I/O interrupts for performing time calculations independent of CPU timing. These interrupts may be used in programs requiring a time base. The frequency of the clock is program selectable to one of four frequencies: ac line frequency, 10 Hz, 100 Hz, or 1 KHz.

Programmable Interval Timer

The programmable interval timer (PIT) is a CPU-independent time base that can be programmed to initiate program interrupts at fixed intervals ranging from 1 microsecond to 65.536 seconds in jumper selectable increments ranging from 1 microsecond to 1 millisecond. (Refer to Table 1.26.) The contents of the PIT counter can be sampled with I/O instructions at any point in its cycle to determine the time remaining until the next interrupt, or following an interrupt to determine interrupt latency.

Power Monitor

The SIO chip monitors the state of a power status signal supplied by the power supply and initiates a CPU interrupt request whenever this signal changes. When the CPU acknowledges the interrupt, the SIO controller returns device code 0 to the CPU.

CPU Status Register

An 8-bit CPU status register within the SIO chip reports on the following conditions:

Powerfail interrupt
Interrupts enabled
System console BREAK key interrupt
Power-up condition
HALT instruction decoding
State of Halt Dispatch bit in the SIO configuration register
Interrupt request

Additional status bits combine with the SIO 8-bit CPU status register to form a full CPU status word. Applied externally to the CPU address/data bus, these bits record the following conditions:

- Depression of program load (PL) switch
- Forced nonmaskable interrupt in virtual console operation
- Validity of system memory data following a power disruption
- Capacity of implemented system memory

Figure 1.2 and Table 2.1 show CPU status register bit positions.

System Memory

The ECLIPSE S/120 system memory provides the processor with 128, 256, or 512 Kbytes of dynamic MOS random-access memory (RAM). Two data bytes (one 16-bit data word) reside in each addressable memory location. Data transfers to memory in either bytes or words; however, system memory always writes a word. Memory operations specifying one byte to be written, first read the entire word from the addressed memory location. Then the data byte to be written replaces the specified byte, and the entire word is written. Data transfers from memory are always word transfers.

Memory Allocation and Protection

The S/120 maximum memory size of 512 Kbytes is fully supported by the memory allocation and protection (MAP) feature. MAP performs logical-to-physical address translation, allowing access to up to 2 Mbytes of physical memory. The MAP unit stores up to four user- and four data-channel address translation tables, or *address maps*. In addition to translating addresses, the MAP feature performs the following functions:

- Validity protection
- Write protection
- I/O protection
- Indirection protection

The MAP feature also allows the implementation of the *Load Effective Address* instruction and the *emulator trap* feature. These and the protection features listed above are discussed in more detail in the SPU "Instruction Set" section later in this chapter.

Error Checking and Correction

The S/120 error checking and correction (ERCC) unit generates and appends a 6-bit checkcode to each word (2 bytes) of data written to memory. During memory-read operations, the ERCC processes the 22-bit word from memory to determine if an error has occured. When a single-bit error is detected, the erronous bit is corrected before the word is transferred, and the corrected word is also written to memory with its new checkcode. In addition, the fault address and an error identification code are saved for transfer to the CPU, and if ERCC interrupt mode is enabled, a CPU interrupt is requested. This fault address and error identification code can be used to identify a marginal or failing system memory RAM chip.

Double-bit errors and some triple-bit errors are detected and logged but not corrected. However, if the ERCC interrupt mode is enabled, a CPU interrupt is requested.

The S/120 also implements an advanced error checking and correction feature (*sniffing*) that continuously tests all on-board memory at the rate of one location per 16 microseconds when enabled. This results in a complete check of the S/120 memory every four seconds and minimizes the accumulation of correctable single-bit errors into multiple bit errors.

Virtual Console

The *virtual console*, resident firmware with 512 bytes of read/write memory, allows the programmer or operator who is using the terminal connected to the SPU's asynchronous interface to inspect and modify the system's state. Additionally, it aids program debugging.

The virtual console allows the user to

- Stop, start, single step, and continue program execution
- Examine and alter CPU registers and memory locations, including MAP contents
- Initiate program load sequences
- Initiate system self tests
- Perform an I/O reset

For more information, refer to Chapter 7, "Virtual Console."

Power Monitoring and Initialization

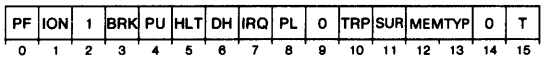
The power supply generates two status signals that are monitored by the SPU. One, **PWROK**, indicates that all voltages are within specified limits; the other, **$\overline{\text{PWRFAIL}}$**, indicates that a power loss is imminent.

After the SPU is notified that all voltages are above the specified minimum limits on power up, it enters the *virtual console mode*. When this occurs, the SPU can either be started automatically or with the PR/LOAD-RESET (program load/reset) switch. (Refer to "Power-Up Response" later in this chapter.)

Name **Format**

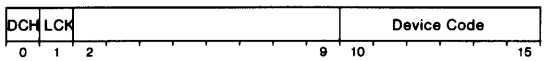
CPU Status

Read with DIS ac,CPU



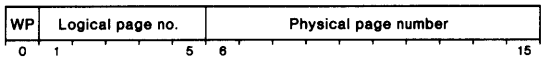
Auto Program Load

Read with DIA ac,APL



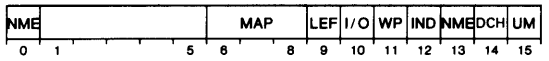
MAP Address

Write with LMP ac



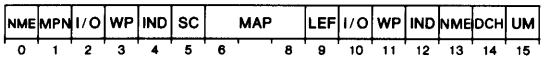
MAP Status Out

Write with DOA ac,MAP



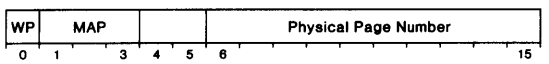
MAP Status In

Read with DIA ac,MAP



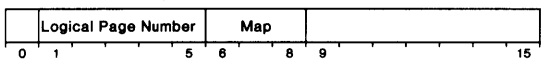
MAP Page Check

Read with DIC ac,MAP



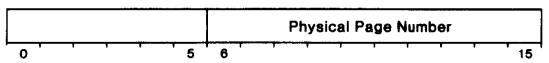
MAP Initiate Page Check

Write with DOC ac,MAP



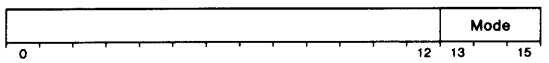
MAP Page 31

Write with DOB ac,MAP



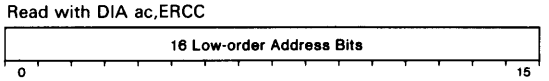
ERCC Enable

Write with DOA ac,ERCC



Memory Fault Address

Read with DIA ac,ERCC



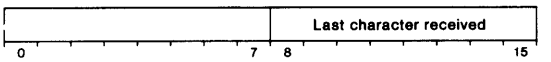
Name **Format**

Memory Fault Code

Read with DIB ac,ERCC

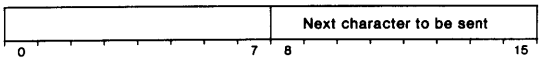
TTI Character Buffer

Read with DIA ac,TTI



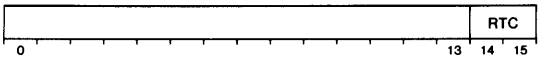
TTO Character Buffer

Write with DOA ac,TTO



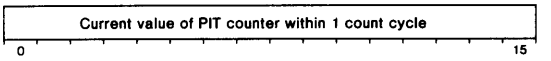
Real Time Clock

Write with DOA ac,RTC



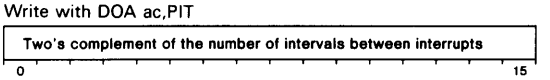
PIT Count

Read with DIA ac,PIT



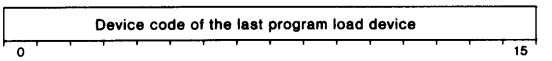
PIT Initial Count

Write with DOA ac,PIT



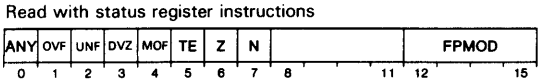
Virtual Console

Read with READS ac



Floating Point Status Register (Word 1)

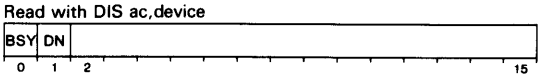
Read with status register instructions



(Word 2)

Device Status

Read with DIS ac,device



NOTE: *The format for "Device Status" applies to any I/O device, internal (e.g., ERCC, TTO) or external.*

DG-09096

Figure 1.2 Program-accessible registers

An imminent power loss generates a program interrupt. When the power loss is first detected, the SPU has approximately one to two milliseconds of operating time before any dc voltages fall below specifications. If battery backup is present and the batteries have not been exhausted during the power disruption, an automatic restart operation is initiated when power is restored and the SPU is locked.

NOVA/ECLIPSE Input/Output Interface

This interface performs the control, timing and buffering activities necessary to pass instructions and data from the CPU to its input/output devices. The interface performs both programmed input/output and data channel transfers.

Instruction Set

This section summarizes the instruction set of the ECLIPSE S/120 SPU. It includes a figure summarizing program-accessible registers (Figure 1.2) and brief descriptions of important programming features, such as addressing modes, data formats, and operators. Summaries of instructions, reserved memory locations, and instruction execution times complete the section.

This section is not a programming reference; rather, it is a guide to the capabilities of the S/120 SPU. For complete information on programming the S/120 SPU, refer to the ECLIPSE *S/120 Assembly Language Programmer's Reference* (DGC No. 014-000686).

Addressing

The S/120 has a logical system memory address space of 64 Kbytes and a physical system memory address space that can be as large as 512 Kbytes. "MAP Operations" later in this chapter summarizes logical-to-physical address translation.

The S/120 computer has two classes of instructions. *Short class* instructions contain an 8-bit address displacement. *Extended class* instructions contain a 15-bit address displacement.

Both classes use one bit to specify either direct or indirect addressing. In addition, indirect addressing can be specified by a bit within the contents of an address. (If bit 0 of an addressed word is 1, the addressed word is used as a pointer to another address.)

Any number of indirection levels is permitted in the S/120, except in mapped mode which can limit indirections to 15 levels.

Addressing Modes

Direct or indirect word-addressing in the S/120 can be done in a number of modes:

Absolute	The address before indirection is the unmodified displacement, that is, the page 0 address.
Program-Counter Relative	The address before indirection is the total of the displacement plus the address of the word containing the displacement, that is, the current instruction.
Accumulator Relative	The address before indirection is the total of the displacement plus the contents of a specified accumulator (AC2 or AC3).

Figure 1.3 illustrates the accessible memory ranges for the two instruction classes and three addressing modes (direct addressing). Note that any memory-reference instruction can access lower page zero (locations 0-377$_8$).

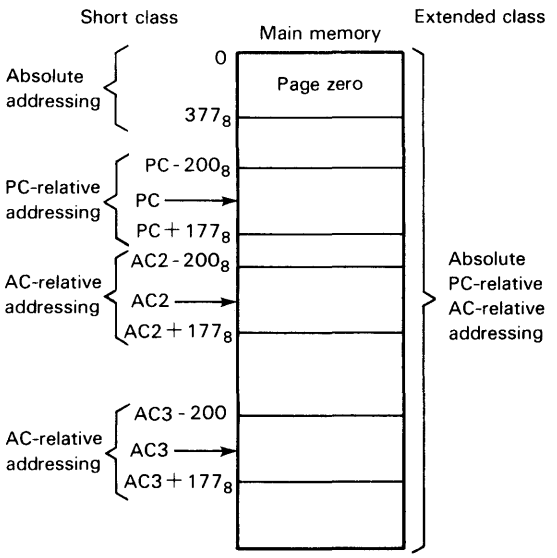


Figure 1.3 Addressing modes

Byte Addressing

A 16-bit *byte pointer* selects a byte in memory. Bits 0-14 of the byte pointer contain the memory address of a 2-byte word. Bit 15 of the byte pointer indicates which byte of the address location is to be used. Short class instructions use an accumulator to hold the byte pointer; extended class instructions use a byte pointer contained in the displacement field of the instruction.

Bit Addressing

A 32-bit bit pointer selects a bit in memory, as illustrated in Figure 1.4. Instructions requiring a bit pointer use two accumulators (specified in the instruction) to hold the pointer.

Data Formats

This section summarizes *integer formats* (Figure 1.5) and *floating-point formats* (Figure 1.6). Floating-point numbers are normalized at the end of all floating-point mathematics operations.

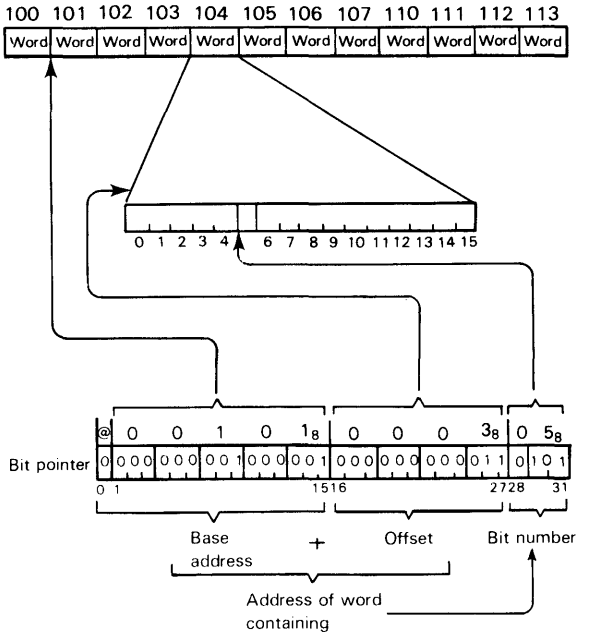


Figure 1.4 Bit pointer

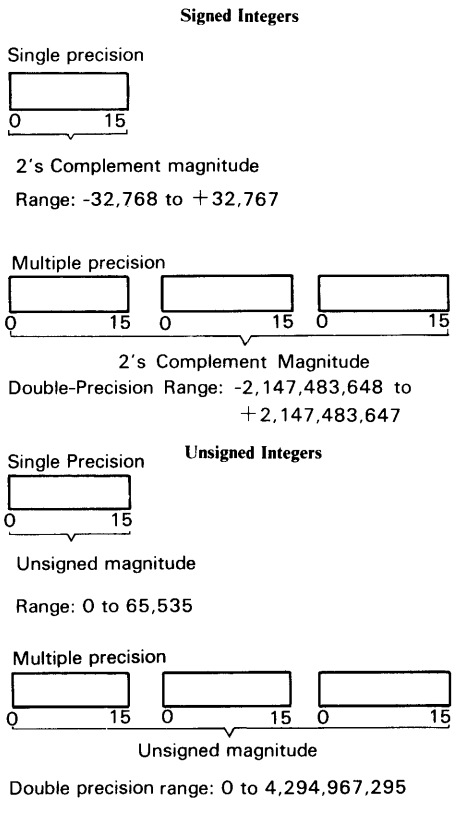
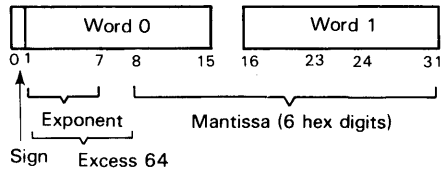


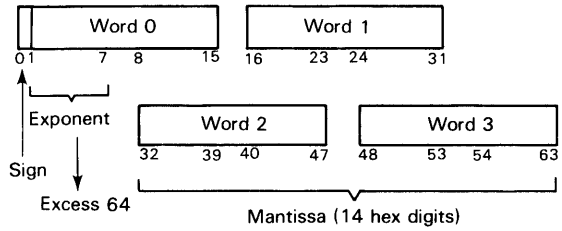
Figure 1.5 Integer formats

Single-precision (2 words)



True value of exponent = (Value in byte 0) - 64

Double precision (4 words)



True value of exponent = (Value in byte 0) - 64

Range of exponent field: 0 to 127
Range of true value of exponent: -64 to 63

Magnitude of floating-point number:
Mantissa x 16^{4} (true value of exponent)
Normalization: Shift mantissa left 1 hex digit and decrement exponent — repeat until high-order hex digit ≠ 0.

DG-08292

Figure 1.6 Floating-point formats

Operations

This section organizes the S/120 instructions into categories, according to the operations they perform. The operations are

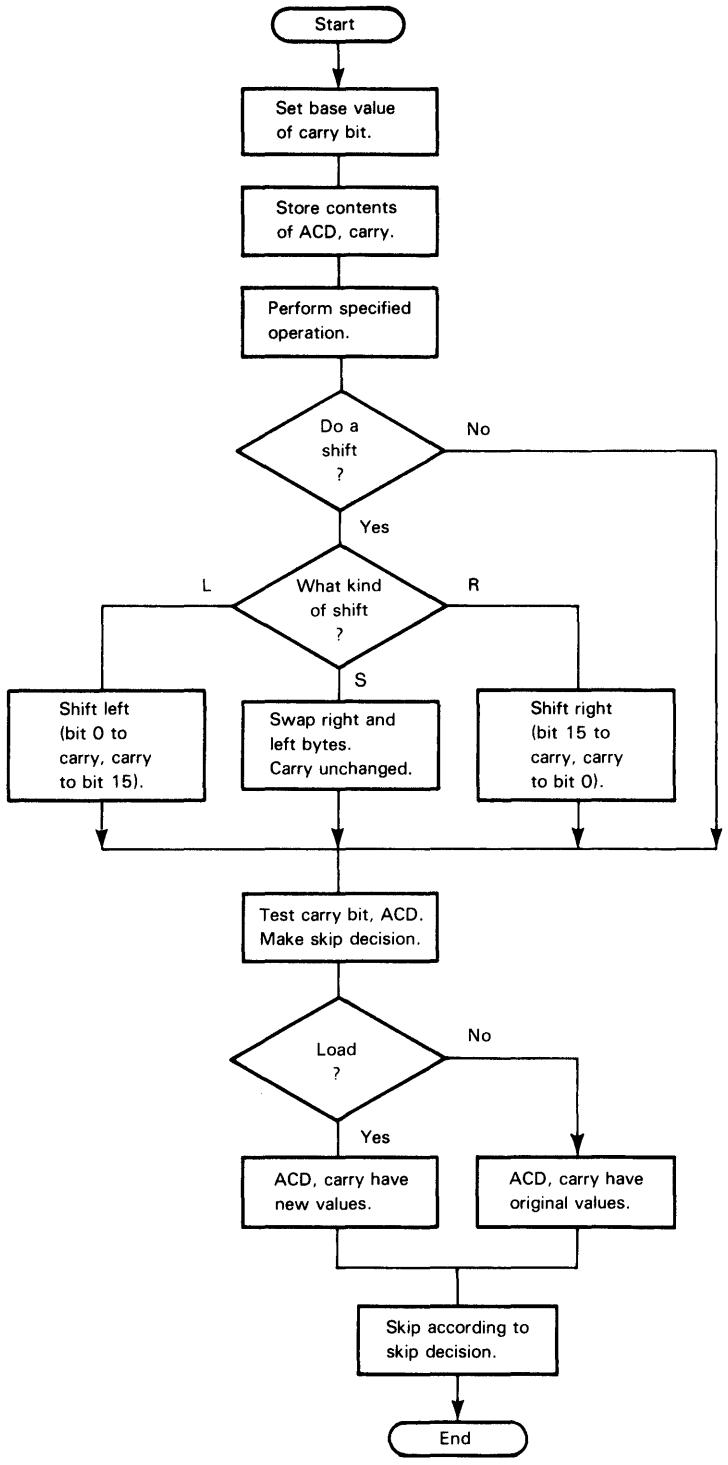
- Arithmetic/Logic class
- Stack
- Floating-point
- String
- MAP
- Extended
- Emulator trap
- Error checking and correction

Arithmetic/Logic Class Operations

Arithmetic/logic class (ALC) instructions—ADC, ADD, AND, COM, INC, MOV, NEG and SUB—perform a group of general functions in addition to the specific functions implied by their names. Encoded in four fields in the ALC instructions, these general functions are

Set carry bit (0, 1, complement, or no change)
Shift (right, left, swap)
Skip test
Load or No Load

Figure 1.7 illustrates the sequence of operations performed by a general ALC instruction.



DG-08293

Figure 1.7 ALC instruction sequences

Stack Operations

The S/120 maintains a last-in/first-out stack in main memory. Stack operations depend on the contents of four reserved lower page zero locations: the stack pointer, the frame pointer, the stack upper limit, and the stack fault routine pointer. The program must establish the initial contents of these locations. Once this is done, the CPU automatically updates the stack pointer and frame pointer, and if an instruction causes a stack overflow, it jumps to a stack fault routine (created by the programmer) using the stack fault routine pointer. A fast and efficient method of changing stacks is also provided so that a priority interrupt handler can optimally use the stack feature.

Floating-Point Operations

The floating-point instructions permit the manipulation of both single- (32-bit) and double- (64-bit) precision numbers. Single precision yields 6 to 7 significant decimal

digits; double precision yields 15 to 17. In both precisions, the decimal range of a floating-point number is approximately 5.4×10^{-79} to $7.2 \times 10^{+75}$.

Four separate 64-bit accumulators (FPACs) are available for floating-point operations. The first floating-point operand is always in an FPAC, but the second operand may be in either an FPAC or retrieved from memory. The four FPACs and their associated status bits can be pushed onto, or popped off, the stack with one instruction.

After every floating-point operation, the floating-point status register is checked for the following possible fault conditions.

Overflow	Exponent overflow occurred. The exponent should be increased by 128; otherwise, the result is correct.
Underflow	Exponent underflow occurred. This condition is analogous to an exponent overflow.
Divide by zero	Zero divisor detected; division aborted.
Mantissa overflow	A bit was shifted out of the high-order end of the mantissa by a **FSCAL** instruction, or the result of a **FFAS** or **FFMD** instruction does not fit into the destination.

A floating-point fault condition initiates a floating-point trap if the program has set the trap enabling bit (5) in the floating-point status register to 1. This trap pushes a return block and causes an indirect jump via location 45_8.

Several floating-point instructions have two forms, one ending in *S* to specify single-precision floating-point format, and one ending in *D* to specify double-precision. Otherwise, the two forms are functionally identical. Floating-point formats are listed under "Data Formats" earlier in this chapter.

String Operations

The string instructions CMP, CMT, CMV, or CTR move strings of bytes from one portion of memory to another, compare one string of bytes with another, and translate a string of bytes from one representation to another. One instruction can scan a string of bytes for one or more delimiters.

MAP Operations

Figure 1.8 illustrates the address translation performed by the S/120 MAP unit. A program can load an address translation map consisting of 32 12-bit words for each of up to four users and four data channels. (The software actually loads 11 bits - *physical page and write protect*; the twelfth bit - *validity* - is produced by the MAP hardware and is the logical AND of the ten physical page address bits and the write-protect bit being loaded.)

Each user's or data channel's logical address space consists of 32 1024-word (2 kbyte) pages. Ten of the 12 bits in the MAP register specify the physical page to which a logical page is mapped; one bit specifies whether that page is write protected; and one bit specifies whether the page is validity protected.

The translation process occurs every time a memory access is made, provided the program has enabled the MAP by manipulating the contents of the MAP status register. A MAP fault occurs when the program attempts to access a validity-protected word or to write to a write-protected word. In either case, the state of the processor is saved and the program jumps to the MAP fault handling routine supplied by the programmer.

Additionally, the program can specify I/O or indirection protection, causing a MAP fault to occur when the processor encounters an I/O instruction or more than 15 levels of address indirection. This specification can also be accomplished by writing to the MAP status register. Also, by the same means, the processor can be instructed to interpret all I/O format instructions as Load Effective Address (**LEF**) instructions. Finally, enabling the MAP also enables the emulator trap facility, which is described in detail later.

Page 31 Register. Unless the MAP is enabled, no address translation occurs. All addresses issued by the CPU in the unmapped mode reference locations in the first 64 Kbytes (32 Kwords) of physical memory, that is, locations in physical pages 0-31. If a program operating in unmapped mode requires access to another part of memory, the *page 31 register* can be used to accomplish this.

A *Map Page 31* instruction (**DOB** *ac*,**MAP**) loads the page 31 register with a 10-bit translation address. This address corresponds to an entry for one page in a user map in the MAP register, but contains no protection bits. A CPU memory reference addressed to logical page 31 (address bits 1-5 are all ones) does not access a word in page 31. Instead, the memory reference accesses a word in the physical page specified by the 10 bits in the page 31 register. Thus, the page 31 register affords a 1-page *window* on memory to a program running in unmapped mode. After power up or a system reset, the page 31 register contains all ones (37_8), thereby mapping logical page 31 to physical page 31.

Extended Operations

Extended operation instructions (**XOP** and **XOP1**) transfer control to called procedures. An **XOP** instruction places all relevant return information on the stack and retrieves the address of the called procedure from a user-constructed table of procedure addresses. Control transfers to the procedure after the address has been retrieved.

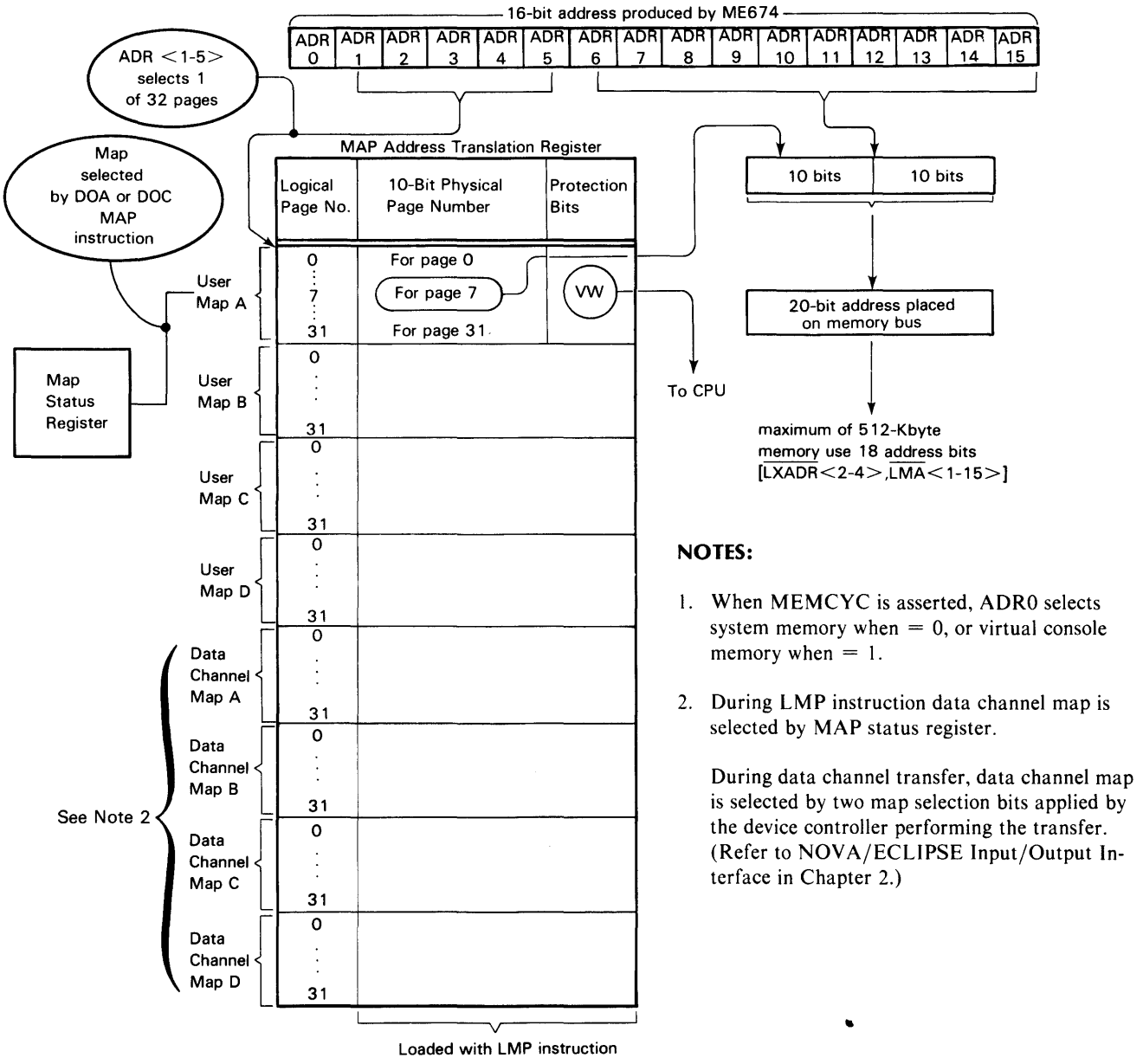


Figure 1.8 MAP address translation

Emulator Trap

The S/120 CPU has a hardware provision for instruction emulation. If the CPU encounters an undefined instruction while operating in mapped mode, it automatically jumps indirectly through location 11_8, provided that the location's contents are not zero. This location can contain the indirect address of an emulator routine. If the contents of location 11_8 are zero, an undefined instruction simply results in a **NOP** (no operation).

Error Checking and Correction Operations

When enabled, the error checking and correction (ERCC) unit detects and corrects single-bit errors occurring in any location in memory. Each memory word consists of 22 bits: 16 data bits plus 6 error checking and correction bits. Each time data is written into a memory location, a single error detection and correction chip constructs a 6-bit check field from the 16 data bits and appends the check field to the 16 data bits being written into memory.

Each time data is read from a memory location, the hardware encoder constructs a 6-bit check field based on the 16 data bits read from memory and compares the check field with the 6-bit check field read from memory. If the two check fields agree, no error occurred, and the 16 data bits are passed to the requesting facility. If the two check fields do not agree, an error has occurred.

The ERCC unit detects and corrects all single-bit errors. If the error is a single-bit error, the S/120 SPU pauses while the hardware encoder corrects the erroneous bit. The SPU then passes the 16 data bits to the requesting facility and rewrites the corrected data and check field into the memory location last accessed. The ERCC detects, but does not correct, all double-bit and some multiple-bit errors.

If errors occur, the ERCC unit can request an interrupt. In addition, the CPU can be programmed to request the address of the memory location containing the erroneous data and a 6-bit error code identifying the erroneous bit. A *Read Memory Fault Address* instruction (**DIA** *ac*,**ERCC**) returns to the CPU the 16 low-order physical memory address bits of the fault location. A *Read Memory Fault Code* instruction (**DIB** *ac*,**ERCC**) returns to the CPU the 4 high-order physical memory address bits of the fault location and the 6-bit error identification code.

In addition to correcting single-bit errors when a memory location is read or written, the S/120 also continuously tests and corrects all on-board single-bit memory errors at the rate of one location per 16 microseconds. As a result, the S/120 memory is completely checked every four seconds. This activity, called *sniffing*, minimizes the accumulation of correctable single-bit errors into multiple-bit errors. Errors detected by *sniffing* are not recorded, nor do they request a CPU interrupt.

I/O Operations

As many as 55 I/O controllers can be connected to the NOVA/ECLIPSE I/O bus. The CPU can address any one of these controllers using the device code occupying bits 10-15 of an I/O instruction.

The basic I/O instruction set is used to control I/O devices, to establish data channel operations, and to transfer status and control information to and from these devices. The basic I/O instruction set also transfers data to and from low speed or character type devices. Refer to specific Data General input/output device documentation for details about programming specific devices.

I/O interrupt control instructions offer the programmer the following selection of I/O control schemes.

Polling (No interrupts)
The CPU checks I/O device status under programmed control.

Single-level interrupts
(Interrupts with no priority system)
The CPU services one device at a time in the order determined by the timing of the interrupt and the physical location of its controller in the chassis.

Multiple-level interrupts
(Interrupts with a priority system)
The CPU services an interrupt from a selected device in the order described above: however, a higher priority device can interrupt a lower priority device's interrupt service routine. The interrupt handler accomplishes this by manipulating the devices' priority mask bits using the **MSKO** instruction.

Under an interrupt-driven scheme, the programmer can select one of the following methods to identify the interrupting device.

- Test the device's Busy/Done flags with an *I/O Skip* instruction
- Place the interrupter's device code in an accumulator with an **INTA** instruction
- Identify the interrupting device, save return information, and jump through a table to an individual device's interrupt handling routine with a **VCT** instruction

Summary of Instructions

This section presents all S/120 instructions in summary form. The instructions are grouped into the following categories.

- Computing instructions
- Program flow management
- Stack and data management
- System, device, and memory management

Computing Instructions

There are nine groups of computing instruction: add, subtract, multiply, divide, move, convert, logic, status, and computational skip. Tables 1.1 through 1.9 summarize these instructions.

Program Flow Management

There are five groups of instructions that manage program flow: noncomputational skip, jump, subroutine, interrupt, and accumulator. Tables 1.10 through 1.14 summarize these instructions.

Stack and Data Management

Stack and data management instructions are summarized in Table 1.15.

System, Device, and Memory Management

There are five groups of instructions for managing the system, its devices, and its memory: system call, basic I/O, I/O interrupt, CPU device, and MAP. The single system call instruction is described in the next paragraph; Tables 1.16 through 1.23 summarize the others.

System Call Handler. The *System Call* instruction turns off the MAP if it is on, can push a return block onto the stack, and places the address of the System Call handler in the program counter. This instruction has three formats.

SYC
SCL Equivalent to **SYC** 1,1
SVC Equivalent to **SYC** 0,0

Mnemonic	Instructions	Action
ADC	Add Complement	Adds an unsigned integer to the logical complement of another unsigned number.
ADD	Add	Adds the contents of one accumulator to the contents of another.
ADDI	Extended Add Immediate	Adds a signed integer in the range of −32,768 to +32,767 to the contents of an accumulator.
ADI	Add Immediate	Adds an unsigned integer in the range of 1 to 4 to the contents.
BAM	Block Add And Move	Moves blocks of memory words from one location to another, adding a constant to each one.
DAD	Decimal Add	Adds together the decimal digits found in bits 12-15 of two accumulators.
FAMS, FAMD	Add (Memory to FPAC)	Adds the floating-point number in memory to the floating-point number in an FPAC.
FAS, FAD	Add (FPAC to FPAC)	Adds the floating-point number in one FPAC to the floating-point number in another FPAC.
INC	Increment	Increments the contents of an accumulator.
ISZ, EISZ	Increment And Skip If Zero	Increments the addressed word, then skips if the incremented value is zero.

Table 1.1 Add instructions

Mnemonic	Instructions	Action
DSB	Decimal Subtract	Subtracts the decimal digit in bits 12-15 of one accumulator from the decimal digit in bits 12-15 of another acumulator.
DSZ, EDSZ	Decrement And Skip If Zero	Decrements the addressed word, then skips if the decremented value is zero.
FSMS, FSMD	Subtract (Memory from FPAC)	Subtracts the floating-point number in memory from the floating-point number in an FPAC.
FSS, FSD	Subtract (FPAC from FPAC)	Subtracts the floating-point number in one FPAC from the floating-point number in another FPAC.
SBI	Subtract Immediate	Subtracts an unsigned integer in the range of 1 to 4 from the contents of an accumulator.
SUB	Subtract	Subtracts the contents of one accumulator from the contents of another.

Table 1.2 Subtract instructions

Mnemonic	Instructions	Action
FMMS, FMMD	Multiply (Memory by FPAC)	Multiplies the floating-point number in memory by the floating-point number in an FPAC.
FMS, FMD	Multiply (FPAC by FPAC)	Multiplies the floating-point number in one FPAC by the floating-point number in another FPAC.
MUL	Unsigned Multiply	Multiplies the unsigned contents of two accumulators and adds the results to the unsigned contents of a third accumulator.
MULS	Signed Multiply	Multiplies the signed contents of two accumulators and adds the results to the signed contents of a third accumulator.

Table 1.3 Multiply instructions

Mnemonic	Instructions	Action
DIV	Unsigned Divide	Divides the unsigned 32-bit integer in two accumulators by the unsigned contents of a third accumulator.
DIVS	Signed Divide	Divides the signed 32-bit integer in two accumulators by the signed contents of a third accumulator.
DIVX	Sign Extend And Divide	Extends the sign of one accumulator into a second accumulator and performs a *Signed Divide* on the result.
FDMS, FDMD	Divide (FPAC by Memory)	Divides the floating-point number in an FPAC by a floating-point number in memory.
FDS, FDD	Divide (FPAC by FPAC)	Divides the floating-point number in one FPAC by the floating-point number in another FPAC.
FHLV	Halve	Divides the floating-point number in FPAC by 2.
HLV	Halve	Divides the unsigned contents of an accumulator by 2.

Table 1.4 Divide instructions

Mnemonic	Instructions	Action
BAM	Block Add And Move	Moves blocks of memory words from one location to another, adding a constant to each one.
BLM	Block Move	Moves blocks of memory words from one location to another.
CMT	Character Move Until True	Moves a string of bytes from one area of memory to another until a table- specified delimiter character is encountered or the source string is exhausted.
CMV	Character Move	Moves a string of bytes from one area of memory to another under control of the values in the four accumulators.
DHXL	Double Hex Shift Left	Shifts the 32-bit contents of two accumulators left 1 to 4 hex digits, depending on the value of a 2-bit number contained in the instruction.
DHXR	Double Hex Shift Right	Shifts the 32-bit contents of two accumulators right 1 to 4 hex digits, depending on the value of a 2-bit number contained in the instruction.
FEXP	Load Exponent	Places bits 1-7 of AC0 in bits 1-7 of the specified FPAC.
FLDS, FLDD	Load Floating Point	Copies a floating-point number from memory to a specified FPAC.
FMOV	Move Floating Point	Moves the contents of one FPAC to another FPAC.

Table 1.5 Move instructions (see continuation)

Mnemonic	Instructions	Action
FRH	Read High Word	Places the high-order 16 bits of an FPAC into AC0.
FSTS, FSTD	Store Floating Point	Copies the contents of a specified FPAC into memory.
HXL	Hex Shift Left	Shifts the contents of an accumulator left 1 to 4 hex digits, depending on the value of a 2-bit number contained in the instruction.
HXR	Hex Shift Right	Shifts the contents of an accumulator right 1 to 4 hex digits, depending on the value of a 2-bit number contained in the instruction.
LDA, ELDA	Load Accumulator	Loads data from memory to an accumulator.
LDB, ELDB	Load Byte	Places a byte of information into an accumulator.
MOV	Move	Moves the contents of an accumulator through the ALU.
POP	Pop Multiple Accumulators	Pops 1 to 4 words off the stack and places them in the indicated accumulators.
POPB	Pop Block	Returns control from a *System Call* routine or an I/O interrupt handler that does not use the stack change facility of the *Vector* instruction.
PSH	Push Multiple Accumulators	Pushes the contents of 1 to 4 accumulators onto the stack.
STA, ESTA	Store Accumulator	Stores data in memory from an accumulator.
STB, ESTB	Store Byte	Stores the right byte of an accumulator in a byte of memory.
XCH	Exchange Accumulators	Exchanges the contents of two accumulators.

Table 1.5 Move instructions (concluded)

Mnemonic	Instructions	Action
CTR	Character Translate	Translates a string of bytes from one data representation to another, and either moves it to another area of memory or compares it to a second string of bytes.
FFAS	Fix To AC	Converts the integer portion of a floating-point number to a signed two's complement integer and places the result in an accumulator.
FFMD	Fix To Memory	Converts the integer portion of a floating-point number to double-precision integer format and stores the result in two memory locations.
FINT	Integerize	Sets the fractional portion of the floating-point number in the specified FPAC to zero and normalizes the result.
FLAS	Float From AC	Converts a signed two's complement number in an accumulator to a single-precision floating-point number.
FLMD	Float From Memory	Converts the contents of two memory locations in integer format to floating-point format and places the result in a specified FPAC.
FNOM	Normalize	Normalizes the floating-point number in FPAC.
FSCAL	Scale	Shifts the mantissa of the floating-point number in FPAC either right or left, depending upon the contents of bits 1-7 of AC0.

Table 1.6 Convert instructions

Mnemonic	Instructions	Action
ANC	AND With Complemented Source	Forms the logical AND of the contents of one accumulator and the logical complement of the contents of another accumulator.
AND	AND	Forms the logical AND of the contents of two accumulators.
ANDI	AND Immediate	Forms the logical AND of a 16-bit number contained in the instruction and the contents of an accumulator.
BTO	Set Bit To One	Sets the bit addressed by the bit pointer to 1.
BTZ	Set Bit To Zero	Sets the bit addressed by the bit pointer to 0.
CMP	Character Compare	Compares one string of characters in memory to another string.
COB	Count Bits	Counts the number of ones in one accumulator and adds that number to the second accumulator.
COM	Complement	Forms the logical complement of the contents of an accumulator.
DLSH	Double Logical Shift	Shifts the 32-bit contents of two accumulators left or right depending on the contents of a third accumulator.
FAB	Absolute Value	Sets the sign bit of an FPAC to 0.
FCMP	Compare Floating Point	Compares two floating-point numbers and sets the Z and N flags accordingly.
FNEG	Negate	Inverts the sign bit of the FPAC.
IOR	Inclusive OR	Forms the logical inclusive OR of the contents of two accumulators.
IORI	Inclusive OR Immediate	Forms the logical inclusive OR of a 16-bit number contained in the instruction and the contents of an accumulator.
LOB	Locate Lead Bit	Counts the number of high-order zeros in one accumulator and adds that number to the second accumulator.
LRB	Locate And Reset Lead Bit	Performs a *Locate Lead Bit* instruction and sets the lead bit to 0.
LSH	Logical Shift	Shifts the contents of an accumulator left or right, depending on the contents of another accumulator.
NEG	Negate	Forms the two's complement of the contents of an accumulator.
XOR	Exclusive OR	Forms the logical exclusive OR of the contents of two accumulators.
XORI	Exclusive OR Immediate	Forms the logical exclusive OR of a 16-bit number contained in the instruction and the contents of an accumulator.

Table 1.7 Logic instructions

Mnemonic	Instructions	Action
DIA MAP	Read MAP Status	Returns the status of the MAP, including the following conditions: last map enabled (by a DOA); MAP state (on/off); type of last MAP fault; last map loaded (by an LMP); state of LEF, I/O protection, write protection, and indirect protection (on/off); data channel map state; user mode (on/off).
DIS	Data In Status	Returns the status of a specified I/O device.*
DIS CPU	Read Processor Status	Returns the status of the processor, including the following conditions: power fail, interrupt on, Break key, power-up or reset, halt instruction, interrupt request, program load key, run light, valid memory data, and memory capacity.
FCLE	Clear Errors	Sets bits 0-4 of the FPSR to 0.
FLST	Load Floating-Point Status	Copies the contents of two specified memory locations to the FPSR.
FSST	Store Floating-Point Status	Copies the contents of the FPSR to two memory locations.

Table 1.8 Status Instructions

**Refer to Figure 1.2 for the accumulator format for this instruction.*

Mnemonic	Instructions	Action
CLM	Compare To Limits	Compares a signed integer with two other numbers and skips if first integer is between the other two.
DSZ, EDSZ	Decrement And Skip If Zero	Decrements the addressed word, then skips if the decremented value is zero.
FSEQ	Skip On Zero	Skips the next sequential word if the Z flag of the FPSR is one.
FSGE	Skip On Greater Than Or Equal To Zero	Skips the next sequential word if the N flag of the FPSR is zero.
FSGT	Skip On Greater Than Zero	Skips the next sequential word if both the Z and N flags of the FPSR are zero.
FSLE	Skip On Less Than Or Equal To Zero	Skips the next sequential word if either the Z flag or the N flag of the FPSR is one.
FSLT	Skip On Less Than Zero	Skips the next sequential word if the N flag of the FPSR is one.
FSND	Skip On No Zero Divide	Skips the next sequential word if the divide by zero (DVZ) flag of the FPSR is zero.
FSNE	Skip On Nonzero	Skips the next sequential word if the Z flag of the FPSR is zero.
FSNER	Skip On No Error	Skips the next sequential word if bits 1-4 of the FPSR are all zero.
FSNM	Skip On No Mantissa Overflow	Skips the next sequential word if the mantissa overflow (MOF) flag of the FPSR is zero.

Table 1.9 Computational skip instructions (see continuation)

Mnemonic	Instructions	Action
FSNO	Skip On No Overflow	Skips the next sequential word if the overflow (OVF) flag of the FPSR is zero.
FSNOD	Skip On No Overflow And No Zero Divide	Skips the next sequential word if both the overflow (OVF) flag and the divide by zero (DVZ) flag of the FPSR are zero.
FSNU	Skip On No Underflow	Skips the next sequential word if the underflow (UNF) flag of the FPSR is zero.
FSNUD	Skip On No Underflow And No Zero Divide	Skips the next sequential word if both the underflow (UNF) flag and the divide by zero (DVZ) flag of the FPSR are zero.
FSNUO	Skip On No Underflow And No Overflow	Skips the next sequential word if both the underflow (UNF) flag and the overflow (OVF) flag of the FPSR are zero.
ISZ, EISZ	Increment And Skip If Zero	Increments the addressed word, then skips if the incremented value is zero.
SGE	Skip If ACS Greater Than Or Equal To ACD	Compares the signed integers in two accumulators and skips if the first is greater than or equal to the second.
SGT	Skip If ACS Greater Than ACD	Compares the signed integers in two accumulators and skips if the first is greater than the second.
SNB	Skip On Nonzero Bit	Skips the next sequential word if the bit addressed by the bit pointer is one.
SZB	Skip On Zero Bit	Skips the next sequential word if the bit addressed by the bit pointer is zero.
SZBO	Skip On Zero Bit And Set To One	Sets the bit addressed by the bit pointer to one and skips the next sequential word if the bit was originally zero.

Table 1.9 Computational skip instructions (concluded)

Mnemonic	Instructions	Action
CLM	Compare To Limits	Compares a signed integer with two other numbers and skips if first integer is between the other two.
FNS	No Skip	No operation.
FSA	Skip Always	Skips the next sequential word.

Table 1.10 Noncomputational skip instructions

Mnemonic	Instructions	Action
DSPA	Dispatch	Compares a signed integer with two other numbers and continues sequential execution if the integer is not between the others; otherwise, uses the integer as an index into a table and places indexed value in the program counter.
JMP, EJMP	Jump	Places an effective address in the program counter.
JSR, EJSR	Jump To Subroutine	Increments program counter and stores incremented value in AC3; then places a new address in the program counter.
POPJ	Pop PC and Jump	Pops the top word off the stack and places it in the program counter.
PSHJ	Push PC and Jump	Pushes the address of the next sequential instruction onto the stack, computes the effective address E, and places it in the program counter.

Table 1.11 Jump instructions

Mnemonic	Instructions	Action
JSR, EJSR	Jump To Subroutine	Increments program counter and stores incremented value in AC3; then places a new address in the program counter.
PSHR	Push Return Address	Pushes the address of the instruction after the next sequential instruction onto the stack.
RSTR	Restore	Returns control from certain types of I/O interrupts.
RTN	Return	Returns control from subroutines that issue a *Save* instruction at their entry points.
SAVE	Save	Saves the information required by the *Return* instruction.
XOP	Extended Operation	Pushes a return block on the stack, placing the address in the stack of the specified accumulators into AC2 and AC3, and transfers control to 1 of 32 other procedures with the XOP table.
XOP1	Extended Operation	Same as XOP, except that 32 is added to the entry number before entering the XOP table and only 16 table entries can be specified.

Table 1.12 Subroutine instructions

Mnemonic	Instructions	Action
DSPA	Dispatch	Compares a signed integer with two other numbers and continues sequential execution if the integer is not between the others; otherwise, uses the integer as an index into a table and places indexed value in the program counter.
FTD	Trap Disable	Sets the trap enable flag of the FPSR to zero.
FTE	Trap Enable	Sets the trap enable flag of the FPSR to one.

Table 1.13 Interrupt instructions

Mnemonic	Instructions	Action
LEF, ELEF	Load Effective Address	Places an effective address in an accumulator.
XCT	Execute	Executes contents of an accumulator as an instruction.

Table 1.14 Accumulator instructions

Mnemonic	Instructions	Action
FPOP	Pop Floating-Point State	Pops an 18-word floating-point return block off the user stack and alters the state of the floating-point unit.
FPSH	Push Floating-Point State	Pushes an 18-word floating-point return block onto the user stack.
MSP	Modify Stack Pointer	Changes the value of the stack pointer and checks for overflow.
POP	Pop Multiple Accumulators	Pops 1 to 4 words off the stack and places them in the indicated accumulators.
POPB	Pop Block	Returns control from a *System Call* routine or an I/O interrupt handler that does not use the stack change facility of the *Vector* instruction.
PSH	Push Multiple Accumulators	Pushes the contents of 1 to 4 accumulators onto the stack.
PSHJ	Push PC Jump	Pushes the address of the next sequential instruction onto the stack, computes the effective address *E*, and places it in the program counter.
PSHR	Push Return Address	Pushes the address of the instruction after the next sequential instruction onto the stack.
RSTR	Restore	Returns control from certain types of I/O interrupts.
RTN	Return	Returns control from subroutines that issue a *Save* instruction at their entry points.
SAVE	Save	Saves the information required by the *Return* instruction.
XOP	Extended Operation	Pushes a return block on the stack, placing the address in the stack of the specified accumulators into AC2 and AC3, and transfers control to 1 of 32 other procedures via the XOP table.
XOP1	Extended Operation	Same as XOP except that 32 is added to the entry number before entering the XOP table and only 16 table entries can be specified.

Table 1.15 Stack and data management instructions

Mnemonic	Instructions	Action
DIA*[f]*	Data In A	Transfers data from the A buffer of an I/O device to an accumulator.
DIB*[f]*	Data In B	Transfers data from the B buffer of an I/O device to an accumulator.
DIC*[f]*	Data In C	Transfers data from the C buffer of an I/O device to an accumulator.
DIS	Data in Status	Returns the status of a specifed I/O device.*
DOA*[f]*	Data Out A	Transfers data from an accumulator to the A buffer of an I/O device.
DOB*[f]*	Data Out B	Transfers data from an accumulator to the B buffer of an I/O device.
DOC*[f]*	Data Out C	Transfers data from an accumulator to the C buffer of an I/O device.
NIO*[f]*	No I/O Transfer	Sets Busy or Done flag. No I/O transfer occurs.

Table 1.16 Basic I/O instructions

**Refer to Figure 1.2 for the accumulator format of this instruction.*

Mnemonic	Flag Value	Action
[f] omitted	00	Does not alter the Busy and Done flags.
[f]=S	01	Starts the device; sets Busy flag to one and Done flag to zero.
[f]=C	10	Idles the device; sets Busy flag to zero. and sets Done flag to zero.
[f]=P	11	I/O pulse; effect depends upon device.

Table 1.17 I/O command flags

Mnemonic	Instructions	Action
INTA (DIB*[f]* CPU)	Interrupt Acknowledge	Returns the device code of an interrupting device.
INTDS (NIOC CPU)	Interrupt Disable	Sets CPU Interrupt On flag to zero.
INTEN (NIOS CPU)	Interrupt Enable	Sets CPU Interrupt On flag to one.
MSKO (DOB*[f]* CPU)	Mask Out	Changes the priority mask.
POPB	Pop Block	Returns control from a *System Call* routine or an I/O interrupt handler that does not use the stack change facility of the *Vector* instruction.
RSTR	Restore	Returns control from I/O interrupts that use the stack change facility of the VCT instruction.
SKP*[t]*	I/O Skip	Skips if the I/O condition *t* is true.
VCT	Vector On Interrupting Device Code	Identifies highest priority interrupt; passes control through a table to a handler routine for device.
XCT	Execute	Executes contents of an accumulator as an instruction.

Table 1.18 I/O Interrupt instructions

Mnemonic	Flag Value	Action
[t]=BN	00	Tests Busy flag for nonzero.
[t]=BZ	01	Tests Busy flag for zero.
[t]=DN	10	Tests Done flag for nonzero.
[t]=DZ	11	Tests Done flag for zero.

Table 1.19 I/O skip flags

Mnemonic	Instructions	Action
DIS CPU	Read Processor Status	Returns the status of the processor, including the following conditions: power fail, interrupt on, Break Key, power-up or reset, halt instructions, interrupt request, valid data, and memory capacity. Program Load key, run light, validity of memory data, and memory capacity.*
HALT (DOC*[f]* CPU)	Halt	Stops the processor.
INTA (DIB*[f]* CPU)	Interrupt Acknowledge	Returns the device code of an interrupting device.
INTDS (NIOC CPU)	Interrupt Disable	Sets CPU Interrupt On flag to zero.
INTEN (NIOS CPU)	Interrupt Enable	Sets CPU Interrupt On flag to one.
IORST (DICC*[f]* CPU)	Reset	Sets all Busy and Done flags and the priority mask to zero.
MSKO (DOB*[f]* CPU)	Mask Out	Changes the priority mask.
READS (DIA*[f]* CPU)	Read Switches	Places the contents of the virtual console register into an accumulator.**
SKP*[t]* CPU	CPU Skip	Tests the Interrupt On or Power Fail flag and skips the next sequential word if the test condition is true.

Table 1.20 CPU device instructions

**Refer to Figure 1.2 for the accumulator format for this instruction.*

***Refer to Chapter 7 for a description of this register.*

Mnemonic	Flag Value	Action
[t]=BN	00	Tests Interrupt On flag for nonzero.
[t]=BZ	01	Tests Interrupt On flag for zero.
[t]=DN	10	Tests Power Fail flag for nonzero.
[t]=DZ	11	Tests Power Fail flag for zero.

Table 1.21 CPU skip flags

Mnemonic	Instructions	Action
DIA MAP	Read Map Status	Reads the status of the current map.
DIC MAP	Page Check	Provides the identity and some characteristics of the physical page that corresponds to the logical page identified by the immediately preceding *Initiate Page Check* instruction.
DOA MAP	Load Map Status	Defines the parameters of a new map.
DOB MAP	Map Supervisor Page 31	Specifies the physical page corresponding to logical page 31 of unmapped address space.
DOC MAP	Initiate Page Check	Identifies a logical page; selects map without changing status.
LMP	Load Map	Loads successive words from memory into the MAP, where they are used to define a user or data channel map.
NIOP MAP	Map Single Cycle	Maps one memory reference using the last user map, or turns off memory mapping.

Table 1.22 MAP instructions

Mnemonic	Instructions	Action
DOA	Enable ERCC	Enables the ERCC facility according to the setting of bits 13-15 of the specified accumulator.
DIA	Read Memory Fault Address	Returns the low-order bits of the memory location which has produced an error.
DIB	Read Memory Fault Code	Returns a 6-bit error code that tells which bit was in error. Also returns the two most significant bits of the memory fault address.

Table 1.23 ERCC instructions

Reserved Memory Locations

Table 1.24 lists program-accessible locations in page 0 of memory. These locations are reserved for storage of data with special meanings for the CPU.

Memory Address (Octal)	Contents or Use
00000	Return address for I/O interrupts. Also, first instruction of auto-restart routine.
00001	Address of I/O interrupt handling routine. Indirectable.
00002	Address of system call instruction handler. Indirectable.
00003	Address of MAP fault handling routine. Indirectable.
00004	Address of the top of the vector stack. Nonindirectable.
00005	Current interrupt priority mask.
00006	Address of the last normally usable location in the vector stack.
00007	Address of the vector stack fault handler. Indirectable.
00011	Address of emulator trap handler. (If contents equal zero, an NOP is performed.)
00040	Address of the top of the stack. Nonindirectable.
00041	Address of the start of the current stack frame minus one. Nonindirectable.
00042	Address of stack upper limit.
00043	Address of stack fault routine. Indirectable.
00044	Address of the beginning of the XOP table. Nonindirectable.
00045	Address of floating-point fault handler. Indirectable.
00046	Reserved for future use.
00047	Reserved for future use.

Table 1.24 Reserved memory locations

Instruction Execution Times

Table 1.25 lists typical execution times for all instructions. The notes referenced follow the table.

Instruction	Execution Time (microsec.)	CPU Cycles	Notes
ADC	0.50	1	1
ADD	0.50	1	1
ADDI	1.00	2	
ADI	0.50	1	
ANC	0.50	1	
AND	0.50	1	1
ANDI	1.00	2	
BAM	6.50 + 2.5/word	13 + 5/word	2,3
BLM	3.50 + 2.0/word	7 + 4/word	2,3
BTO	5.50	11	4
BTZ	4.50	9	4
CLM	3.50	7	1
CMP	8.50 + 7.0/byte	17 + 14/byte	3
CMT	1.50 + 9.0/byte	3 + 18/byte	3
CMV	5.50 + 5.5/byte	11 + 11/byte	3
COB	7.50	15	4
COM	0.50	1	1
CTR	5.50 + 7.0 or 9.5/byte	11 + 14 or 19/byte	5
DAD	8.00	16	
DHXL	7.50	15	5
DHXR	7.00	14	5
DIA,B,C	3.50	7	
DIS	3.50	7	
DIV	12.00	24	
DIVS	20.50	41	
DIVX	19.50	39	
DLSH	6.50	13	4
DOA,B,C	3.50	7	
DSB	8.00	16	
DSPA	6.00	12	2
DSZ	2.00	4	2,1
EDSZ	2.00	4	2,1
EISZ	2.00	4	2,1
EJMP	1.50	3	2
EJSR	1.50	3	2
ELDA	1.00	2	2
ELDB	3.50	7	
ELEF	1.00	2	2
ESTA	1.00	2	2
ESTB	3.50	7	
FAB	11.00	22	7
FAD	87.00	174	7
FAMD	92.00	184	7,8
FAMS	67.00	134	7,8
FAS	65.00	130	7
FCLE	3.00	6	
FCMP	29.00	58	

Table 1.25 Instruction execution times

Instruction	Execution Time (microsec.)	CPU Cycles	Notes
FDD	900.00	1800	7,9
FDMD	900.00	1800	7,8,9
FDMS	190.00	380	7,8,9
FDS	190.00	380	7,9
FEXP	13.00	26	7
FFAS	46.00	92	7
FFMD	45.50	90	7, 8
FHLV	30.00	60	7, 9
FINT	20.50	41	7
FLAS	31.00	62	
FLDD	16.50	32	8
FLDS	15.00	30	8
FLMD	31.00	62	8
FLST	11.50	23	7, 8
FMD	266.00	532	7
FMMD	266.00	532	7, 8
FMMS	80.50	161	7, 8
FMOV	17.00	34	7
FMS	80.50	161	7
FNEG	12.50	25	7
FNOM	43.00	86	
FNS	1.00	2	
FPOP	32.00	64	
FPSH	31.50	63	
FRH	6.00	12	
FSA	1.50	3	
FSCAL	48.00	96	7
FSD	87.00	174	7
FSEQ	5.00	10	
FSGE	4.50	9	
FSGT	5.00	10	
FSLE	5.00	10	
FSLT	4.50	9	
FSMD	92.00	184	7, 8
FSMS	67.00	134	7, 8
FSND	5.00	10	
FSNE	5.00	10	
FSNER	5.00	10	
FSNM	5.00	10	
FSNO	5.00	10	
FSNOD	5.00	10	
FSNU	5.00	10	
FSNUD	5.00	10	
FSNUO	5.00	10	
FSS	65.00	130	7
FSST	7.50	15	8
FSTD	12.50	25	8
FSTS	9.50	19	8

Table 1.25 Instruction execution times (continued)

Instruction	Execution Time (microsec.)	CPU cycles	Notes
FTD	3.00	6	
FTE	5.00	10	7
HALT	6.50	13	
HLV	1.50	3	
HXL	2.00	4	5
HXR	2.00	4	5
INC	0.50	1	1
INTA	3.50	7	
IOR	1.50	3	
IORI	1.50	3	
IORST	9.00	18	
ISZ	2.00	4	2, 1
JMP	1.50	3	2
JSR	1.50	3	2
LDA	1.00	2	2
LDB	1.50	3	
LEF	1.00	2	2
LMP	4.50	9	2,10
LOB	4.00	8	5
LRB	5.00	10	5
LSH	8.50	17	5
MOV	0.50	1	1
MSKO	3.50	7	
MSP	3.00	6	11
MUL	9.50	19	
MULS	9.50	19	
NEG	0.50	1	1
NIO	3.50	7	
POP	1.50	3	12
POPB	6.50	13	
POPJ	3.00	6	
PSH	3.00	6	11, 12
PSHJ	4.50	9	11
PSHR	4.50	9	
RSTR	10.50	21	
RTN	6.00	12	
SAVE	8.00	16	11
SBI	0.50	1	
SGE	0.50	1	1
SGT	0.50	1	1
SKP	1.50	3	1
SNB	4.50	9	1,4
STA	1.00	2	2
STB	1.50	3	
SUB	0.50	1	1
SYC	10.50	21	2
SZB	4.00	8	1,4
SZBO	6.00	12	1,4
VCT	8.00 to 34.00	16 to 68	13
XCH	1.50	3	
XCT	2.00	4	14
XOP	20.50	41	
XOP1	21.50	43	
XOR	2.50	5	
XORI	2.50	5	

Instruction execution times (continued)

Table 1.25 Instruction execution times (concluded)

[1] *If skip occurs, add 0.50.*

[2] *If indirect chain followed, add 0.50 + (number of indirects − 1)*1.00.*

[3] *For each item moved, add the amount shown.*

[4] *Execution time is operand-dependent.*

[5] *If ACS<>ACD, add 1.00 + 2(Number of indirects).*

[6] *Byte moves require 7.0 μs/byte; compares require 9.5 μs/byte.*

[7] *This instruction can take a floating point trap, which would add 19μs to the execution time.*

[8] *This instruction does an effective address calculation, which can add 15μs to the execution time.*

[9] *Floating-point divide execution times depend on the number of zero and one bits in the quotient (the more one's contained in the quotient, the longer the execution time).*

[10] *For each word moved, add 1.50.*

[11] *For stack overflow, add 9.00 + (Note 2).*

[12] *For each accumulator pushed/popped, add 0.50.*

[13] *Vector execution times depend on the mode employed.*

[14] *Add instruction execution time.*

Power-up Response

As power begins to rise, the power supply asserts **PWROK** and **PWRFAIL** both low to the SPU board. These signals remain low until the power requirements are above specified limits. While **PWROK** is asserted low, the SPU board asserts system reset signals, **RESET** and **RESET**. These signals

- Initialize the central processor unit (mE674 CPU)
- Initialize the external microcode controllers (XMCs)
- Initialize the system input/output controller (mE676 SIO)
- Initialize the data channel timing state machine in the I/O interface
- Set all device Busy and Done flags to zero
- Transmit I/O Reset on the NOVA/ECLIPSE I/O bus
- Initialize the error checking and correction unit to write check code, and enable error checking and correction during memory refresh cycles (sniffing).

Approximately 10 microseconds after power is applied to the SPU, the CPU is initialized and enters the Idle state. During initialization, CPU firmware clears the CPU status register, clears the MAP status register and sets the contents of the MAP page 31 register to 37_8 (logical page 31_{10} translates to physical page 31_{10}.)

When power requirements exceed specified limits, the power supply asserts **PWROK** and **PWRFAIL** both high. As a result, the system reset lines are released. In addition, the power monitor within the SIO chip sets the power-up bit (bit 4) in the CPU status register and asserts a nonmaskable interrupt, causing the CPU to exit the Idle state.

The CPU then enters virtual console, which clears the power-up bit and performs a short diagnostic routine. In about 1 second, this routine checks physical memory locations 0 through 32K, the virtual console read/write memory, the CPU, the system console interface, and the error checking and correction hardware. If the routine detects an error, it sends a one-letter error code to the system console; the virtual console waits for the operator to depress the Break key on the system console. (The error codes are described in Chapter 7.)

When the self-test completes and if the front console is unlocked or if the automatic program load (APL) device code jumpers supply device code zero, the CPU remains in virtual console until a command is entered. If the front console is locked, the CPU performs an automatic program load from the device specified by the APL jumpers. These processes are illustrated in Figure 1.9.

Power Fail/Autorestart

When an abnormal power condition occurs, the power supply asserts **PWRFAIL**, which generates the signal **PFAIL** on the SPU board. This signal directs a power monitor within the SIO chip to set the CPU Done flag and issue an interrupt request to the CPU. Unless CPU interrupts are disabled, the CPU enters a user-supplied interrupt handler routine. This routine should issue an *Interrupt Acknowledge* instruction (**INTA**) to determine the interrupt's cause.

If a power failure caused the interrupt, the SIO chip responds to **INTA** by returning device code 0 to the CPU. Since an **INTA** instruction with no interrupting device also returns device code 0, the interrupt handler must execute a **SKPDN CPU.** This insures that the device code 0 is being returned because of a power failure. If the CPU (device code 77_8) Done flag is set, the interrupt handler should respond by entering a user-supplied powerfail interrupt routine. The effect of the powerfail routine depends on whether battery backup is present.

If battery backup is present, the powerfail routine should save the state of the processor in system memory, load a return instruction into physical memory location 0 and then execute a *Halt* instruction. (See note below reguarding the *Halt* procedure.)

If there is no battery backup, the routine simply executes a *Halt* instruction. If the *Halt* instruction is executed within one millisecond and if the Halt Dispatch SIO operating characteristic is selected, virtual console is entered; otherwise a hard halt occurs. (Refer to SIO jumpering in the section "Installation and Jumpering" and CPU status register in the section "CPU Support Elements" later in this chapter.)

> **NOTE:** *It is not necessary for the interrupt handler to enter a power failure routine and execute a* Halt *instruction. Instead, the user can ignore the powerfail interrupt by disabling CPU interrupts and continue to execute instructions until dc voltages fall below specifications. This is not recommended, however. When dc power falls below specifications, the SPU system reset logic asserts a* reset, *stopping the execution of instructions. An instruction executing when* system reset *occurs may not execute correctly.*

If battery backup is present and the batteries have not been exhausted during power disruption, the virtual console automatically clears the powerfail interrupt when power is restored. The action next taken by the virtual console depends on whether the front console is locked. If it is locked, the instruction contained in physical memory location 0 executes, returning control to the user's program. If the front console is unlocked, the virtual console

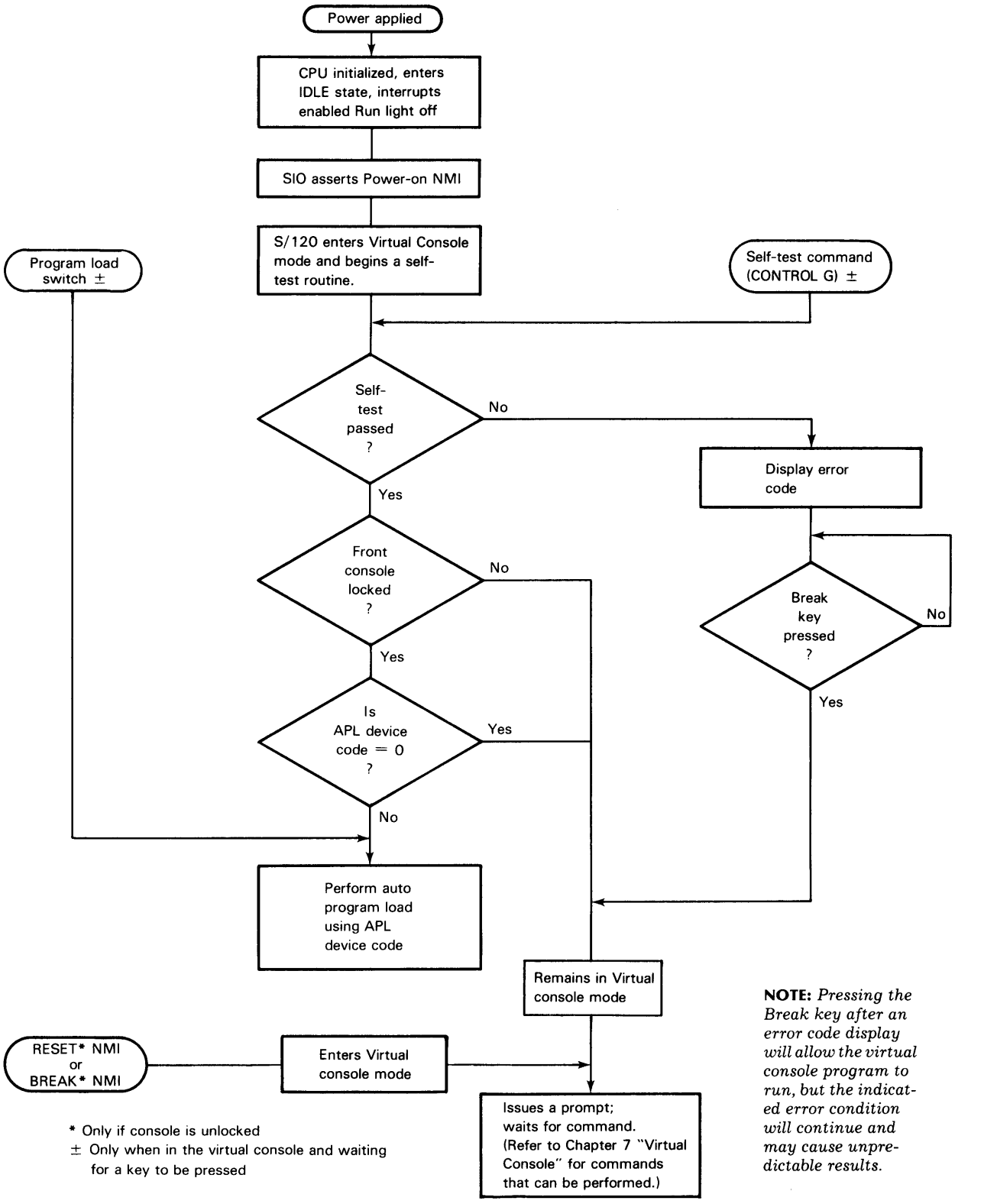


Figure 1.9 Power-up sequence for the ECLIPSE S/120

retains control and issues a prompt character to the system console when power returns. The user can continue by typing **OR** on the system console, if

- Battery backup is present.
- The state of the processor had been saved.
- A return instruction was stored in physical memory location 0 when the power failure occurred.

Otherwise, the user can initiate a program load or other appropriate action. (Refer to Chapter 7 for more information on the virtual console.)

If the batteries have been exhausted when power is restored, the virtual console performs the power-up response described above and shown in Figure 1.9.

The sequence of events described in the section is illustrated in Figure 1.11.

Installation and Jumpering

The S/120 SPU printed circuit board fits into Slot 1 of a standard DGC chassis. Figure 1.10 shows the board dimensions.

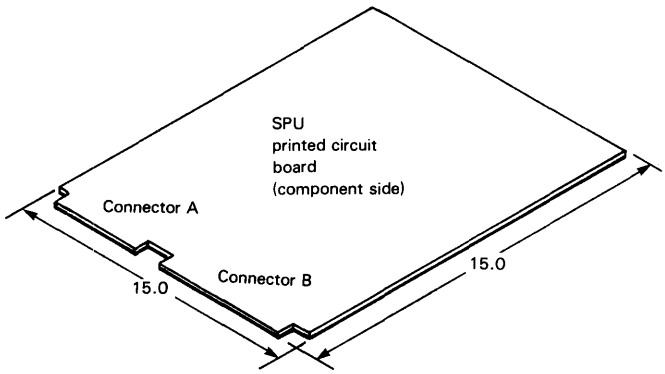


Figure 1.10 Board dimensions

Power Requirements

Table 1.26 lists the normal power requirements of the S/120 SPU board and Table 1.27 lists the battery-backed power required to maintain system memory data during a power disruption.

Supply Voltage	Current Draw Maximum	Current Draw Typical	Pin Numbers
+5V	5.0A	4.0A	A-3, A-4, A-97, A-98 B-3, B-4, B-38, B-97, B-98
+5VMEM	3.0A	2.5A	B-93, B-94
−5V	0	0	A-6, B-81
−5VMEM	0	0	B-91
+12V	50mA	10mA	B-87, B-88, B-90
+12VMEM	0	0	B-70, B-71, B-72
−12V	20mA	9mA	B-77
+15V	0	0	A-10, B-46, B-84
Ground	—	—	A-1, A-2, A-14, A-25, A-33, A-34, A-37, A-41, A-45, A-65, A-99, A-100 B-1, B-2, B-21, B-39, B-50, B-68, B-80, B-89, B-92, B-99, B-100

Table 1.26 Power requirements (normal conditions)

Battery-Backed Voltage	Current Draw Maximum	Current Draw Typical
+5VMEM	3.0A	1.5A
−5VMEM	0	0
+12VMEM	0	0

Table 1.27 Battery backup power requirements

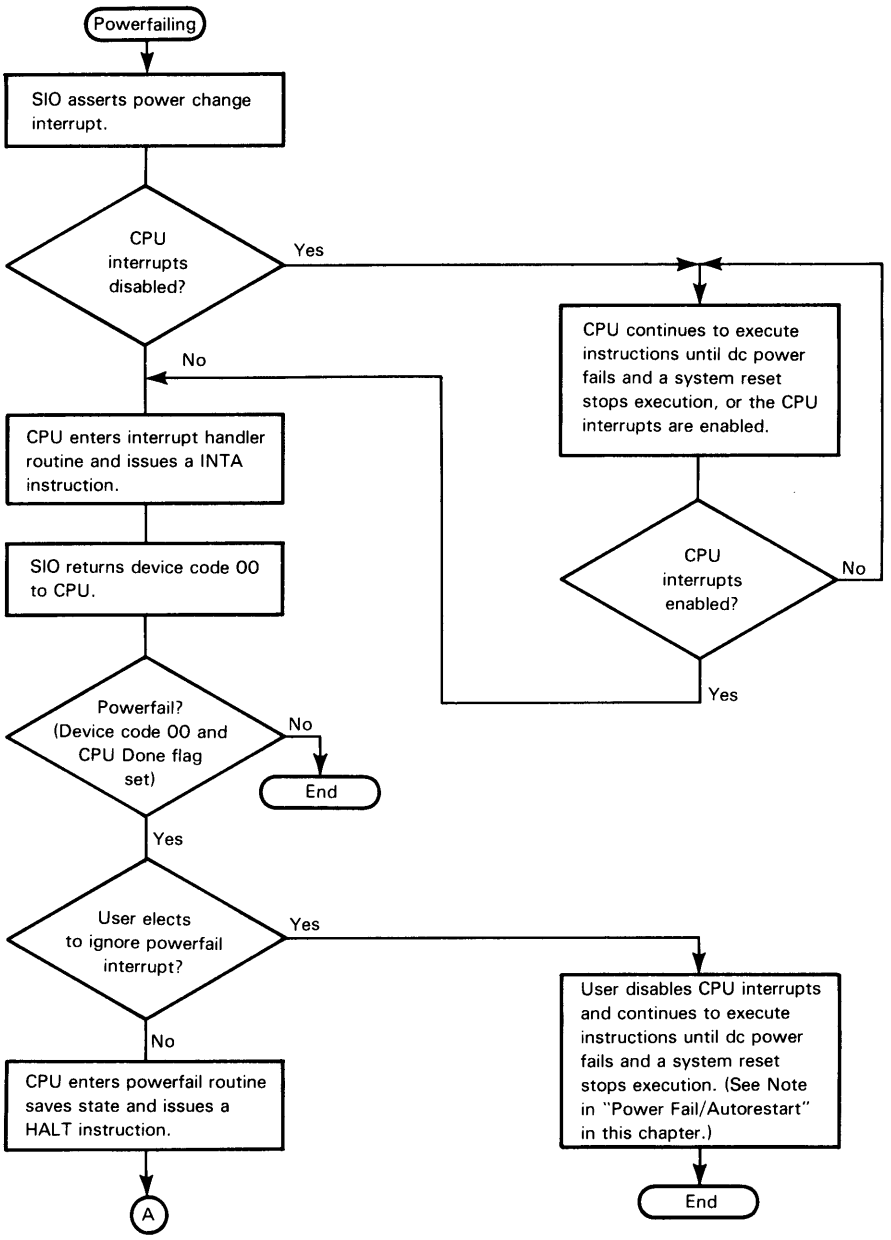
Jumpering

The locations of the nineteen jumpers present on the SPU board are shown in Figure 1.12.

Jumpers W1 and W2 select the operating characteristics of the asynchronous interface receiver as shown in Table 1.28.

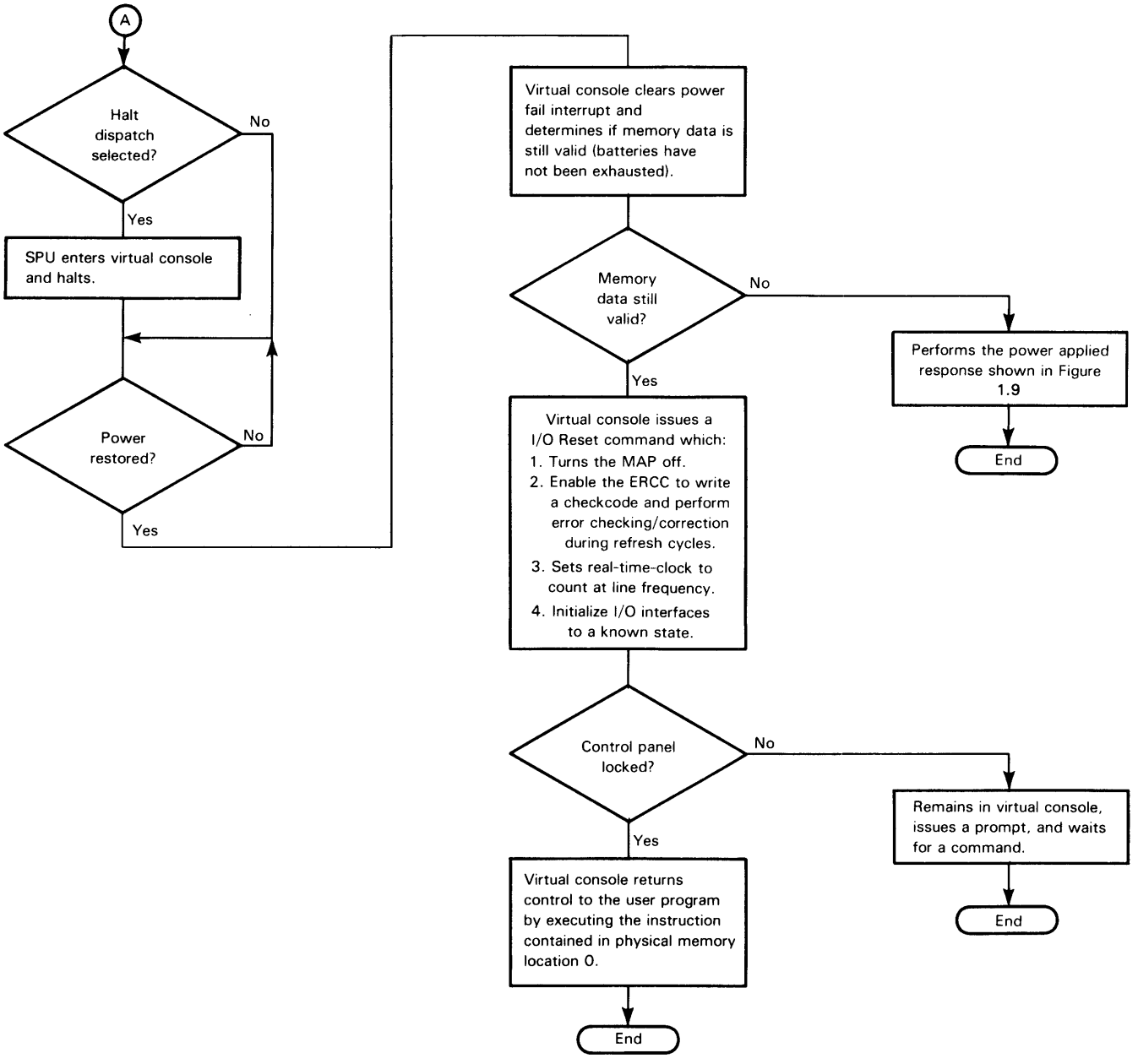
Line Type	Jumpers W1	Jumpers W2
EIA RS-232-C	Out	Out
Current Loop (600 baud and below)	In	In
Current Loop (above 600 baud)	In	Out

Table 1.28 Asynchronous interface receiver, jumper configurations



DG-09042

Figure 1.11 Powerfail/autorestart sequence


A
Halt dispatch selected?
No
Yes
SPU enters virtual console and halts.
Power restored?
No
Yes
Virtual console clears power fail interrupt and determines if memory data is still valid (batteries have not been exhausted).
Memory data still valid?
No
Yes
Performs the power applied response shown in Figure 1.9
End
Virtual console issues a I/O Reset command which:
1. Turns the MAP off.
2. Enable the ERCC to write a checkcode and perform error checking/correction during refresh cycles.
3. Sets real-time-clock to count at line frequency.
4. Initialize I/O interfaces to a known state.
Control panel locked?
No
Yes
Remains in virtual console, issues a prompt, and waits for a command.
End
Virtual console returns control to the user program by executing the instruction contained in physical memory location 0.
End

The APL jumpers (W3 through W9) select the automatic program load device type and device code. To make this selection, install the APL jumpers shown in Figure 1.12 as follows.

- Install jumper W9 to select automatic program loading from a programmed I/O device.
- Remove (or omit) jumper W9 to select automatic program loading from a data channel device.
- Install or remove jumpers W3 through W8 as required to enter the device code of the APL device:

  to enter a 0 bit, insert a jumper
  to enter a 1 remove (or omit) a jumper

  Jumper W3 enters the least significant bit; jumper W8 enters the most significant bit.

  Example: to enter device code 27_8, insert jumpers W6 and W8 and remove (or omit) all others.

The SIO jumpers (W10 through W17) select operating characteristics of the system input/output controller. To make this selection, install the SIO jumpers shown in Figure 1.12 as follows.

- Install or remove jumpers W10 through W13 to set the transmission rate of the asynchronous interface. These jumpers are detailed in Table 1.29.
- Install or remove jumpers W14 and W15 to set the counter rate for the programmable interval timer (PIT). These jumpers are detailed in Table 1.30.
- Install jumper W16, the Break enable jumper, if you do not want the system console Break key to generate CPU interrupts. Remove (or omit) jumper W16 if you want the system console Break key to generate CPU interrupts.
- Install jumper W17, the Halt dispatch jumper, to bring the processor to a hard halt when it encounters a *Halt* instruction. Remove (or omit) jumper W17 to direct the S/120 processor to enter the virtual console when it encounters a *Halt* instruction.

	Jumpers			
Baud Rate[1] (Hz)	W10	W11	W12	W13
50	In	In	In	In
75	In	Out	Out	Out
110	In	In	Out	Out
134.5	Out	Out	Out	Out
150	Out	Out	In	In
200	Out	In	Out	Out
300	Out	Out	In	Out
600	In	In	Out	In
1200	In	Out	In	Out
1800	In	Out	In	In
2000	Out	In	In	In
2400	Out	In	Out	In
4800	Out	In	In	Out
9600	In	Out	Out	In
19200	In	In	In	Out
38400	Out	Out	Out	In

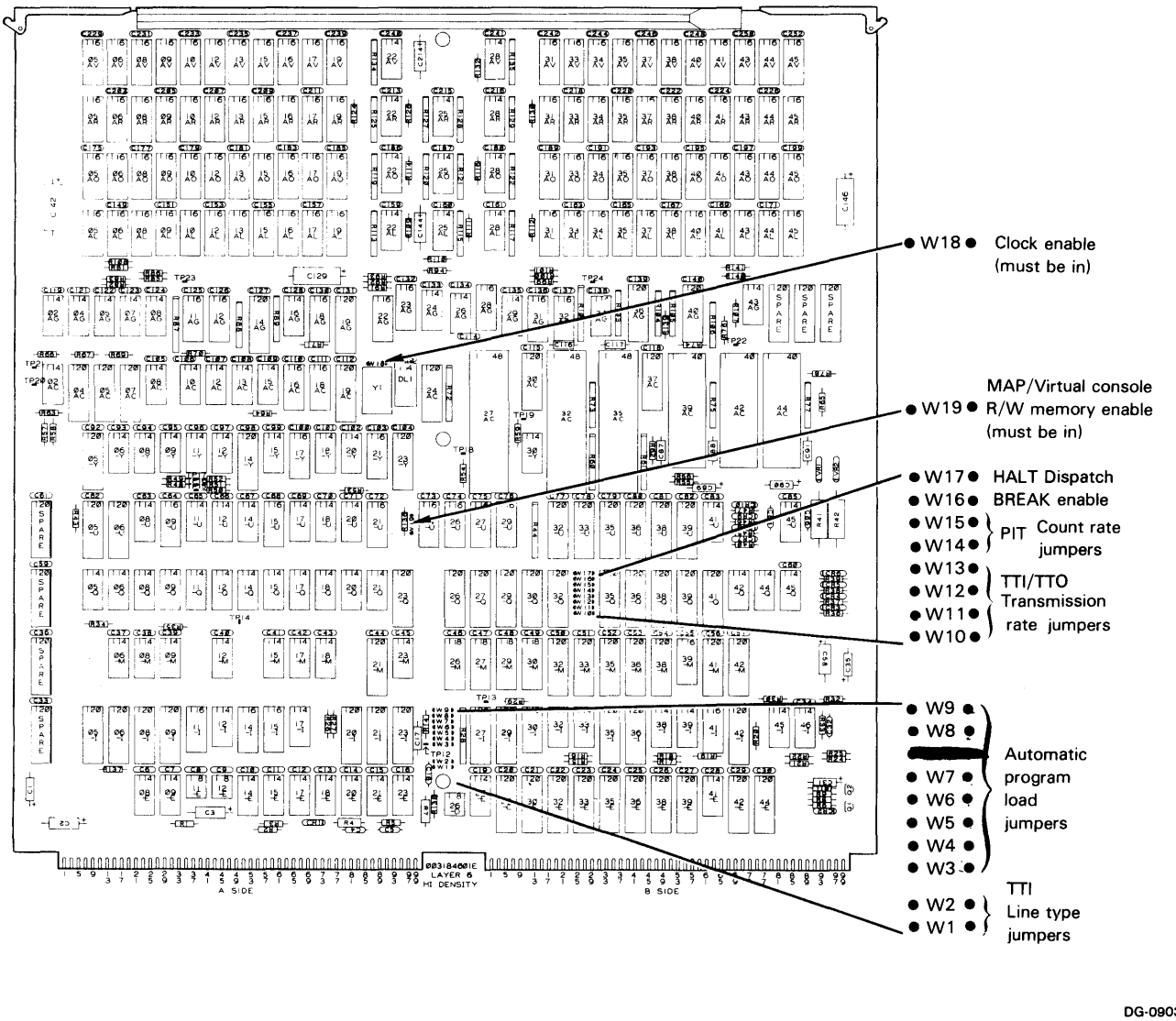
Table 1.29 Asynchronous interface transmission rate, jumper configurations

[1] *50, 75, 110, and 134.5 baud rates have two stop bits, all others have one. The SIO transmits or receives no parity.*

	Jumpers	
PIT Count Rate (KHz)	W14	W15
1 KHz	Out	Out
10 KHz	In	Out
100 KHz	Out	In
1 MHz	In	In

Table 1.30 PIT count rate jumpers

Jumpers W18 and W19 are removed only for DGC manufacturing diagnostic purposes. Removing W18 disconnects the crystal oscillator from the system timing circuitry. Removing W19 disables the MAP and virtual console read/write memory. Otherwise, these jumpers must remain in.



DG-09037

Figure 1.12 Jumper locations

Asynchronous Interface Device Cables

A number of cables are available from Data General Corporation for connecting the S/120 to a serial, asynchronous terminal. Figures 1.13 and 1.14 show typical pinouts, signal names, and cable numbers. For specific noncompliant cables with device cable information, refer to the Installation Data Sheets pertaining to the terminal you are installing.

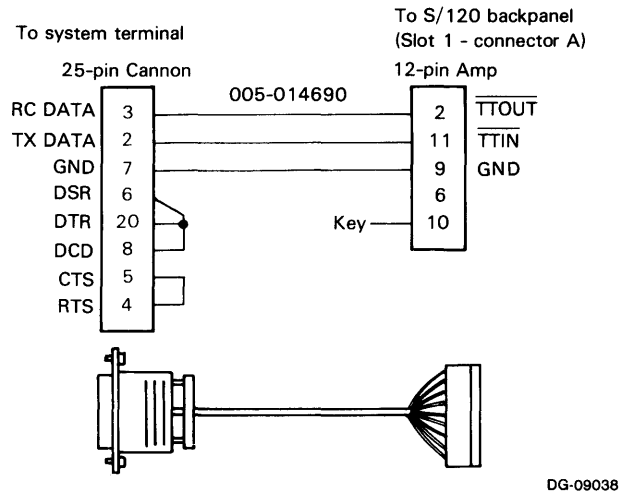


Figure 1.13 Backpanel-to-EIA RS-232-C interface terminal cable (005-014690)

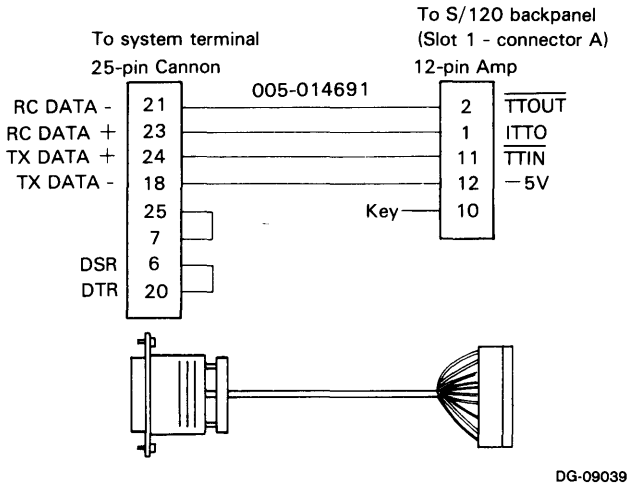


Figure 1.14 Backpanel-to-20mA interface terminal cable (005-014691)

Interfacing: A & B Connectors

The S/120 SPU communicates with other parts of the system via its edge connectors. It receives power and communicates with I/O boards contained in the card cage/chassis through its A and B connectors. The SPU receives power status signals from the power supply through the B connector. It also receives a power line frequency reference signal from the power supply through the A connector. This signal can be used by the real-time clock facility to generate low-frequency interrupts.

The A and B connectors contain 100 pins each. The signals carried on these pins are TTL-compatible unless marked otherwise. Figure 1.15 and 1.16 show the pin assignments of the A and B connectors, respectively.

2	GND	GND	1
4	+5V	+5V	3
6	-5V		5
8			7
10	+15V		9
12			11
14	GND		13
16			15
18			17
20			19
22			21
24			23
26		GND	25
28			27
30			29
32			31
34	GND	GND	33
36			35
38	$\overline{\text{MSKO}}$	GND	37
40	INTA		39
42	DATIB	GND	41
44	DATIA		43
46	$\overline{\text{DS3}}$	GND	45
48	DATOC		47
50	CLR		49
52	STRT		51
54	DATIC		53
56	DATOB		55
58	DATOA		57
60	$\overline{\text{DCHA}}$		59
62	$\overline{\text{DS4}}$		61
64	$\overline{\text{DS5}}$		63
66	$\overline{\text{DS2}}$	GND	65
68	$\overline{\text{DS1}}$		67
70	IORST		69
72	$\overline{\text{DS0}}$		71
74	IOPLS		73
76			75
78			77
80	$\overline{\text{SELD}}$		79
82	$\overline{\text{SELB}}$		81
84		ITTO	83
86		$\overline{\text{TTOUT}}$	85
88	LCLK		87
90			89
92			91
94	$\overline{\text{TTIN}}$	CTS	93
96	MEMOK		95
98	+5V	+5V	97
100	GND	GND	99

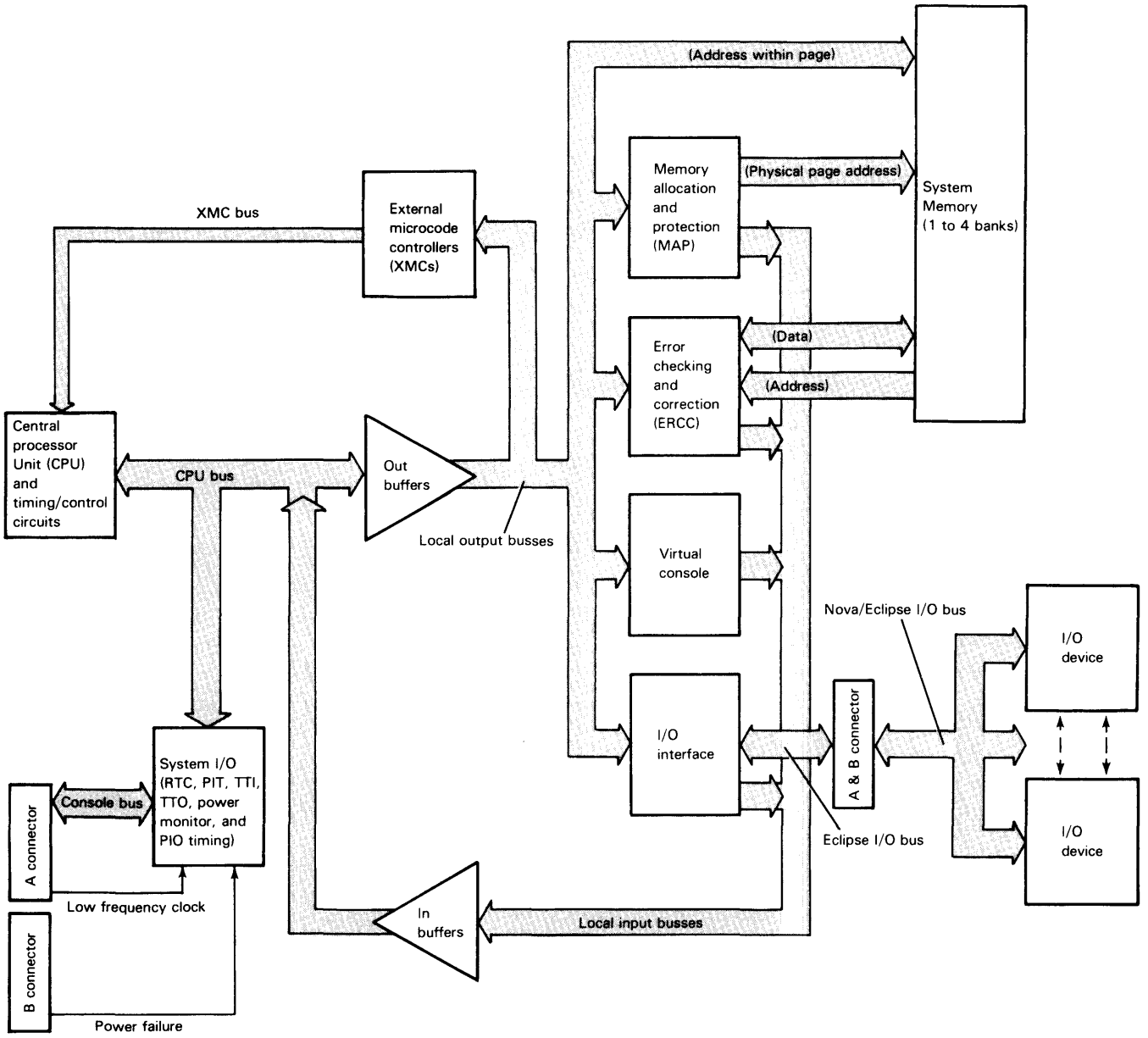
DG-09040

Figure 1.15 Pin assignments, A connector

2	GND	GND	1
4	+5V	+5V	3
6			5
8			7
10			9
12			11
14			13
16			15
18		$\overline{\text{DCHM0}}$	17
20			19
22		GND	21
24			23
26			25
28			27
30		$\overline{\text{INTR}}$	29
32			31
34		DCHO	33
36		$\overline{\text{DCHR}}$	35
38	+5V	DCHI	37
40	$\overline{\text{CONLOCK}}$	GND	39
42		$\overline{\text{RQENB}}$	41
44			43
46	+15V		45
48	$\overline{\text{CONPL}}$		47
50	GND	$\overline{\text{CONLED}}$	49
52	$\overline{\text{CONRSTL}}$	$\overline{\text{CONRSTH}}$	51
54	$\overline{\text{PWRFAIL}}$		53
56	$\overline{\text{DATA14}}$	$\overline{\text{DATA7}}$	55
58	$\overline{\text{DATA11}}$	$\overline{\text{DATA5}}$	57
60	$\overline{\text{DATA8}}$	$\overline{\text{DATA12}}$	59
62	$\overline{\text{DATA0}}$	$\overline{\text{DATA4}}$	61
64	$\overline{\text{DATA13}}$	$\overline{\text{DATA9}}$	63
66	$\overline{\text{DATA15}}$	$\overline{\text{DATA1}}$	65
68	GND		67
70	+12VMEM		69
72	+12VMEM	+12VMEM	71
74	$\overline{\text{EXDCH}}$	$\overline{\text{DATA3}}$	73
76		$\overline{\text{DATA10}}$	75
78		-12V	77
80	GND	PWROK	79
82	$\overline{\text{DATA2}}$	-5V	81
84	+15V		83
86			85
88	+12V	+12V	87
90	+12V	GND	89
92	GND	-5VMEM	91
94	+5VMEM	+5VMEM	93
96	$\overline{\text{MEMDISASTER}}$	$\overline{\text{DATA6}}$	95
98	+5V	+5V	97
100	GND	GND	99

DG-09041

Figure 1.16 Pin assignments, B connector



DG-09043

Figure 2.1 S/120 system architecture

Chapter 2

Theory of Operation

This chapter is devoted to SPU operating theory. It begins with discussions of the ECLIPSE S/120 system architecture and timing and provides an overview of the SPU. Two topics related to the CPU follow: "The CPU Section" and "CPU Support Elements." Discussions of system memory and the NOVA/ECLIPSE input/output interface end the chapter.

S/120 System Architecture

As Figure 2.1 shows, the S/120 is organized around the NOVA/ECLIPSE I/O bus. In addition, the SIO TTI/TTO asynchronous interface communicates with the S/120 system console over the console bus.

The 48-line NOVA/ECLIPSE I/O bus transfers I/O instructions and data or status information between the SPU and I/O device controllers. These transfers occur on 16 bidirectional parallel lines ($\overline{\text{DATA<0-15>}}$). Both programmed input/output (PIO) and data channel (DCH) transfers are controlled by 26 dedicated unidirectional control lines: 20 for PIO, and 6 for data channel. Two request lines, programmed I/O interrupt and data channel service request, allow I/O interfaces to request processor time. Two system control lines, I/O reset and request enable, allow the SPU to initialize all interfaces in the system and synchronize interrupt and data channel requests. Although they are not connected to the SPU, two device priority lines, interrupt and data channel, allow the SPU to service interrupt or data channel requests on a serial priority basis. A summary of NOVA/ECLIPSE I/O bus signals is provided in Appendix A.

System Timing

The S/120 SPU participates in two kinds of data exchange, memory and I/O. Addresses and data are multiplexed on the CPU and local busses; therefore, data exchanges on these busses take place in two phases -- an address phase and a data phase. An address phase/data phase pair corresponds to one CPU cycle, which has a duration of 500 nanoseconds.

The SIO chip detects instructions coded in the I/O format (instruction formats starting 011...) and refers them to its internal devices (real time clock, programmable interval timer, asynchronous interface input, or asynchronous interface output) the MAP unit, error checking and correction unit; or to the I/O interface by the I/O decoder circuitry in the CPU section. The I/O interface produces I/O timing signals based on a 250-nanosecond I/O clock cycle for instructions and data intended for I/O devices connected to the NOVA/ECLIPSE input/output bus.

During the address phase of input/output instructions, the CPU places an encoded version of the actual programmed input/output instruction on the CPU bus. This address (encoded version of the programmed input/output instruction) is applied to the local busses and decoded by the I/O decode circuitry.

During the data phase, detection of the encoded instruction causes the SIO chip to apply timing and control signals to the I/O decode circuitry to effect the data transfer.

Devices requiring more than 500 nanoseconds to send or receive data can extend the data phase of the cycle as needed. To do this, an external device holds the memory bus control signal READY low. The CPU does not initiate another system cycle until READY is asserted high.

SPU Overview

From a functional viewpoint, the System Processing Unit consists of four major sections:

- CPU section
- CPU support section
- System memory
- I/O interface

NOTE: *A major section in this chapter is devoted to each of these elements.*

Descriptions in these sections relate to the Data General logic schematic No. 001-003071.

Figure 2.2 shows the organization of these sections around eight internal busses:

16-bit bidirectional CPU bus (**$\overline{\text{MB}}$<0-15>**),
Six 16-bit unidirectional buffered busses: (**MB<0-15>**), **$\overline{\text{BMB}}$<0-15>**, **$\overline{\text{CMB}}$<0-15>**, **MEMIO<0-15>**, **$\overline{\text{RAMD}}$<0-15>**, and **$\overline{\text{IOOUT}}$<0-15>**)
16-bit latched unidirectional address bus (**A<0-15>**)

The CPU *section* sends and receives addresses and data over the CPU bus to and from the SIO chip and the various local busses. In response to control and timing signals from the CPU control section, addresses are latched in the address latch during CPU address phases.

The CPU *bus* is a multimaster bus; that is, any system component capable of asserting the signals necessary to control the bus cycle specification can be a bus master. In particular, the state machine in the I/O interface section becomes the bus master during data channel operations, as will be described.

The *local bus buffers* convey addresses and data to and from the CPU support section, system memory, and the I/O interface as shown in Figure 2.2. The CPU support elements perform various functions to support CPU operations. The MAP unit supports the memory allocation and protection function of the CPU. The ERCC unit supports the system memory error detection and correction function and the CPU error reporting function. The virtual console contains object code in read-only memory that runs system self-tests and allows the user to access SPU registers and system memory locations directly. The MAP/virtual console read/write memory supports both the MAP unit and the virtual console. For more information about these elements and the functions they perform, see "CPU Support Elements" later in this chapter.

The *system memory* contains 64 Kbit dynamic RAMs and the logic and signal processing circuitry necessary to perform random access read or write operations. The system memory section latches a physical memory address, uses that latched address to select a memory location, and processes timed signals from the CPU section to perform the read or write. The system memory section also contains the logic required to insure that all memory locations are periodically refreshed.

The *address bus* carries addresses from the address latch to the virtual console, the MAP/virtual console read/write memory address selection circuitry, and the I/O decoder circuitry.

The *I/O decoder* contains circuitry that converts the CPU encoded I/O instructions contained in the address latch into standard I/O control signals used by the CPU support elements and the I/O interface. These I/O control signals are applied to both on-board and off-board I/O devices during I/O instructions, by timing and control signals applied by the SIO chip.

The *I/O interface* section contains logic and signal processing circuitry that transfer data or status to and from the external I/O bus along with the appropriate control signals applied by the I/O decoder. This section also handles data channel operations, taking control of the system bus and asserting bus cycle signals as necessary.

The CPU Section

The CPU section, shown in Figure 2.3, contains

- An mE674 16-bit central processing unit
- Three external microcode controller chips (XMCs)
- Timing/control circuitry
- Seven bus buffers

mE674 CPU

The mE674 CPU is a single, large-scale integrated circuit (IC) device that executes the kernel of the ECLIPSE instruction set listed earlier and supports up to 2 megabytes of memory while implementing indirect protection, I/O protection, the LEF instruction, and the emulator trap. The CPU executes 16-bit register-to-register operations in one 500-nanosecond cycle and external bus-to-register moves in two cycles.

Figure 2.4 is a simplified block diagram showing the major functional components of the CPU:

CPU bus transceiver
Four internal busses (A, B, C, and M)
Autonomous prefetch control unit
Program counter (PC) pipeline
Instruction register (IR) pipeline
Register file
Arithmetic/logic unit (ALU) and shifter
MAP status register
Microprogrammed control logic.

The *CPU bus transceiver* forms the interface between the CPU's internal busses and the time-multiplexed external CPU bus. The transceiver drives addresses onto the external CPU bus during the first half (address phase) of a cycle. During the second half (data phase) of the cycle, data is read from, or placed on, the external CPU bus.

The *CPU's four internal busses* enable rapid and flexible transfer of data between the components they connect. For example, the A and B busses allow the simultaneous transfer of two source operands from the register file to the ALU and shifter.

The *autonomous prefetch control unit* causes two words from the instruction stream, beyond the currently executing instruction, to be fetched and placed in the IR pipeline; the corresponding instruction sequence is preserved in the PC pipeline. In cases where instructions alter the program flow or make memory references to the locations of prefetched instructions, the contents of the pipelines are invalidated or *flushed.*

The *register file* includes the four program-accessible accumulators and four general registers accessible only to the microcode.

The *ALU* and *shifter* perform all arithmetic and logic operations under the control of the microprogrammed control logic. Part of the microprogram for the control logic is in microcode resident on the CPU; the remainder is on external microcode controller chips (XMCs).

In addition to supervising the manipulation of data, the microprogrammed control logic within the CPU produces CPU and bus control signals; responds to interrupt requests; handles the microcode transfer protocol; and sends, receives, and processes MAP control signals.

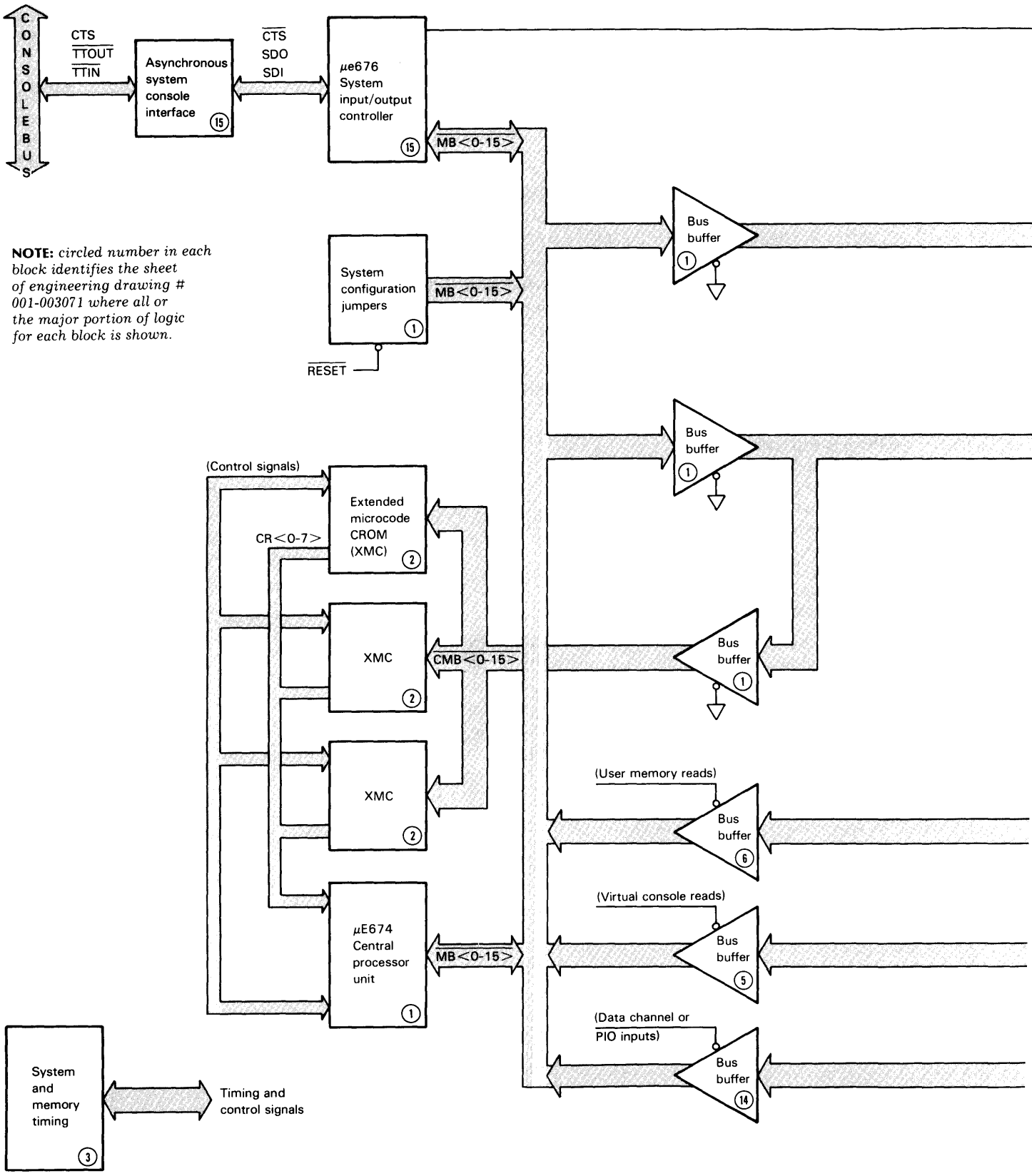
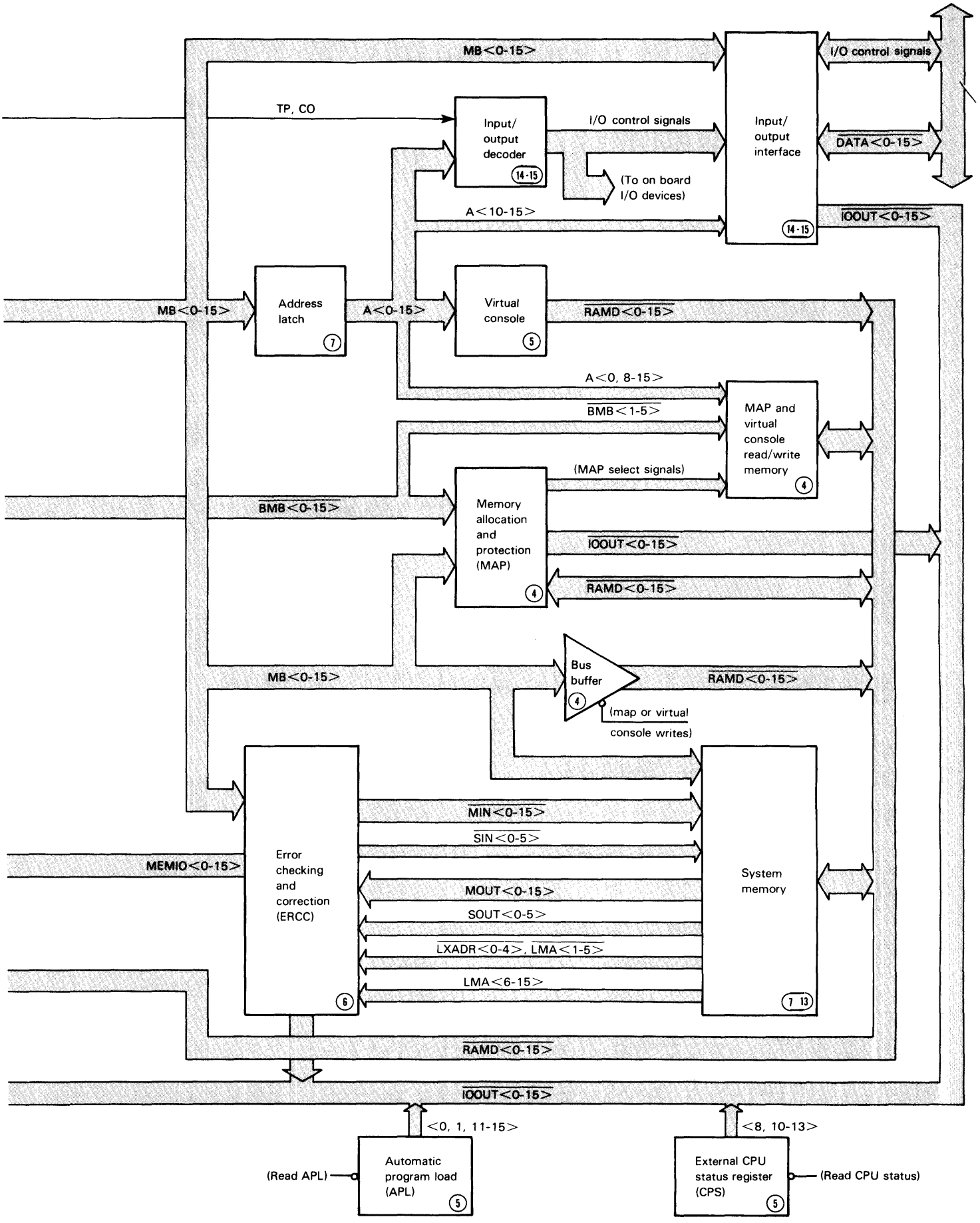
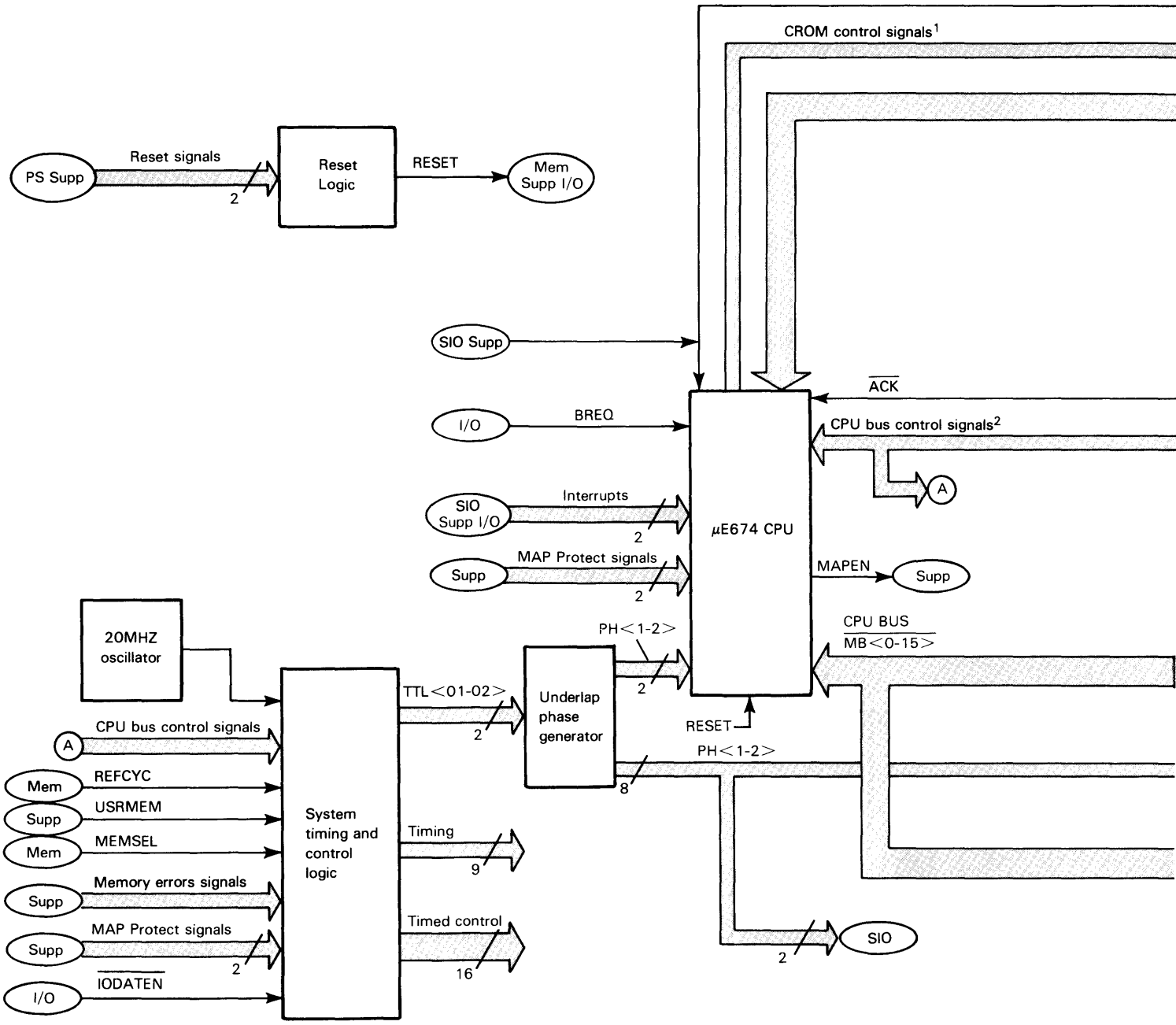


Figure 2.2 System processing unit


MB<0-15>
I/O control signals
NOVA/ECLIPSE I/O BUS
TP, CO
Input/ output decoder
14-15
I/O control signals
(To on board I/O devices)
Input/ output interface
DATA<0-15>
A<10-15>
IOOUT<0-15>
14-15
MB<0-15>
Address latch
7
A<0-15>
Virtual console
5
RAMD<0-15>
A<0, 8-15>
BMB<1-5>
MAP and virtual console read/write memory
4
(MAP select signals)
BMB<0-15>
Memory allocation and protection (MAP)
4
IOOUT<0-15>
RAMD<0-15>
MB<0-15>
Bus buffer
4
RAMD<0-15>
(map or virtual console writes)
MIN<0-15>
SIN<0-5>
MEMIO<0-15>
Error checking and correction (ERCC)
6
MOUT<0-15>
SOUT<0-5>
LXADR<0-4>, LMA<1-5>
LMA<6-15>
System memory
7 13
RAMD<0-15>
IOOUT<0-15>
<0, 1, 11-15>
<8, 10-13>
(Read APL)
Automatic program load (APL)
5
External CPU status register (CPS)
5
(Read CPU status)

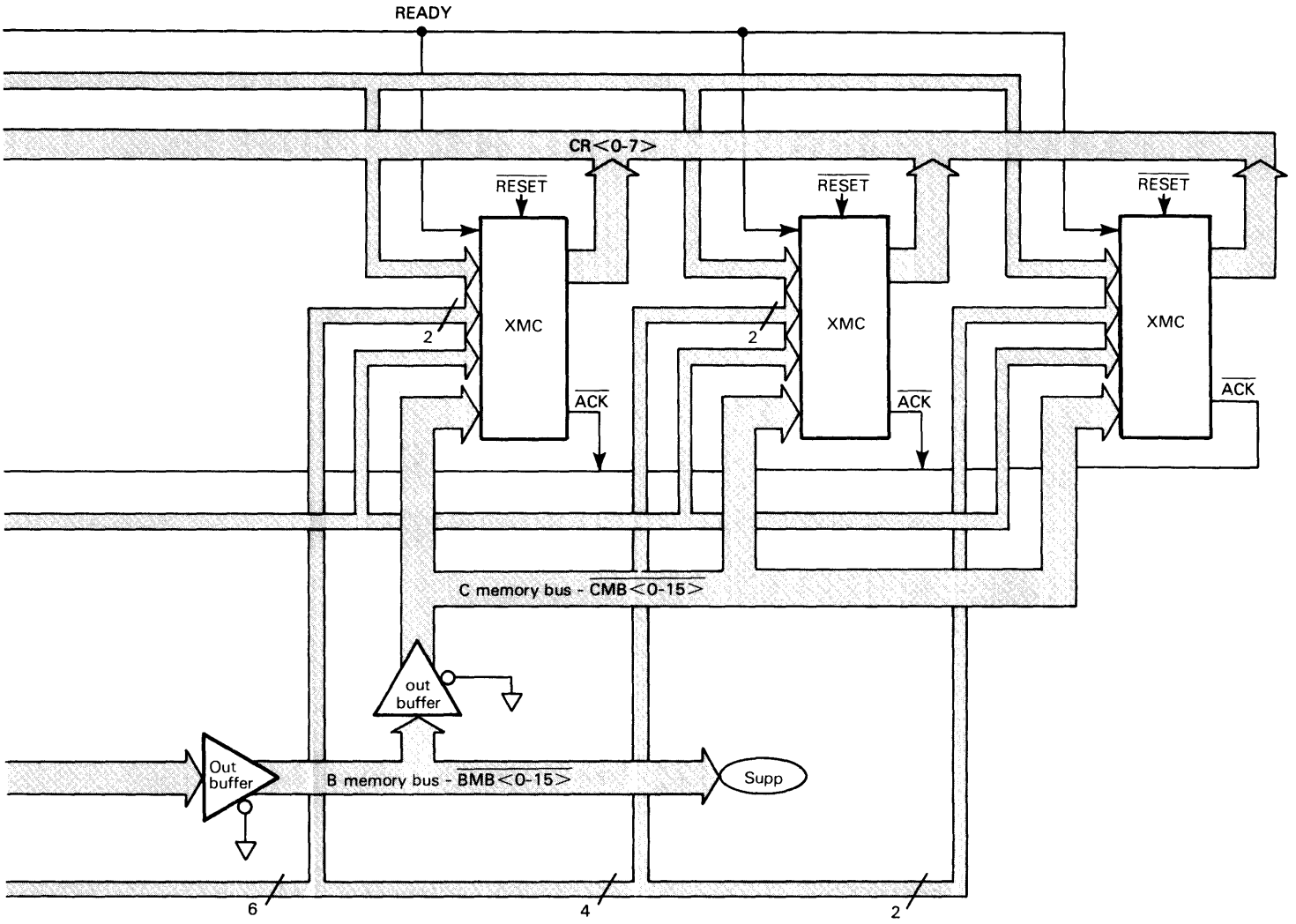


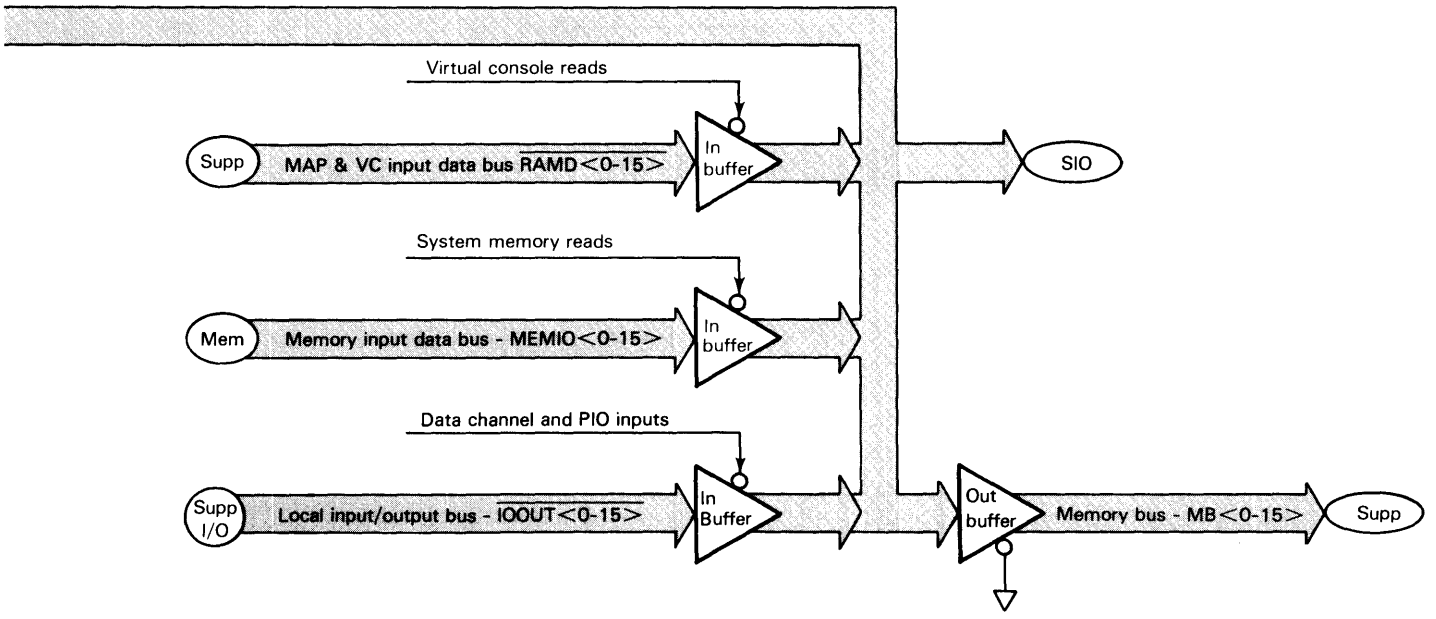
[1]*CROM control signals*
- *REQ*
- *STATUS*
- *PIPE*
- *FETCH*

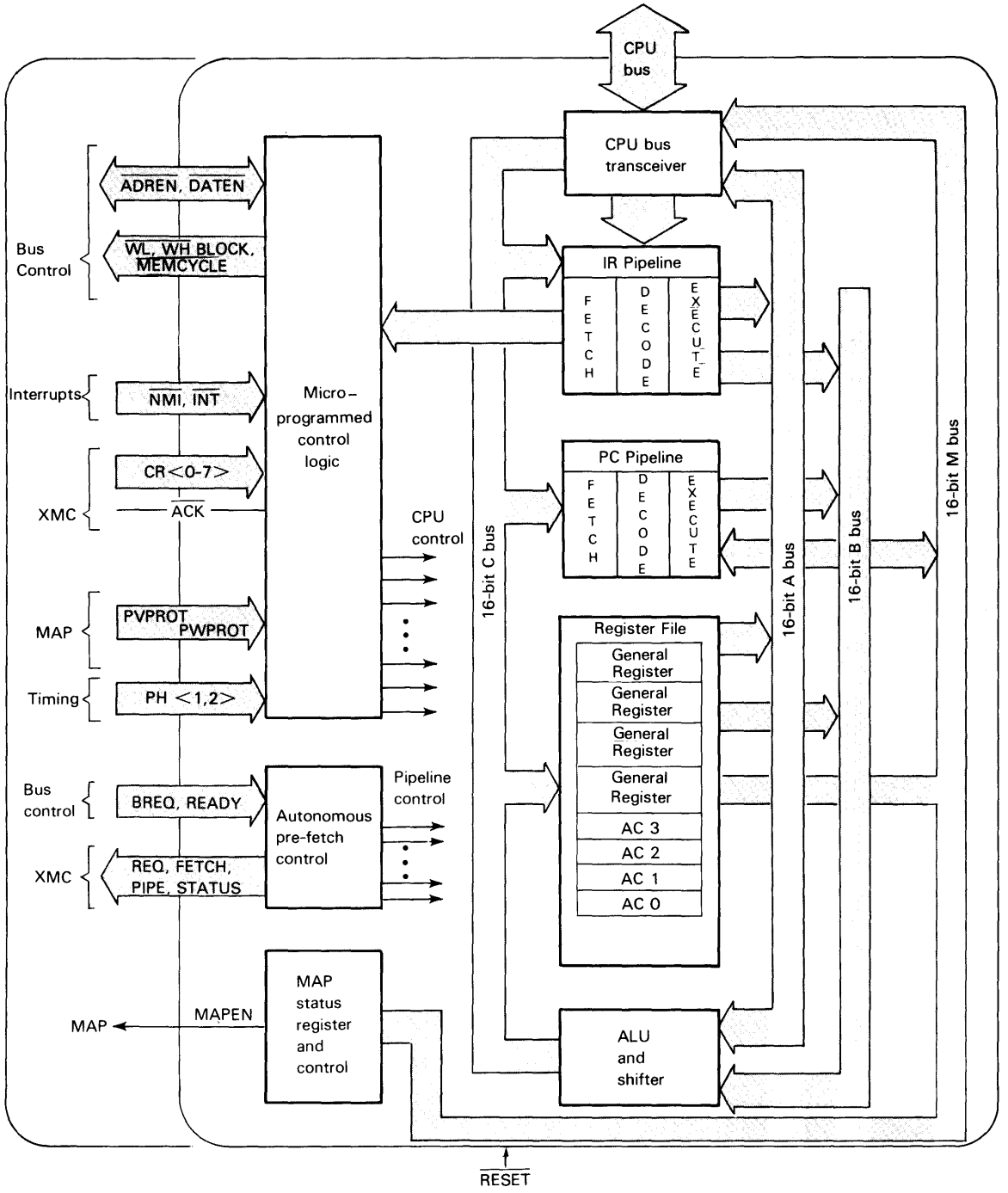
[2]*Bus control signals*
- ADREN
- DATEN
- *MEMCYC*
- WH
- WL

DG-09046

Figure 2.3 Central processing unit


READY
CR<0-7>
RESET
RESET
RESET
XMC
XMC
XMC
2
2
ACK
ACK
ACK
C memory bus - CMB<0-15>
out buffer
Out buffer
B memory bus - BMB<0-15>
Supp
6
4
2


Virtual console reads
Supp
MAP & VC input data bus RAMD<0-15>
In buffer
SIO
System memory reads
Mem
Memory input data bus - MEMIO<0-15>
In buffer
Data channel and PIO inputs
Supp I/O
Local input/output bus - IOOUT<0-15>
In Buffer
Out buffer
Memory bus - MB<0-15>
Supp



DG-08301

Figure 2.4 Internal CPU

Microcode Controller Chips

Each external microcode controller chip (XMC) contains a decode programmed logic array (PLA) that can decode up to 64 macroinstructions. When the PLA decodes an instruction contained on the XMC, it transfers control to the XMC's microcode controlling logic. The XMC then sends microcode for the microinstruction to the CPU.

The CPU accepts 16-bit external microcode instructions from the microcontroller chips (XMCs) via a dedicated 8-bit time-multiplexed bus (**CR<0-7>**). In addition, the CPU and XMCs use five control signals for microcode transfer protocol. The protocol allows conditional branching that depends on the result of CPU microcode execution.

The S/120 SPU board contains three XMCs. Each contains microcode for an instruction set that supplements the kernal of ECLIPSE instructions resident on the mE674, as well as microcode for the floating-point instruction set. This enables the CPU to execute the entire ECLIPSE instruction set listed earlier, including the floating-point extension.

Each XMC monitors the system bus to maintain a duplicate of the CPU's macroinstruction pipeline. This allows the CPU and multiple XMCs to decode macroinstructions simultaneously. As a result, only one cycle is required to obtain the microcode for an instruction not contained on the mE674.

Timing

System and memory timing circuitry consists of a state machine (an address counter, two 32 x 8 PROMs, and a 16-bit storage register) and cycle and timing control logic. A 20-MHz crystal-controlled oscillator and a phase-clock generator are associated with these. The oscillator clocks the state-machine address counter at a rate of 20-MHz. The phase-clock generator produces the clocking signals for the CPU, SIO, and XMC chips. Each SPU cycle is 500 nanoseconds in length.

Figures 2.5 and 2.6 show the timing signals produced by the timing state machine for a normal SPU cycle and for a normal cycle with an appended memory correction cycle. Also shown are the 20-MHz reference signal and the clock signals produced by the phase-clock generator.

The timing signals are listed and described below.

TTL<01-02> Two synchronizing signals for the SPU elements. These signals also input to the phase-clock generator which produces the two-phase non-overlapping clock signals for the CPU, SIO, and XMCs: **PH1<A-D>** and **PH2<A-D>**.

2XCLK Clocks the ECLIPSE input/output interface timing state machine and the system input/output (SIO) controller.

CYCTST Tests for a system bus operation to be performed during this cycle.

MEMSTRT Memory start-up synchronizing signal.

HWRTPLS Write-enable signal for the virtual console read/write memory, and the memory allocation and protection memory.

WRTEN Combines with various signals to control the write pulse to system memory.

$\overline{\text{CNTRST}}$ Initiates each timing cycle and resets the state-machine address counter. This signal also starts a memory-correction cycle when required and synchronizes data channel service requests.

ECCCOR Places the error detection and correction unit in correction mode.

ECCGEN Combines with a system memory write-cycle control signal to place the error detection and correction unit in generate mode.

ECCWRT, $\overline{\text{ECCWRT}}$ Combines with various signals to control system memory-write operations during a correction cycle.

ECCLATCH Latches data read from, or being written to, system memory in the error detection and correction unit.

$\overline{\text{CORCLR}}$ Clears the correction cycle control flip-flop and the multiple-bit error detection during a correction cycle.

The state machine outputs are always present after powerup.

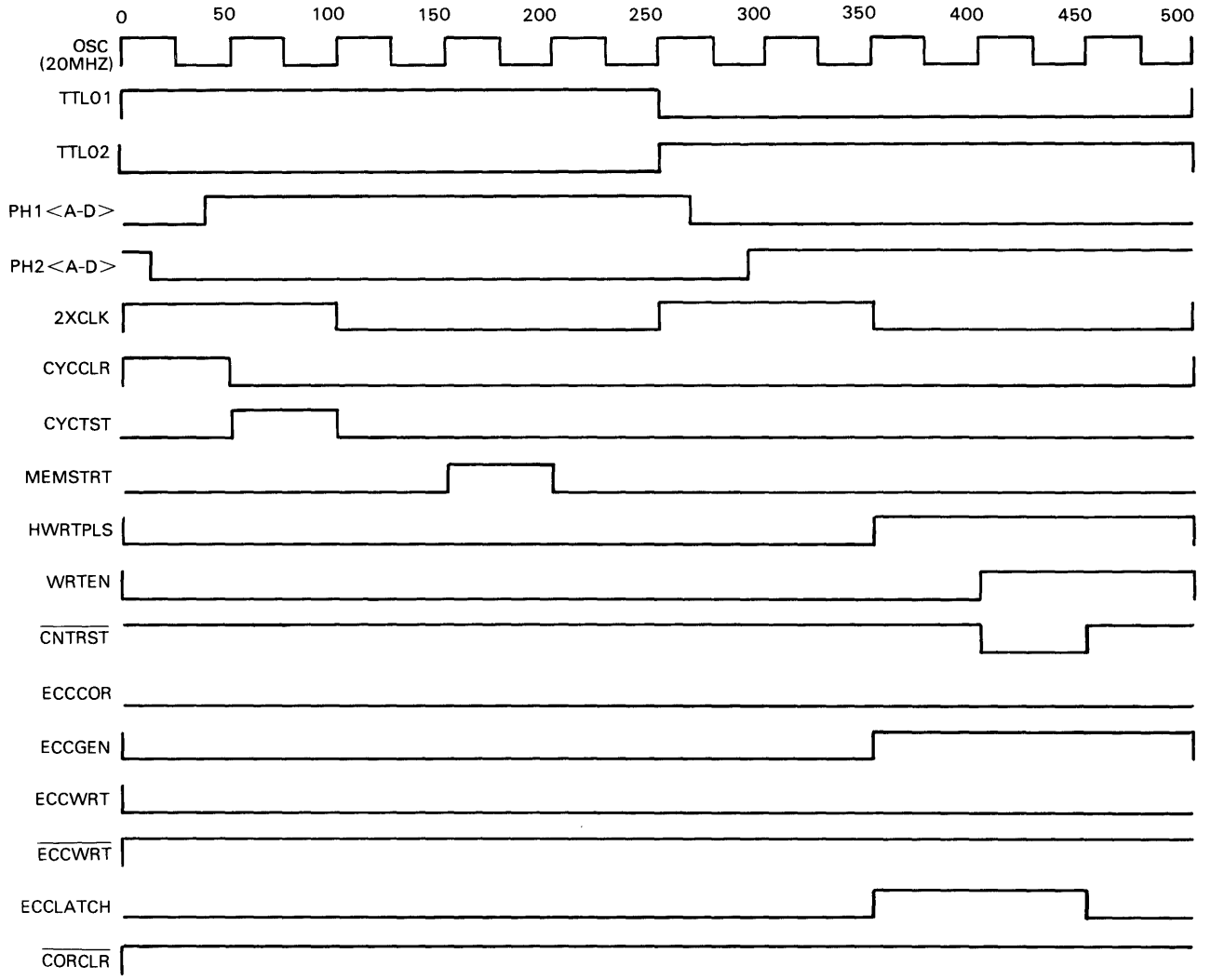


Figure 2.5 System and memory timing state machine signals (normal cycle)

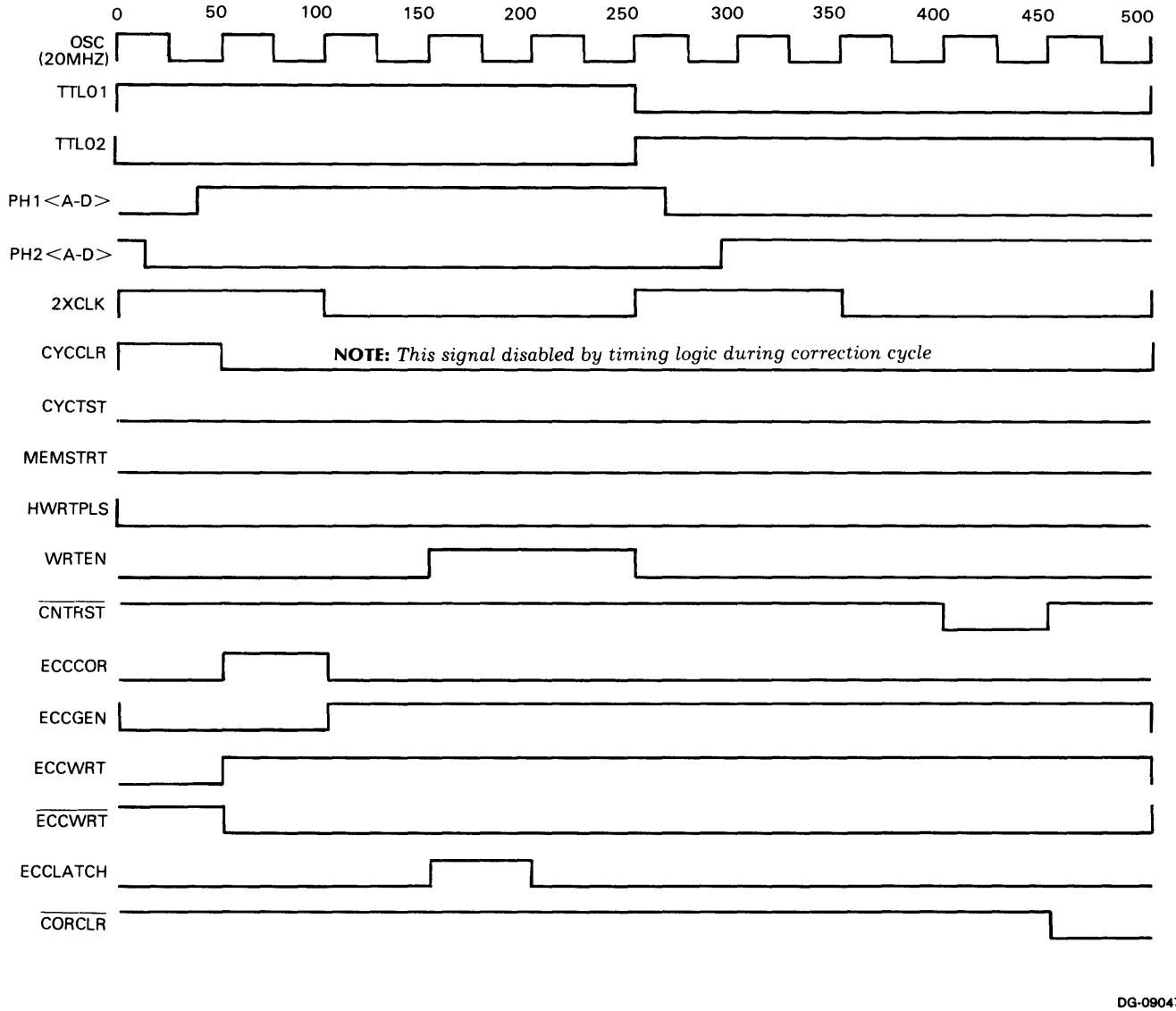


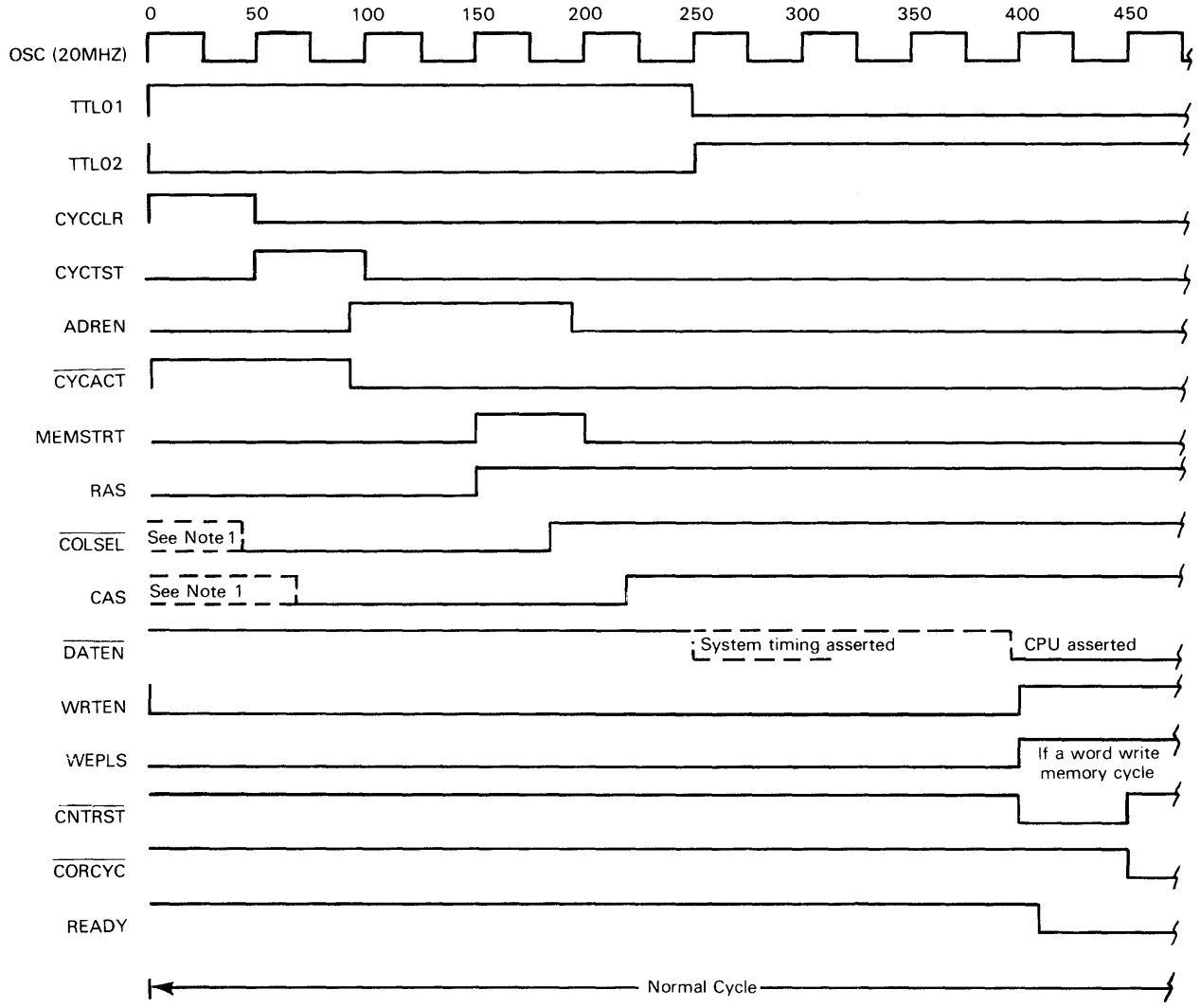
Figure 2.6 System and memory timing state machine signals (correction cycle)

Figure 2.7 shows the input to, and output from, the cycle and timing control logic and the CPU-sourced bus control signals **ADREN** and **DATEN**. These timed control signals are listed and described below.

CORCYC, CORCYC — Provide the control necessary to perform the error correction cycle.

CYCACT, CYCACT — Enable the decoding of on-board I/O device selection and, together with other signals, enable data transfers between the CPU bus and the local SPU busses.

READY — Signals that data is available for transfer to the receiving element. Asserts low when another cycle time is required to complete the data phase. This can be caused by a memory-refresh cycle that is in progress when the bus master access system memory, a memory error detected during a system memory-read operation, or a system memory-write operation is writing only one byte. (A one-byte write operation first reads the accessed location, corrects any single-bit error, and then writes the byte being retained and the byte being written. Refer to "System Memory" and "Error Checking and Correction Unit" in this chapter for additional information on a one-byte write operation.)



NOTE: 1. COLSEL *and* CAS *are still asserted at this time if previous cycle was a system memory access.*

DG-09048

Figure 2.7 Timed control signals (normal cycle with appended correction cycle)

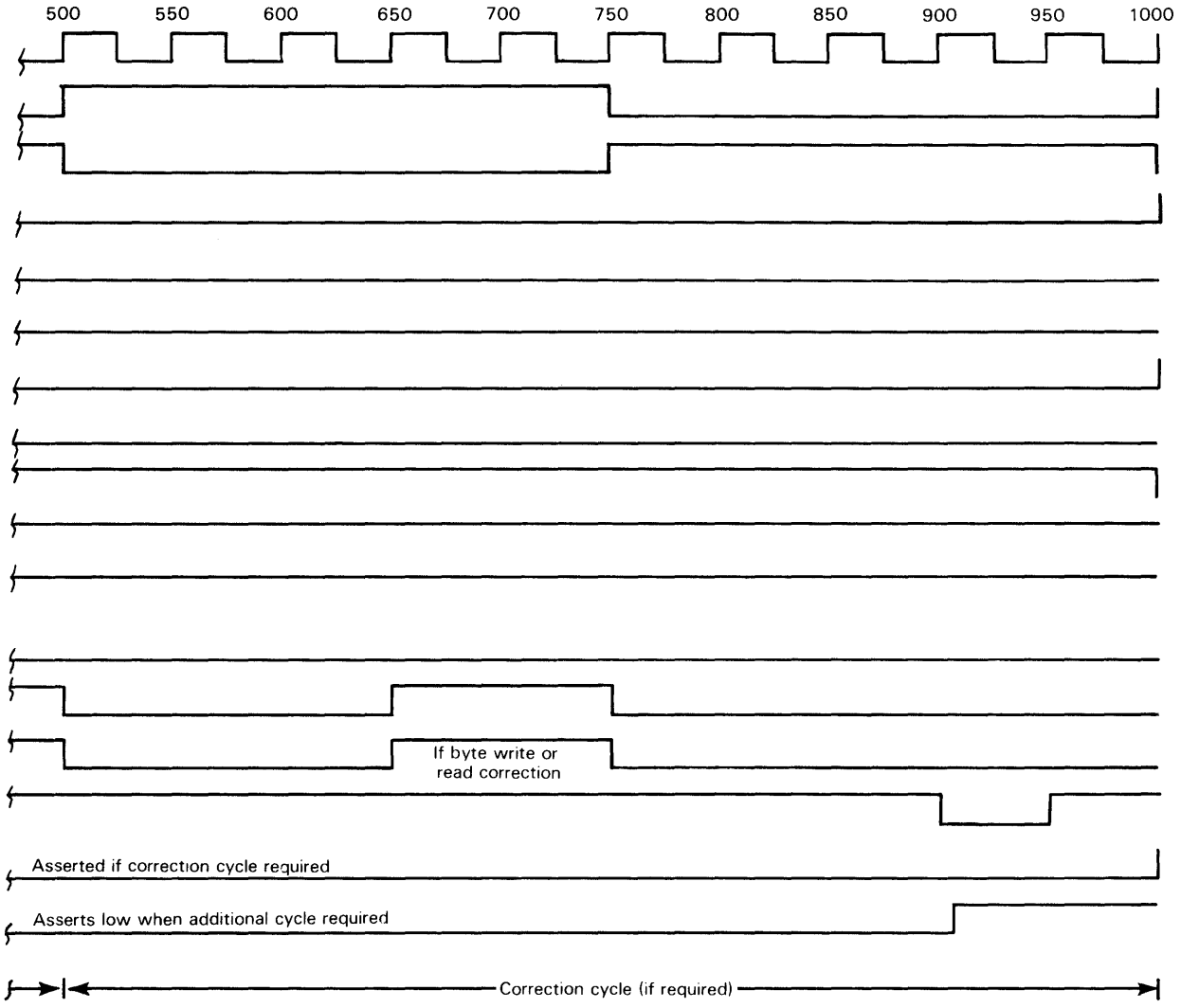
RAS Latches the eight row-address bits into the selected system memory-bank RAM chips and performs an enable to those chips.

COLSEL Selects the eight-column address bits to be applied to the system memory RAM chips.

CAS Latches the eight column-address bits into the selected system memory-bank RAM chips and performs a select to those chips.

WEPLS Selects the read or write mode for the system memory RAM chips. When **WEPLS** asserts high, the write mode is selected and data is latched into the RAM chips on the falling edge of this signal.

DATEN Data phase control signal which combines with other timing and control signals to enable data transfers between the CPU bus and its local busses. This signal is produced by the SPU element that is the current source of the data. System timing produces this signal during all read bus cycles, unless a system element within the SIO chip is accessed.



$\overline{\text{CYCCLR}}$ Clears the cycle control logic. This signal is disabled during the error correction cycle.

The presence of these signals depends on **TTL01** and the particular operation being performed. For example, the system memory signals **RAS, $\overline{\text{COLSEL}}$, CAS,** and **WEPLS** does not occur unless a bus master asserts **$\overline{\text{ADREN}}$** or the memory unit asserts **$\overline{\text{REFCYC}}$** indicating a memory operation is required.

Other timed control signals derive from these two sets of signals in logical combination with system control signals. These timed control signals are described later in this chapter.

Control

The control logic produces three types of signals:

- Latched bus control signals
- Bus grant signal
- Bus buffer control signals

Latched Bus Control Signals

When the timing signals **TTL01** and **ADREN** are asserted, the control logic produces a signal which latches

memory control signals: **MEMCYCLE**, **WH**, **WL**
map enable signal: **MAPEN**
map protection signals: **RAMD0** and **RAMD1**.

RAMD0 and **RAMD1** specify that the logical page being accessed is write protected and validity protected, respectively. The values of the signals latched during the address phase of an operation remain constant during an entire bus cycle.

When both latched **LWH** and latched **LWL** are active, the signal **WRTCYC** is produced. When only one of these signals is active (specifying only one byte is to be written) and the system memory is selected, the signal **WRTCYC** is produced when the timed signal **ECCWRT** is active. **ECCWRT** is active during a correction cycle. **WRTCYC** determines the direction of system memory and programmed input/output data transfers.

Bus Grant Signal

As previously described, the CPU bus is a multimaster bus. The CPU either retains control of the bus or relinquishes it to the I/O interface for data channel transfers. The I/O interface can take control of the CPU bus when

- **BLOCK** is not asserted by the CPU. The CPU asserts **BLOCK** when it is engaged in an operation requiring uninterrupted bus accesses, for instance, during execution of an **ISZ** instruction.
- **READY** is asserted. **READY** is pulled low by a device that has not completed the current data transfer.

If both conditions are met, the timing signal **TTL02** issues **BUSOK**, the signal that gives control to the I/O interface to perform a data-channel transfer. "NOVA/ECLIPSE I/O Interface" in this chapter provides a detailed description of how the I/O interface requests control of the CPU bus.

Bus Buffer Control Signals

Addresses and data contained on the CPU bus (**MB<0-15>**) are applied to the memory bus (**MB<0-15>**), the B memory bus (**BMB<0-15>**), and the CROM bus (**CMB<0-15>**) at all times. Their bus buffers are always enabled. (Refer to Figures 2.2 and 2.3.)

Data contained on the memory bus (**MB<0-15>**) is applied to the MAP and virtual console memory data bus (**RAMD<0-15>**) during the data phase of a *Load Map* instruction or virtual console write operation when the timing signal **HWRTPLS** is asserted. (Refer to Figures 2.2, 2.3, 2.12, and 2.16.)

This bus buffer can be enabled to apply the high-order byte (**MB<0-7>**), the low-order byte (**MB<8-15>**), or both bytes to their respective **RAMD** bits. **HBEN** applies the high-order bits, while **LBEN** applies the low-order bits. The latched bus-enable signals, **LWH** and **LWL**, determine the generation of these two signals. During a *Load Map* instruction the CPU asserts both **LWH** and **LWL**.

This bus buffer is enabled as follows.

During a *Load Map* instruction by the following signals:

MAPSEL
A1 = 1
DATEN
LWH and **LWL** both = 1
HWRTPLS

and during writes to virtual console read/write memory by the following signals:

MEMCYC
A0 = 1
A1 = 0
LWH or **LWL** or both = 1
DATEN
HWRTPLS

Data read from the virtual console program memory or read/write memory (**RAMD<0-15>**) is applied to the CPU bus (**MB<0-15>**) during the data phase of virtual console read operations as follows.

During virtual console program memory accesses by the following signals:

LMEMCYC
A0 = 1
A1 = 1
READ
LWH and **LWL** both = 0
DATEN

and during virtual console read/write memory accesses by the following signals:

LMEMCYC
A0 = 1
A1 = 0
READ
LWH and **LWL** both = 0
DATEN

Data from the on-board or off-board I/O devices (**IOOUT<0-15>**) is applied to the CPU bus (**MB<0-15>**) during the data phase of input programmed input/output instructions or data channel request transfers. This bus buffer is enabled as follows.

During the data phase of an input programmed input/output instruction by the following signals:

TP
CO = 0
WRTCYC = 0

and during the address phase of data channel transfer (input/output) and data phase of data channel input transfer by the following signal:

DCHDRV

Data read from system memory (**MEMIO<0-15>**) is applied to the CPU bus (**$\overline{\text{MB<0-15>}}$**) during the data phase of a memory read operation, provided no validity protection fault has been detected by the MAP unit. This bus buffer is enabled by the following signals.

LMEMCYC
$\overline{\text{A0}}$=0
DATEN
READ
$\overline{\text{LVALERR}}$

System Memory Read/Write Operations

A flow diagram and a basic timing diagram for memory read/write operations are shown in Figures 2.8 and 2.9, respectively. During the address phase, the bus master asserts **$\overline{\text{ADREN}}$**. The bus master is either the CPU, for normal memory operations; or the I/O interface state machine, for data channel operations. The bus master then places the address on the memory bus. During the address phase, the bus master asserts **$\overline{\text{WH}}$**, **$\overline{\text{WL}}$**, or both—depending on whether the memory operation is write high byte, write low byte, or write word. If no write signal is asserted, the operation is a memory read. Data channel write (input) operations always write a word.

During the address phase, the logical-page memory address bits are applied to the MAP unit for translation to a physical-page memory address under two conditions:

MAP unit is enabled

or

a CPU memory operation addresses logical page 31 and the MAP is not enabled

When neither of these conditions is not met, the logical-page address is applied to the memory unit as the physical page address. During address translation (if enabled), the MAP unit asserts memory-protection signals if a write protected or validity protected logical page is addressed.

The address, memory control signals, and memory protection fault signals are latched in the address and control latch when the high-to-low transition of **TTL01** occurs.

During the data phase, the device supplying the data applies it to the memory bus and asserts **$\overline{\text{DATEN}}$**. When the falling edge of **PH2<A-D>** occurs, the device receiving the data latches it.

If the error checking and correction facility is enabled, single-bit memory errors detected during a read operation are corrected and written into memory before the data is applied to the system bus. During a byte write operation, the addressed word is first read from memory and any single-bit error is corrected. Then the byte to be written and the byte to be retained are written into memory.

When a MAP fault occurs during a mapped memory operation (**$\overline{\text{LMAPEN}}$**), the MAP logic asserts **LVALERR** and/or **LWRTINH**. If a memory-read operation to a validity-protected logical page is attempted, **LVALERR** prevents data read from memory being applied to the CPU bus (**$\overline{\text{MB<0-15>}}$**). If a memory-write operation to a write-protected logical page is attempted, **LWRTINH** prevents data being written into memory.

Refer to "System Memory Unit," "Error Checking and Correction Unit," and "Memory Allocation and Protection" in this chapter for more detailed descriptions of system memory operations.

Extended Memory Cycles

A device can request as much additional time as needed to complete a memory cycle by pulling the **READY** line low. As Figure 2.10 shows, the system and memory timing logic pulls the **READY** line low. This is because the memory unit received a CPU memory operation request during a refresh operation that was pended during a data channel operation.

READY is also pulled low to extend a data phase when the error checking and correction unit is enabled, and a memory error is detected during a read operation. In addition, a one-byte memory write operation also pulls **READY** low to extend the data phase. This allows the byte of memory data that is to be retained to be checked for errors and corrected. (Refer to "System Memory" in this chapter for a more detailed description of these operations.)

At the end of each data phase, the CPU samples the state of **READY**. If **READY** is low, the CPU extends the memory for another complete 500-nanosecond cycle. If **READY** is still low when the second cycle ends, the current memory operation is extended for yet another cycle. Otherwise, the CPU can begin a new memory operation. If the CPU is the data source, the data is available for the entire extension of the memory cycle. However, the memory does not use the data until the refresh and/or error correction cycles, if required, are completed. If the memory is the data source, the data is placed on the bus during the first data phase after the refresh and/or error correction cycles, if required, have been performed.

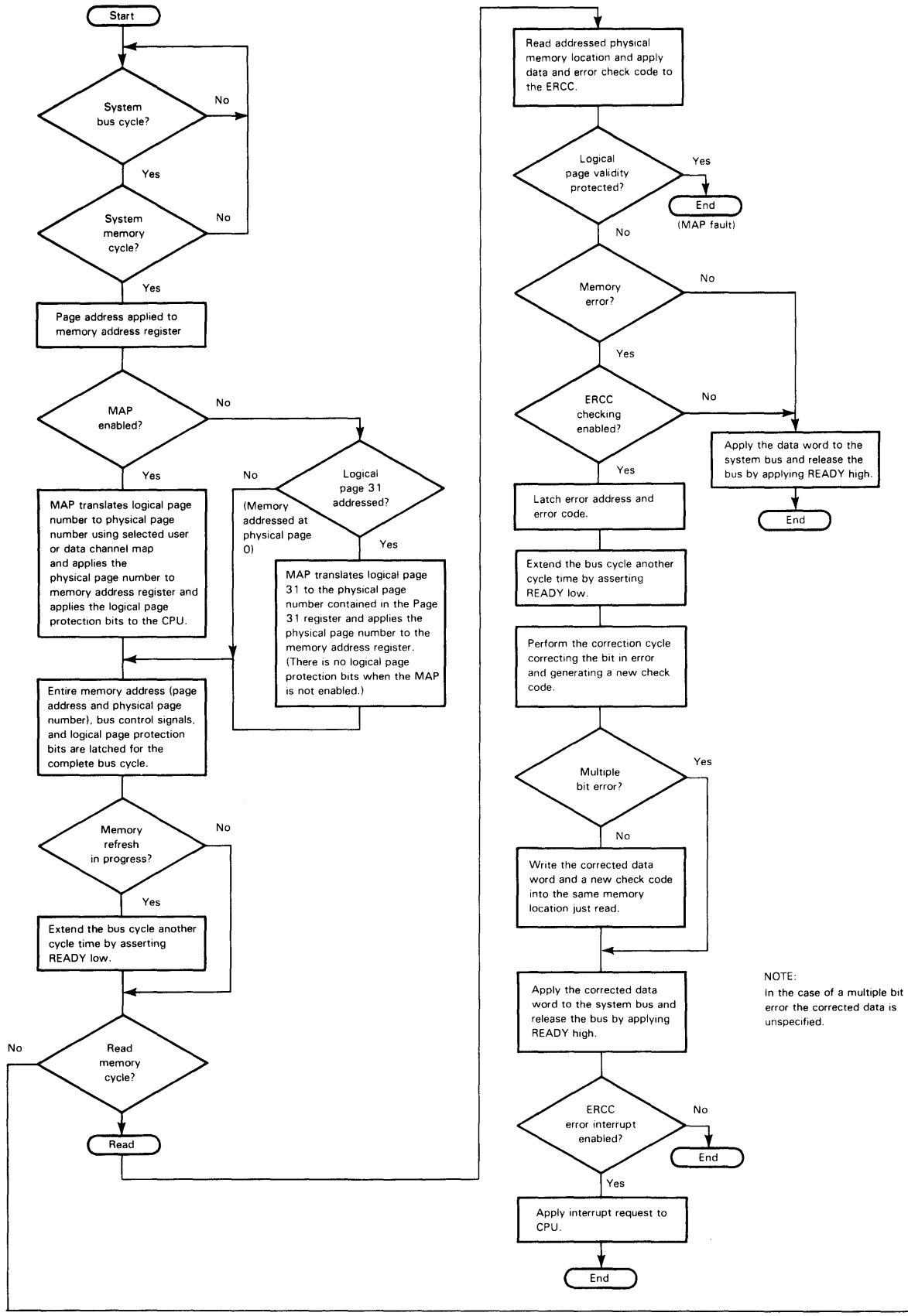
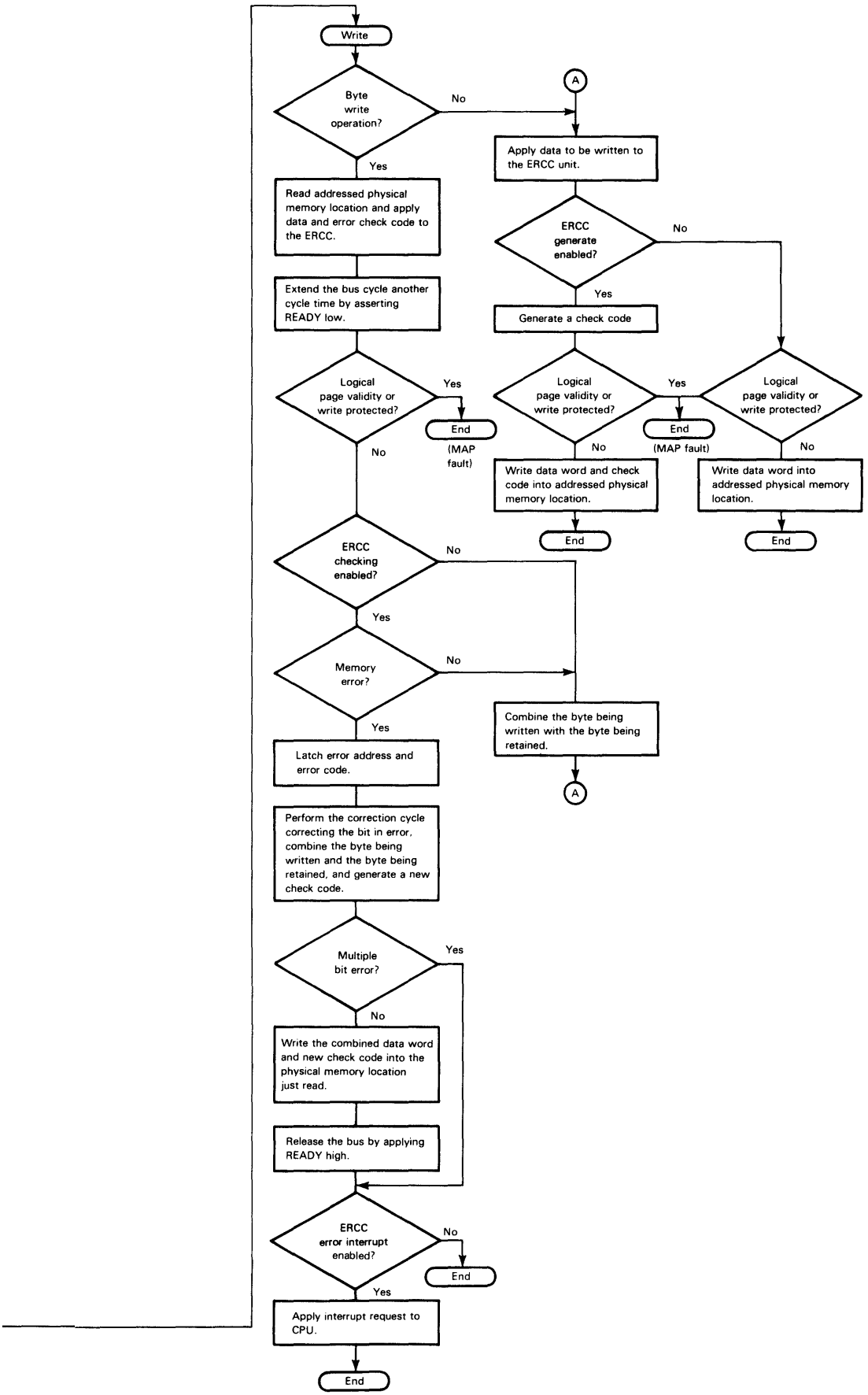


Figure 2.8 System memory read/write


Write
Byte write operation?
No
Yes
A
Apply data to be written to the ERCC unit.
Read addressed physical memory location and apply data and error check code to the ERCC.
ERCC generate enabled?
No
Yes
Extend the bus cycle another cycle time by asserting READY low.
Generate a check code
Logical page validity or write protected?
Yes
End
(MAP fault)
No
Logical page validity or write protected?
Yes
End
(MAP fault)
No
Logical page validity or write protected?
No
Write data word and check code into addressed physical memory location.
End
Write data word into addressed physical memory location.
End
ERCC checking enabled?
No
Yes
Memory error?
No
Yes
Combine the byte being written with the byte being retained.
A
Latch error address and error code.
Perform the correction cycle correcting the bit in error, combine the byte being written and the byte being retained, and generate a new check code.
Multiple bit error?
Yes
No
Write the combined data word and new check code into the physical memory location just read.
Release the bus by applying READY high.
ERCC error interrupt enabled?
No
End
Yes
Apply interrupt request to CPU.
End

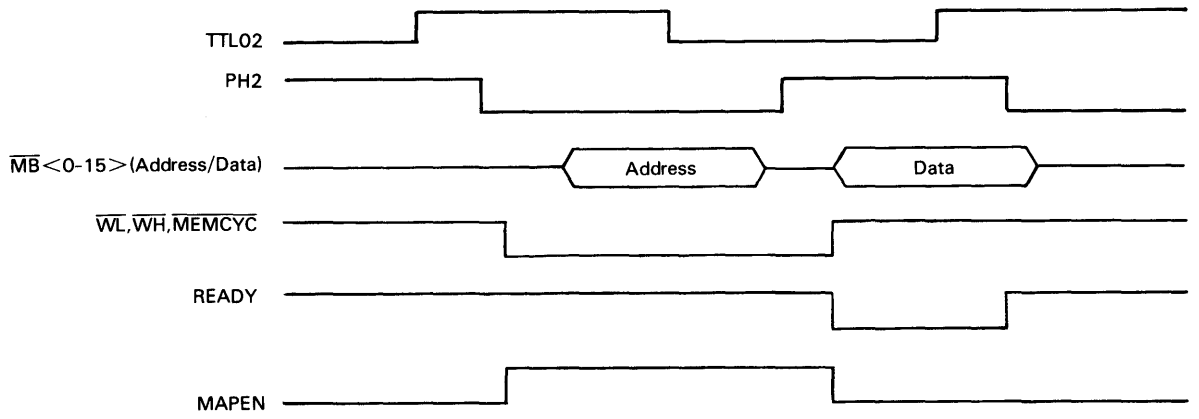


Figure 2.9 Memory bus control signals

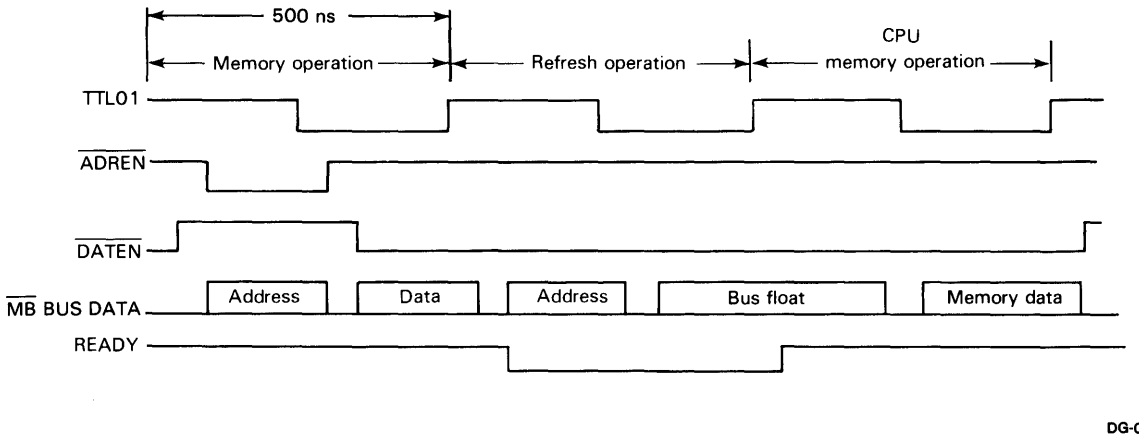


Figure 2.10 Extended memory cycle

CPU Support Elements

The CPU support elements shown earlier in the CPU block diagram (Figure 2.2) include the following:

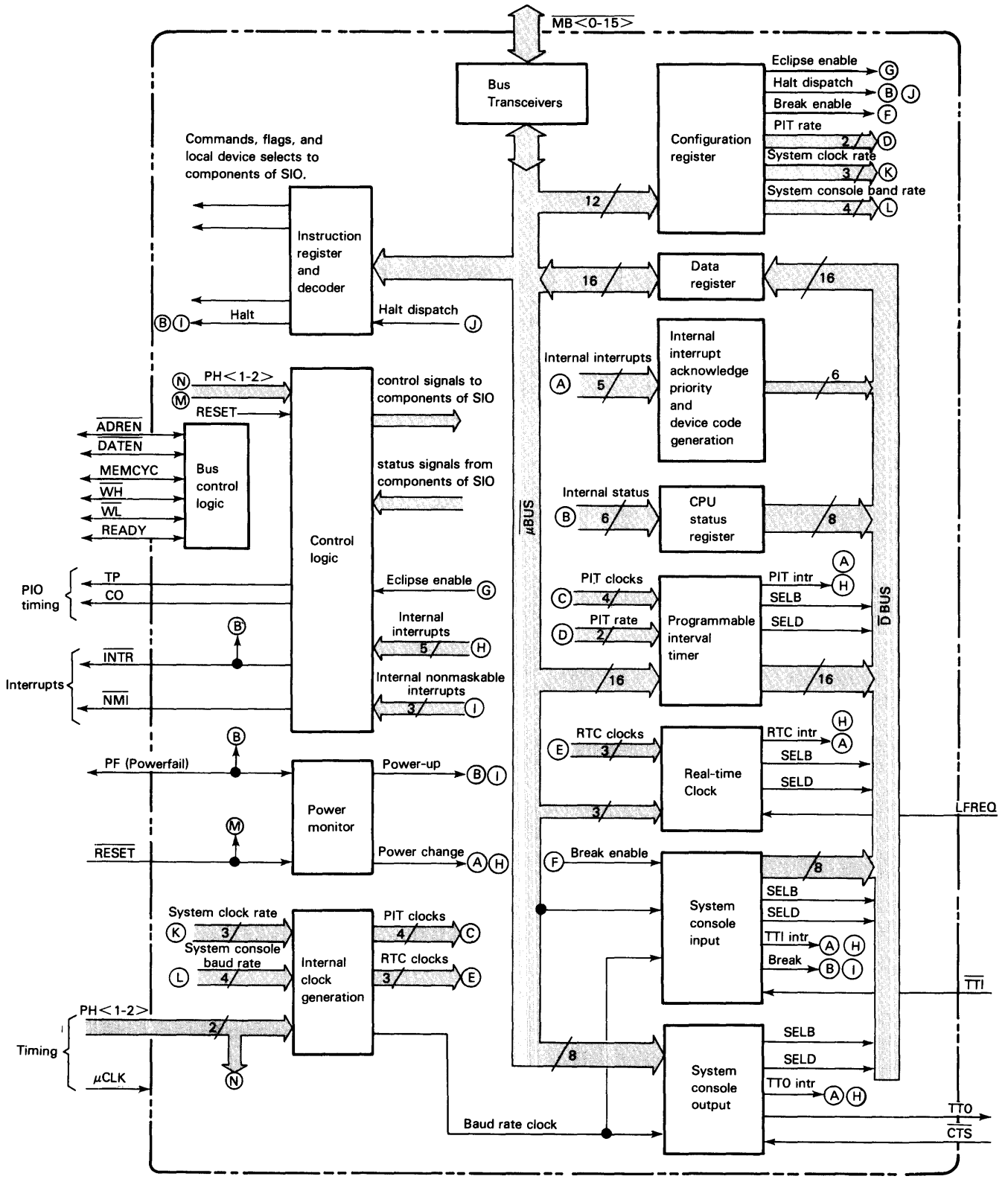
- mE676 SIO chip
- Address latch
- Input/output decoder
- Virtual console
- Three registers: SIO, APL, and the external CPU status
- Error checking and correction facility
- Memory allocation and protection unit (MAP)

The mE676 SIO Chip

The mE676 SIO is a single large-scale integrated circuit (IC) device that supports CPU operations. It contains five devices that support the CPU. These devices are listed below and illustrated in Figure 2.11.

power monitor (CPU device)
programmable interval timer (device 43)
real time clock (device 14)
asynchronous interface (TTI — device 10, TTO — device 11)
part of the CPU status register

The SIO chip also provides timing and control signals to the I/O decoder, I/O interface, and system and memory timing during programmed input/output operations. These two signals, **TP** and **CO**, are detailed in the "NOVA/ECLIPSE I/O Interface" section of this chapter.



DG-09093

Figure 2.11 mE676 SIO chip

Power Monitor

This monitor resides in the SIO chip and monitors the power status signal **PFAIL**. When ac power is interrupted, the power supply applies **PWRFAIL** to the SPU. The resulting signal, **PFAIL**, is applied to the power monitor via the SIO chip **PF** input. The power monitor generates a power-change interrupt under one of two circumstances:

- **PFAIL** changes from negative to positive, indicating that ac power has failed.
- **PFAIL** changes from positive to negative *and* dc power has remained within specifications.

The second condition indicates that ac power has returned after a momentary interruption. The SIO actually detects this condition by noticing that there was no system reset. A system reset occurs when the power status signal **PWROK** goes from negative to positive.

Under either condition, the SIO chip drives its interrupt request pin (**INTRQ**) low, causing a power-change interrupt. Unless interrupts are disabled, the CPU responds to this interrupt with an *interrupt acknowledge* (**DIB CPU**), which causes the SIO to return device code 0.

The power-change interrupt is cleared by a *CPU Acknowledge* instruction (**DOAP CPU**), with bit 0 of the specified accumulator set to 1. (The *CPU Acknowledge* instruction is issued automatically by the virtual console.) If the powerfail condition still exists, the interrupt is reasserted immediately.

The power monitor generates a power-up, nonmaskable interrupt (NMI) when **PFAIL** changes from positive to negative and dc power has just risen; that is, the SIO has received a **RESET** signal. In this case, the SIO chip sets bit 4 in the SPU status register (power-up) to 1 and asserts **NMI**. Bit 4 is cleared as just described by a *CPU Acknowledge* instruction with bit 4 of the specified accumulator set to 0. (The virtual console automatically issues the *CPU Acknowledge* instruction.)

Programmable Interval Timer (PIT)

This timer generates an interrupt when its counter overflows. A *Specify Initial Count* instruction (**DOA PIT**) loads the counter with the two's complement of the desired count. When the counter overflows, the SIO chip asserts **INTRQ**, but does not stop the counter. When the CPU acknowledges the interrupt, the SIO chip returns device code 43.

The user program can measure interrupt latency by interrogating the present counter value with a *Read Count* instruction (**DIA PIT**). A Start command (**NIOS PIT**) starts the counter, and a Clear command (**NIOC PIT**) stops it. Since the PIT counter/register is double-buffered, no counts are lost when the register is read. The PIT counter rate is determined by installing the system configuration jumpers as described earlier in "Installation and Jumpering" and summarized in Table 1.29. The PIT interrupt priority mask bit is bit 6.

Real Time Clock

The RTC generates interrupts (SIO chip asserts **INTRQ**) at any one of four program-selectable frequencies: ac line frequency, 10 Hz, 100 Hz or 1 KHz. When the CPU acknowledges an interrupt caused by the RTC, the SIO chip returns device code 14. The RTC rate is set according to bits 14 and 15 of the specified accumulator with a *Select RTC Frequency* instruction (**DOA RTC**). Refer to the ECLIPSE *S/120 Assembly Language Programmer's Reference* for details. The RTC interrupt priority mask bit is bit 13.

After power-up or an *I/O Reset* instruction, the real time clock frequency is automatically set to the ac line frequency.

Asynchronous Interface

This universal asynchronous receiver/transmitter (UART) consists of a system terminal input (TTI) with device code 10 (interrupt priority mask bit 14) and a system terminal output with device code 11 (interrupt priority mask bit 15). The SIO chip contains the buffers, registers, and state-machine logic for the interface. The driver and receiver for the EIA RS-232-C or 20-mA current loop lines are separate IC devices.

The interface operates at one of 16 baud rates, selected by the installation of four of the system configuration jumpers as described in the "Installation and Jumpering" section and summarized in Table 1.28. The following baud rates have two stop bits: 50, 75, 110, 134.5. The remaining rates have 1 stop bit: 150, 200, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, 19200, and 38400.

The system terminal output section is equipped with a Clear To Send (CTS) input, which inhibits the shifting out of the serial data when the device is not asserting CTS. The CTS input is only monitored at the start of a TTO cycle: a cycle does not abort if CTS is changed while it is in progress.

The system terminal input section contains logic that detects the depression of the console Break key. Logic contained on the SIO chip interprets two consecutive framing error character codes as a depression of this key. If the Break key enable system configuration jumper has been installed as described in the "Installation and Jumpering" section, this logic then causes the Break key bit in the CPU status register to be set and the **NMI** signal to be asserted. The nonmaskable interrupt causes the processor to enter the virtual console mode. Refer to Chapter 7 for information on the virtual console.

NOTE: *The S/120 asynchronous communications interface receives and transmits eight-bit data characters without parity. If the system console device being used with the S/120 operates with a data character length of seven bits, you should configure the device to operate with* mark parity. *If the system console operates with a data character length of eight bits you should configure the device to operate with* no parity. *When receiving data characters from a seven bit system console device, software should mask out the parity bit after the character has been loaded into an accumulator. The parity bit is the most significant bit of the character and is contained in bit 8 of the specified accumulator.*

CPU Status Register

The CPU status register is a 13-bit register that reports the status information listed in Table 2.1. Eight of these bits are contained within the SIO chip. The remaining bits are provided by circuitry contained on the SPU board. A *Read CPU Status* instruction (**DIS** *ac*,**CPU**) places the contents of this status register into the specified accumulator. One of these status bits (bit 0) reports the occurrence of a maskable powerfail interrupt. Four of the status bits (bits 3, 4, 5, and 10) report the occurrences of nonmaskable interrupts. These bits can be cleared (set to 0) with a **DOAP** *ac*,**CPU** instruction with the corresponding bit in the specified accumulator set to 1. The remaining bits report the status or configuration information listed in Table 2.1. These bits cannot be cleared although the Interrupt On bit becomes 0 when CPU interrupts are disabled.

NOTE: *The virtual console clears those bits that cause it to be entered (bits 3, 4, 5, 8, and 10). They are not ordinarily manipulated by the user.*

The CPU status signals, external to the SIO chip, are applied to the CPU address/data bus in response to the signals $\overline{\text{DIS}}$ and $\overline{\text{DV77SEL}}$. These signals are asserted by the I/O decoder section during the data phase of a **DIS** *ac*,**CPU** instruction. The external status bits combine with the internal SIO CPU status bits on the CPU address/data bus to form the full CPU status word.

SIO Configuration Jumpers

The SIO configuration jumpers select the operating characteristics for the SIO internal circuitry. These characteristics were detailed in the "Installation and Jumpering" section of Chapter 1. The configuration jumper contents are applied to the CPU bus ($\overline{\text{MB<0-15>}}$) whenever a system reset is performed. A system reset is performed when power is first applied or the control panel is unlocked and the RESET switch is pressed. Either of these conditions assert **RESET** and $\overline{\text{RESET}}$.

Bits	Name	Function
0	Powerfail	Signifies the state of the power supply powerfail line.
1	Interrupt On	1 signifies that CPU interrupts are on (enabled); 0 signifies that CPU interrupts are off (disabled).
2	SIO	Used for diagnostics; always set to 1.
3	Break Key	1 signifies that the system console Break Key has been depressed.
4	Power Up	1 signifies that the system has just been powered up, or the front console RESET switch has been depressed.
5	Halt	1 signifies that a *Halt* instruction has been decoded and the Halt Dispatch bit in the SPU register is 1. (See bit 6.)
6	Halt Dispatch	Signifies the state of the Halt Dispatch jumper in the SIO system configuration jumper register. A 1 means the Halt Dispatch jumper is installed. In this case, control passes to the virtual console when a *Halt* instruction has been decoded. 0 means that the Halt Dispatch jumper is removed. In this case, the CPU simply halts when a *Halt* instruction is decoded; bit 5 in the SPU register will not be set.
7	Interrupt Request	Signifies the state of the SIO interrupt request pin ($\overline{\text{INTRQ}}$). 1 means a device is requesting an interrupt. When interrupts are disabled, this bit can be used to test for interrupt requests. Any transition (low-to-high or high-to-low) of $\overline{\text{PWRFAIL}}$ sets this bit to 1.
8	Program Load	1 signifies the front console program load (PL) switch has just been depressed.
9	-----	Reserved for future use. Set to 0.
10	TRAP	1 signifies the CPU is operating in user mode; 0 signifies operation in the virtual console mode.
11	Memory Data Valid	1 signifies memory data is valid following a power disruption.
12-13	Memory Capacity	Signify capacity of implemented system memory.
14	-----	Reserved for future use. Set to 0.
15	Test	Used for diagnostics; always set to 0.

Table 2.1 CPU status register definitions

Address Latch

The *address latch* preserves the data on the memory bus (**MB<0-15>**) from the time **TTL01** is asserted during an address phase (**ADREN**) until the next address phase occurs. ADREN is asserted by the CPU, the SIO chip, or the data channel logic during address phases. **TTL01** is asserted by system and memory timing logic. The latched address word is applied to the address bus (**A<0-15>**). Part or all of this address is supplied to the I/O decode circuits of the CPU section, the virtual console, the MAP and virtual console scratch-pad memory address selection logic, and the system memory.

The signals **ADREN** and **TTL01** also latch the following system status and control signals for the same time period: **$\overline{\text{MAPEN}}$**, **MEMCYC**, **$\overline{\text{WH}}$**, **$\overline{\text{WL}}$**, **$\overline{\text{RAMD0}}$** and **$\overline{\text{RAMD1}}$**. The states of **$\overline{\text{RAMD0}}$** and **$\overline{\text{RAMD1}}$** indicate whether a logical page of memory is write protected or invalid.

I/O Decoder

The I/O decoder consists of three 1 of 8 decoders, one 1 of 4 decoder, and associated logic. Its functions are to decode I/O format instructions and to route the decoded signals to the I/O devices. It decodes the I/O command field and the flag control pulse field. It also decodes the device code field to determine if the addressed device is one of the on-board I/O devices (CPU, MAP, ERCC, and APL). When the device code field specifies the I/O format instruction is intended for one of the on-board devices, the I/O decoder disables the external I/O data bus drivers and receivers. The I/O decoder receives timing and control signals from the CPU section and the latched mE674 CPU encoded PIO instruction from the address bus (**A<0-15>**). Depending on the states of these inputs, this section asserts the following signals:

- All I/O command strobes (**DOA, DOB, DOC, DIA, DIB, DIC, IORST, MSKO, INTA**) and flag control strobes (**STRT, CLR, IOPLS**) to the ECLIPSE I/O interface and SPU devices addressed by I/O format instructions.
- All select signals for SPU devices addressed by I/O format instructions (**DV77SEL, APLSEL, ECCSEL, MAPSEL**).
- **INTACYC**, which signals the I/O interface timing that an interrupt acknowledge cycle is in progress.
- **$\overline{\text{LCLSEL}}$** and **LCLSEL**, which disable the I/O bus drivers and receivers when an on-board I/O device has been selected.

Virtual Console

The virtual console shown in Figure 2.12 consists of a programmed read-only memory containing a 1024 word (16 bits per word) program and a 512 byte array of random access (read/write) memory. The program consists of a self-test routine, an automatic program loading (APL) routine, and a set of commands for user access to CPU registers and system memory to be used for debugging object language programs. All of these programs are described in Chapter 7.

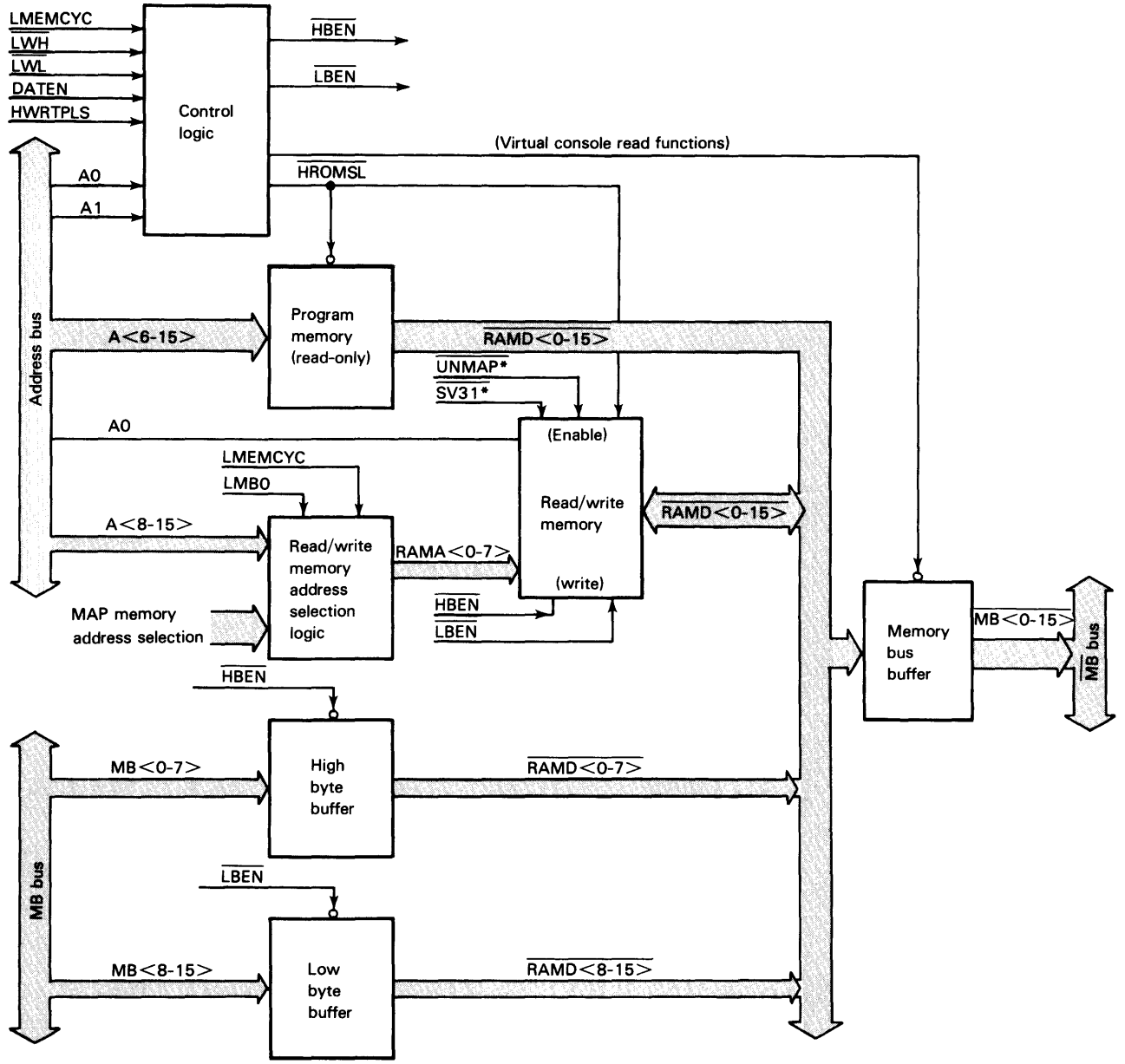
The contents of the virtual console read-only memory location addressed by **A<6-15>** are placed on the memory bus during the data phase (**DATEN**) of a SPU read cycle (**$\overline{\text{LWH}}$** and **$\overline{\text{LWL}}$** both high) that addresses the virtual console program memory. The virtual console program memory is addressed when **LMEMCYC** is asserted, and **A0** and **A1** both equal 1.

The contents of the virtual console read/write memory location addressed by **A<8-15>** (via the address selector) are either read or written during the data phase of an SPU cycle that addresses the virtual console read/write memory. The virtual console read/write memory is addressed when **LMEMCYC** is asserted and **A0** and **A1** both equal 0. Data transfers occur via the memory bus. Two system control signals, **$\overline{\text{LWH}}$** and **$\overline{\text{LWL}}$**, determine the direction of transfer:

- When neither is asserted, a word is read from the memory and applied to **$\overline{\text{MB<0-15>}}$**.
- When both **$\overline{\text{LWH}}$** and **$\overline{\text{LWL}}$** are asserted, a word is written.
- When only one of the two control signals asserts, a byte is written: **$\overline{\text{LWH}}$** causes the data contained on **MB<0-7>** to be written when the system and memory timing unit asserts **HWRTPLS**. (The control logic asserts **$\overline{\text{HBEN}}$** to enable the high-byte buffer and write the high byte); **$\overline{\text{LWL}}$** causes the data contained on **MB<8-15>** to be written when the system and memory timing unit asserts **HWRTPLS**. (The control logic asserts **$\overline{\text{LBEN}}$** to enable the low-byte buffer and write the low byte.)

Automatic Program Load

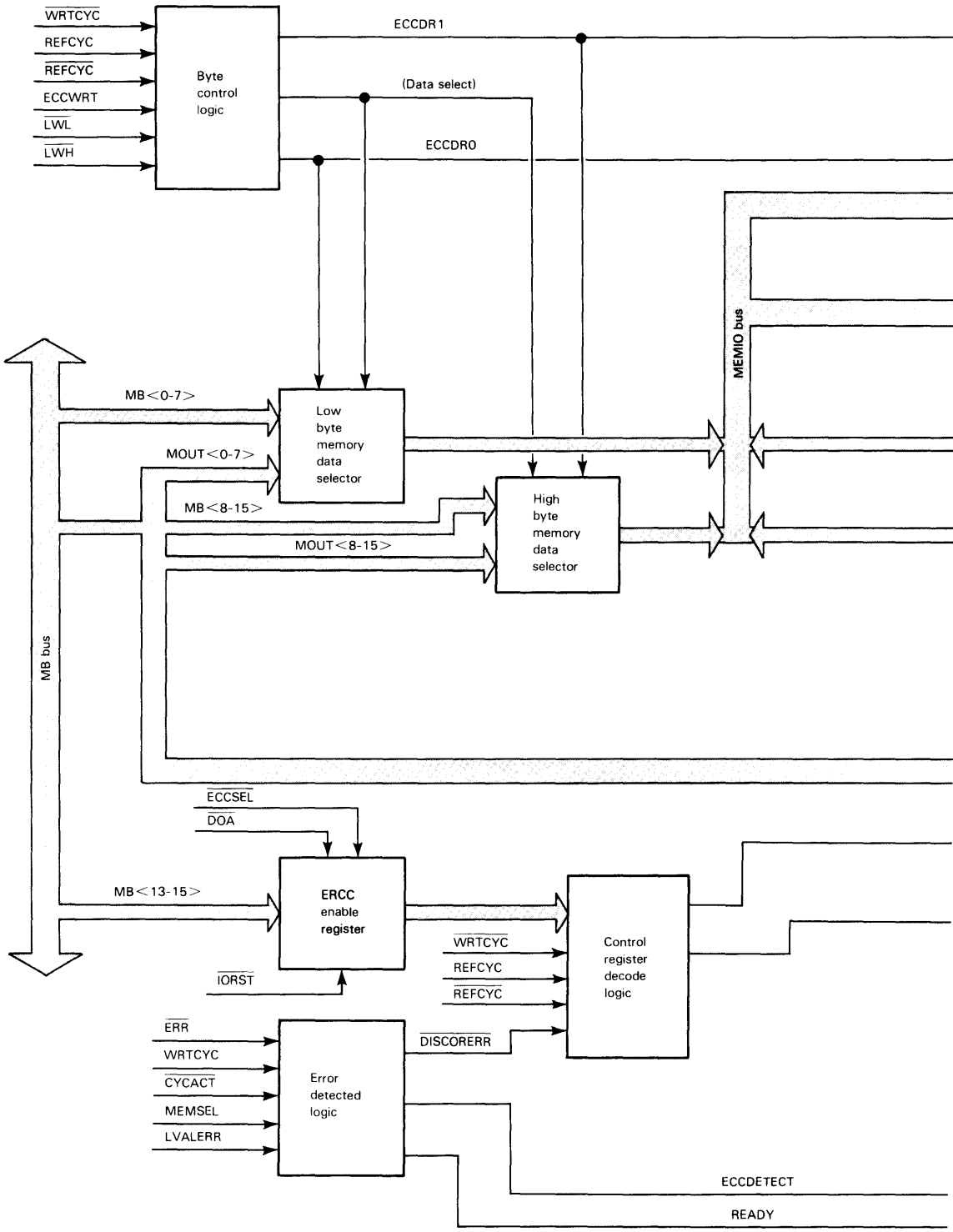
The automatic program load (APL) consists of seven jumpers and an 8-bit tri-state driver. The driver passes the APL device selection code and the state of the front console Lock switch to the memory bus (via the IOOUT bus) when the virtual console executes a read APL command. (The I/O decoder is asserting **$\overline{\text{DIA}}$** and **$\overline{\text{APLSEL}}$**.)



* Signals asserted low only during MAP functions.

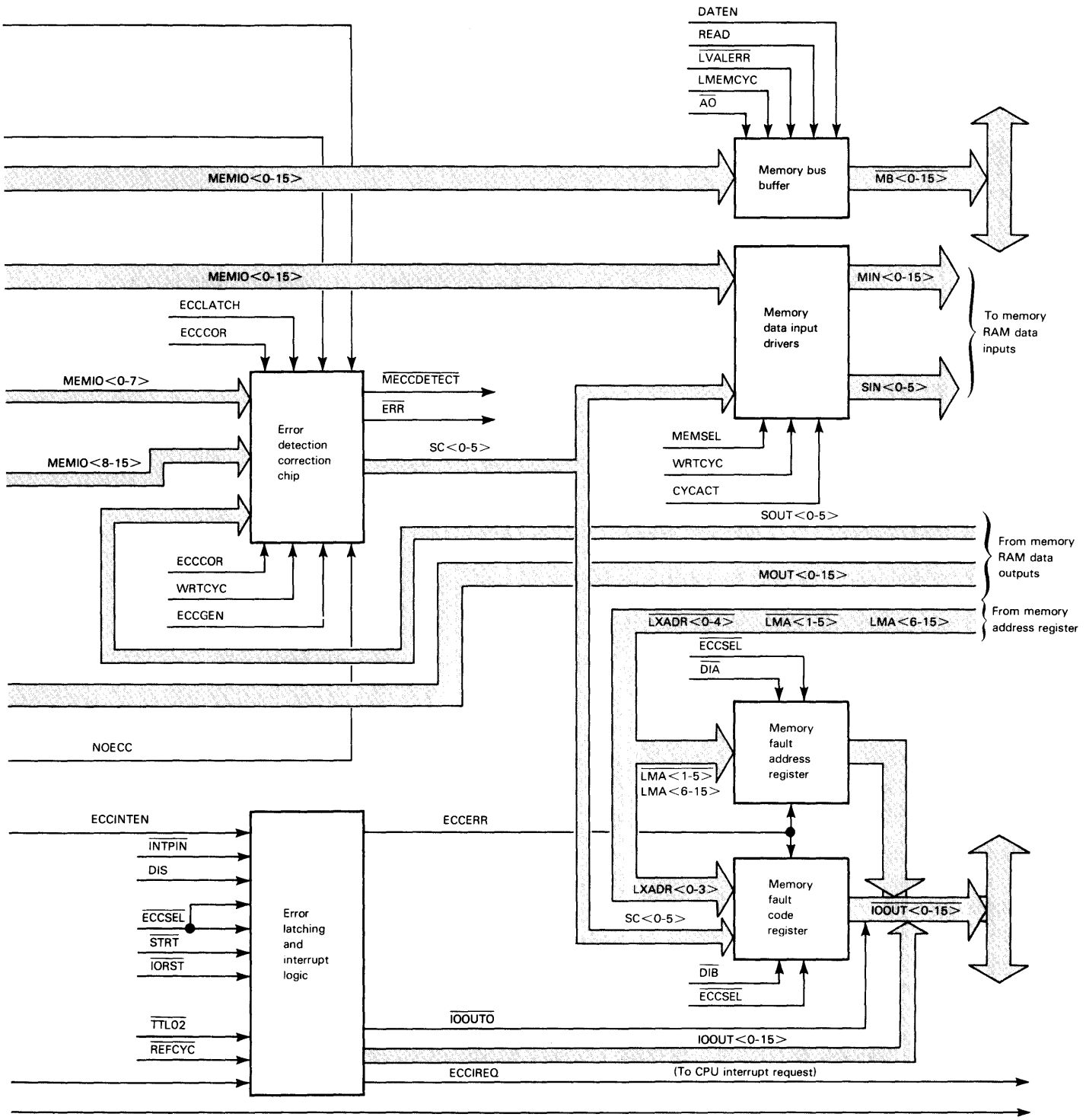
DG-09050

Figure 2.12 Virtual console



DG-09052

Figure 2.13 Error checking and correction unit


DATEN
READ
LVALERR
LMEMCYC
AO
Memory bus buffer
MB<0-15>
MEMIO<0-15>
MEMIO<0-15>
Memory data input drivers
MIN<0-15>
SIN<0-5>
To memory RAM data inputs
ECCLATCH
ECCCOR
MEMIO<0-7>
MEMIO<8-15>
Error detection correction chip
MECCDETECT
ERR
SC<0-5>
MEMSEL
WRTCYC
CYCACT
SOUT<0-5>
From memory RAM data outputs
MOUT<0-15>
From memory address register
LXADR<0-4>
LMA<1-5>
LMA<6-15>
ECCCOR
WRTCYC
ECCGEN
ECCSEL
DIA
Memory fault address register
NOECC
LMA<1-5>
LMA<6-15>
ECCINTEN
ECCERR
INTPIN
DIS
ECCSEL
STRT
IORST
Error latching and interrupt logic
LXADR<0-3>
SC<0-5>
Memory fault code register
IOOUT<0-15>
DIB
ECCSEL
TTL02
REFCYC
IOOUTO
IOOUT<0-15>
ECCIREQ
(To CPU interrupt request)

The virtual console can also generate a nonmaskable interrupt (**FRCNMI**) to the CPU and control the front console RUN indicator by executing a **DOA** *ac*,**APL** instruction with bit 15 of the specified accumulator set as required. When bit 15 is 1, the RUN indicator is turned on; no NMI is asserted. When bit 15 is 0, the RUN indicator is turned off; an NMI is asserted and bit 10 in the CPU status register is set to 1.

Error Checking and Correction Unit

The S/120 error checking and correction (ERCC) facility generates and appends a 6 bit check code to each word (2 bytes) of data written to memory. During memory-read operations, the 22-bit word from memory is processed by the ERCC facility (when enabled) to determine if an error has occurred. If a single-bit error is detected, the erroneous bit is corrected before the word is transferred. The corrected word is also written to memory along with a new check code. In addition, the fault address and an error syndrome code are recorded for transfer to the CPU, and an interrupt is issued (when enabled). This fault address and error syndrome code can be used to identify a marginal or failing system memory RAM chip.

Double-bit and some triple-bit errors are detected but not corrected. However, their fault addresses and error syndrome codes are recorded and an interrupt (when enabled) is issued.

In a process called *sniffing*, the ERCC facility also detects and corrects single-bit errors during memory refresh cycles (when enabled). However, their fault addresses and error syndrome codes are not recorded and no interrupt is issued.

The ERCC facility, shown in Figure 2.13 (on the two preceding pages), is comprised of

ERCC enable register with decode logic
memory fault code register
memory fault address register
error detection/correction (EDC) chip
input data selectors with byte control logic
memory data input drivers
memory bus buffers
error detecting logic
error latching and interrupt logic

Enable Register

This register's content determines the ERCC logic operational mode as detailed in Table 2.2. An *Enable ERCC* instruction (**DOA** *ac*,**ERCC**) loads the ERCC enable register with memory bus bits **MB<13-15>** when the I/O decoder asserts the control signals **DOA** and **ECCSEL**. An *I/O Reset* instruction (**DIC** *ac*,**CPU**) sets this register to zeros when the I/O decoder asserts **IORST**.

Bit 13	Bit 14	Bit 15	Operation Mode
0	0	0	Write checkcode; disable checking and correction during memory read; disable interrupts; enable checking and correction during memory refresh.
0	0	1	Disable writing checkcode; disable checking and correction during memory read; disable interrupts; enable checking and correction during memory refresh.
0	1	0	Write checkcode; enable checking and correction during memory read and refresh; disable interrupts.
0	1	1	Write checkcode; enable checking and correction during memory read and refresh; enable interrupts.
1	0	0	Write checkcode; disable checking and correction during memory read and refresh; disable interrupts.
1	0	1	Disable writing checkcode; disable checking and correction during memory read and refresh; disable interrupts.
1	1	0	Write checkcode; enable checking and correction during memory read; disable interrupts; disable checking and correction during memory refresh.
1	1	1	Write checkcode; enable checking and correction during memory read; enable interrupts; disable checking and correction during memory refresh.

Table 2.2 ERCC enable register bit definitions

Fault Registers

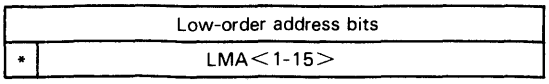
Two registers, the fault location register and the fault code register, record the physical address of the memory location in error and the error syndrome code when error checking and correction is enabled and a memory error is detected. The contents of the ERCC enable register determine whether error checking and correction is enabled.

When a memory error is detected, the error syndrome code (**SC<0-5>**) produced by the EDC chip and the current contents of the memory address register (**LXADR<0-4>**, **LMA<1-5>**, **LMA<6-15>**) from the memory unit logic are loaded into the two fault registers by the control signal **ECCERR**. Once an error has been recorded, the contents of these registers remain latched until that error is cleared by a programmed instruction (described later in this section). The formats of the memory fault address and fault code registers are shown in Figures 2.14 and 2.15. Refer to the *ECLIPSE S/120 Assembly Language Programmer's Reference* (DGC No. 014-000686) for the error syndrome codes produced by the EDC chip.

A *Read Memory Fault Address* instruction (**DIA** *ac*,**ERCC**) transfers the content of the fault address

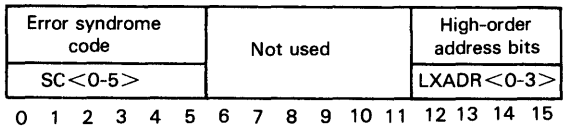
register to the processor's specified accumulator. This transfer is performed via the I/O out bus (**IOOUT<0-15>**) and the memory bus (**MB<0-15>**) when the control signals **DIA** and **ECCSEL** are asserted by the I/O decoder.

A *Read Memory Fault Code* instruction (**DIB** *ac*,**ERCC**) transfers the content of the fault code register to the processor's specified accumulator. This transfer is performed via the I/O out bus (**IOOUT<0-15>**) and the memory bus (**MB<0-15>**) when the control signals **DIB** and **ECCSEL** are asserted by the I/O decoder.



DG-09053

Figure 2.14 ERCC memory fault address register format



DG-09054

Figure 2.15 ERCC memory fault code register format

EDC Chip

The EDC chip, corresponding to control signals applied by the system and memory timing unit and ERCC facility control logic, functions in one of four modes: generate, detect, correct, or pass through.

In the *generate mode*, the EDC chip generates a check code corresponding to the input data bits (**MEMIO<0-15>**). This check code is applied to the **SC<0-5>** signal lines.

In the *detect mode*, the EDC chip compares the check code read from memory (**SOUT<0-5>**) against a new check code generated from the input data bits (**MEMIO<0-15>**). If the check codes agree, there is no error. If the check codes do not agree, one or more of the data or check bits applied to the EDC is erroneous and an error syndrome code is generated and applied to the **SC<0-5>** signal lines. In addition, the EDC chip asserts an error signal (**ERR**). If the generated error syndrome code indicates more than one bit is in error the EDC chip also asserts the multiple error signal (**MECCDETECT**).

In the *correct mode*, the EDC chip functions as it does in detect mode, except any single-bit error contained in the input data bits (**MEMIO<0-15**) is corrected. The generated error syndrome code determines the bit in error and that bit is complemented.

In the *pass through mode*, the EDC chip error signals are disabled and the check code bits applied to the EDC input are placed on the **SC<0-5>** signal lines. When error checking and correction is disabled (as determined by the content of the ERCC control register), the EDC chip is placed in the pass through mode.

Read Operation.

During system memory read operations, with error correction enabled, control signals place the EDC chip in the detect mode. The data bits read from memory (**MOUT<0-15>**) are applied to the EDC chip and the memory bus drivers via the input data selector and the **MEMIO<0-15>** signal lines. (**WRTCYC** high, selects **MOUT<0-15>** and **ECCWRT** low enables the selector outputs.) The check-code bits read from memory (**SOUT<0-5>**) are applied directly to the EDC chip. Both the data and check code bits are latched within the EDC chip. Provided no error is detected and a validity-protected page was not addressed (**LVALERR** high), the data bits (**MEMIO<0-15>**) from the input data selectors are output to the memory bus (**MB<0-15>**) during the data phase (**DATEN**) of a system memory-read reference. **DATEN** originates from the system and memory timing logic during system memory-read operations.

If an error is detected during the read operation (**ERR** asserted), the ERCC error flag is set, unless the MAP unit detected a validity-protection fault. (Refer to "MAP Operations" and "Memory Allocation and Protection" in this chapter for validity protection fault detail.) When the error flag is set, the error syndrome code produced by the EDC chip, and the physical address of the memory location at fault are latched in the two fault registers. The error flag also causes an interrupt (if enabled by the ERCC control register) to be generated. The error causes system and memory timing to initiate a correction cycle. During this cycle, the bit in error is corrected before the data is transferred to the receiving element. This cycle also rewrites the corrected data word and a new generated check code into system memory unless a multiple-bit error (**LMECCDETECT**) was detected. In the correction cycle, control signals place the EDC chip in the correct mode and the data latched in the EDC is corrected and then applied to the memory bus drivers via the **MEMIO<0-15>** signal lines. (Input data selectors are disabled during the correction cycle.) The corrected data (**MEMIO<0-15>**) and new clock code (**SC<0-5>**) are applied to the system memory via the memory data input drivers, and the memory data signal lines (**MIN<0-15>** and **SIN<0-5>**).

Write Operation.

The S/120 can write a data word (2 bytes) or data byte to system memory; however, system memory always writes a word to memory. Therefore, when a single byte is to be written, the other byte must be retained and rewritten. In this case, the entire word is first read from memory and any single-bit error in the data word is corrected. The unchanged byte is then combined with the changed byte, and the entire word is written to memory. Six check-code bits are calculated and appended to the word when it is written in system memory. If a multiple-bit error is detected, the entire word remains unchanged in memory. Data-channel memory-write operations always write a word to memory.

Word Write Operation.

During a system memory word-write operation with error correction enabled, control signals place the EDC chip in the generate mode. The data word to be written (**MB<0-15>**) is applied to the EDC chip via the input data selector and the **MEMIO<0-15>** signal lines (**$\overline{\text{WRTCYC}}$** low, **REFCYC** low, selects **MB<0-15>** while **ECCWRT** low enables the selector output). The **MEMIO<0-15>** signal lines also drive the memory data input drivers. The EDC produces and applies the 6-bit check code to the memory data input drivers via the **SC<0-5>** signal lines and the 22 bits are written into the addressed memory location. The memory input data drivers apply the data word and check code to the memory via the **MIN<0-15>** and **SIN<0-5>** signal lines.

Byte Write Operation.

During a system memory byte-write operation with error correction enabled, control signals first place the EDC chip in the detect mode. After a memory location has been addressed, the contents of that location (**MOUT<0-15>**) are read and applied to the EDC chip via the input data selector and the **MEMIO<0-15>** signal lines (**$\overline{\text{WRTCYC}}$** high, selects **MOUT<0-15>** while **ECCWRT** low enables the selector outputs). The check code bits read from memory (**SOUT<0-5>**) are applied directly to the EDC chip. If an error is detected during the read phase (**$\overline{\text{ERR}}$** or **$\overline{\text{MECCDETECT}}$** asserted), the error syndrome code produced by the EDC chip and the physical address of the memory location at fault are stored in the two fault registers. In addition, the ERCC error flag is set and an interrupt (if enabled by the ERCC control register) is generated.

Next, the EDC chip is placed in the correction mode. The data word is corrected, if neccessary, and latched within the EDC chip.

Next the EDC chip is placed in the generate mode. Depending on the states of **$\overline{\text{LWH}}$** and **$\overline{\text{LWL}}$** (only one is active for a byte write operation), the data byte to be written is applied to the EDC chip via the respective input data selector and the **MEMIO<0-15>** signal lines. (**$\overline{\text{WRTCYC}}$** low, **REFCYC** low, selects **MB<0-15>** while **REFCYC** low, and the respective **$\overline{\text{LWL}}$** or **$\overline{\text{LWH}}$** low, enables the selector output.) The unchanged byte is applied to the respective **MEMIO<0-15>** signal lines via the EDC chip. The **MEMIO<0-15>** signal lines also drive the memory data input drivers. The EDC produces and applies the six-bit check code to the memory data input drivers via the **SC<0-5>** signal lines and the entire 22 bits are written into the addressed memory location, unless a multiple-bit error was detected during the detect mode. The memory input data drivers apply the data word and check code to the memory via the **MIN<0-15>** and **SIN<0-5>** signal lines.

Refresh Operation.

A system memory-refresh operation refreshes 256 locations of all the read/write memory chips contained in the memory. In addition, when error checking and correction is enabled, the ERCC unit continuously checks memory at the rate of one sequential memory location per refresh cycle. (Refer to "System Memory" in this chapter for a detailed description of the memory refresh operation.)

During the system memory-refresh cycle, control signals place the EDC chip in the detect mode. The data bits read from memory (**MOUT<0-15>**) are applied to the EDC chip via the input data selector and the **MEMIO<0-15>** signal lines. (**$\overline{\text{WRTCYC}}$** high selects **MOUT<0-15>**, and **ECCWRT** low enables the selector outputs).

The check code bits read from memory (**SOUT<0-5>**) are applied directly to the EDC chip. Both the data and check code bits are latched within the EDC chip. If no error is detected, the refresh operation is complete. If an error is detected (**$\overline{\text{ERR}}$**), system and memory timing initiates a correction cycle. This cycle corrects the bit in error, generates a new check code, and rewrites the corrected data word and the newly generated check code into system memory unless a multiple bit error (**$\overline{\text{MECCDETECT}}$**) is detected. When a multiple-bit error is detected, the correction cycle is performed, but the uncorrected word is retained in memory. (Write pulse to memory is inhibited.) In the correction cycle, control signals place the EDC in the correct mode. The data latched within the EDC is corrected and then applied to the memory via the **MEMIO<0-15>** signal lines, the memory data input drivers, and the memory data signal lines (**MIN<0-15>** and **SIN<0-5>**). (Input data selectors are disabled during the correction cycle.) Errors detected during refresh are not recorded and no interrupt is generated.

Error Flag Logic.

The ERCC Error (**ECCERR**) flag is set to 0 when any instruction addressed to the ERCC facility specifying a START flag control function (**STRT** and **ECCSEL**) or an *IO Reset* instruction is executed. If error checking has been enabled, then any check error detected during detect mode sets the ERCC error flag to 1. If the ERCC error interrupt has been enabled, setting the Error flag causes a CPU interrupt. Once the ERCC Error flag is set, the memory fault address and memory fault code register remain latched until the Error flag is cleared.

Additional errors detected while the Error flag is set are not be recorded. When a *Data In Status* instruction specifying the ERCC is executed, the control (**ECCSEL**) and strobe (**DIS**) signals cause the state of the Error flag to be placed on memory bus bit position 0 via the I/O out bus.

Memory Allocation and Protection

The memory allocation and protection (MAP) unit translates logical memory page addresses into physical memory page addresses. It permits a 15-bit memory address to be expanded to 18 bits. This allows the CPU to access the maximum memory (512 kbytes) contained on an S/120 SPU board. The MAP unit stores the eight address translation functions (address maps) that can be used by the S/120 system. Each address map can store the translation information for up to 32 logical pages, each containing 2,048 bytes. In addition to translating addresses, the MAP feature performs validity protection, write protection, I/O protection and indirection protection.

The MAP unit receives addresses and data (to be placed in the MAP memory and registers) from the memory bus or buffered memory bus. It produces the physical page number bits (**RAMD<6-15>**) required to map the current user's logical address space into physical address space. The functional elements of the MAP unit, shown in Figure 2.16 are

address translation (MAP) memory
MAP memory address selection logic
MAP memory data input buffer
page 31 address register
MAP status register (partial)
page check register
page 31 control logic
set logical page invalid decode logic
memory protection logic
unmapped control logic
unmapped address drivers

NOTE: *The MAP status register, physically located in the MAP unit, duplicates seven selected bits (bits 0, 6-8, 11, 13, and 14) of the MAP status register contained in the mE674 CPU.*

MAP Operations

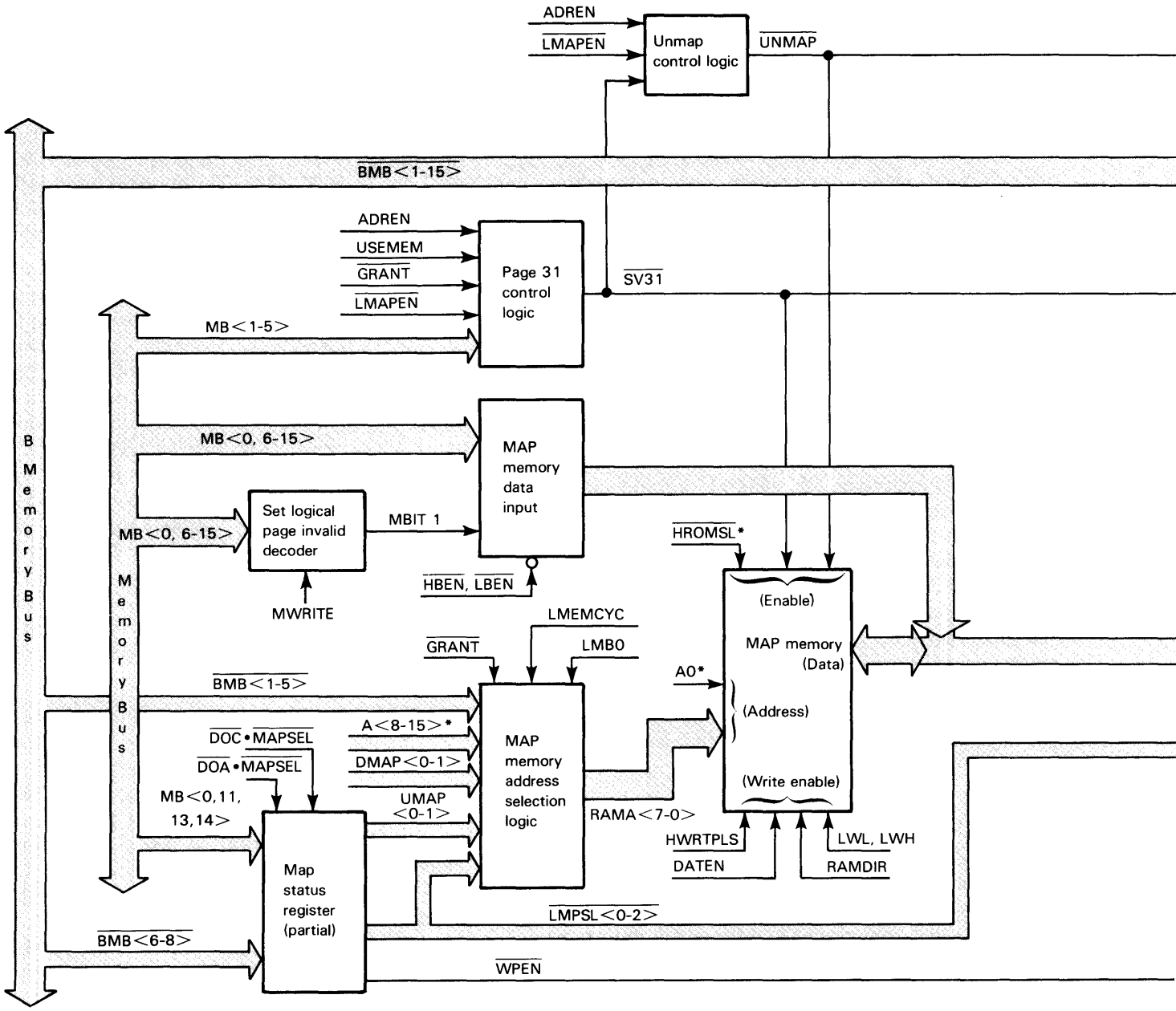
Load map, mapped mode, and unmapped mode operations are discussed in the following paragraphs.

Load Map Operation.

Address translation data for four user maps (A, B, C, or D) and four data channel maps (A, B, C, or D) is stored in the MAP memory. One user or data-channel map is loaded in response to a *Load Map* instruction (**LMP**) preceded by a *Load Map Status* instruction (**DOA** *ac*,**MAP**) or an *Initiate Page Check* instruction (**DOC** *ac*,**MAP**). A single *Load Map* instruction can load all 32 ($0\text{-}37_8$) logical page locations of a selected map.

A *Load Map Status* or *Initiate Page Check* instruction loads the MAP status register with B memory bus bits (**BMB<6-8>**) when the I/O decoder asserts control signals **DOA** and **MAPSEL** or **DOC** and **MAPSEL**. (The *Initiate Page Check* instruction loads these bits into the page check register as well.) These three bits of the status register produce map select signals, **LMPSL<0-2>**, which select one of the eight maps within the MAP memory to be loaded by the next *Load Map* instruction. In addition, the *Load Map Status* instruction loads memory bus bits **MB0, MB11, MB13,** and **MB14** into the MAP status register. MAP status register bits 0 and 13 specify the next user map to be enabled; bit 11 determines if write protect is enabled for the next user; and bit 14 determines if mapping of data channel addresses will be enabled after this instruction.

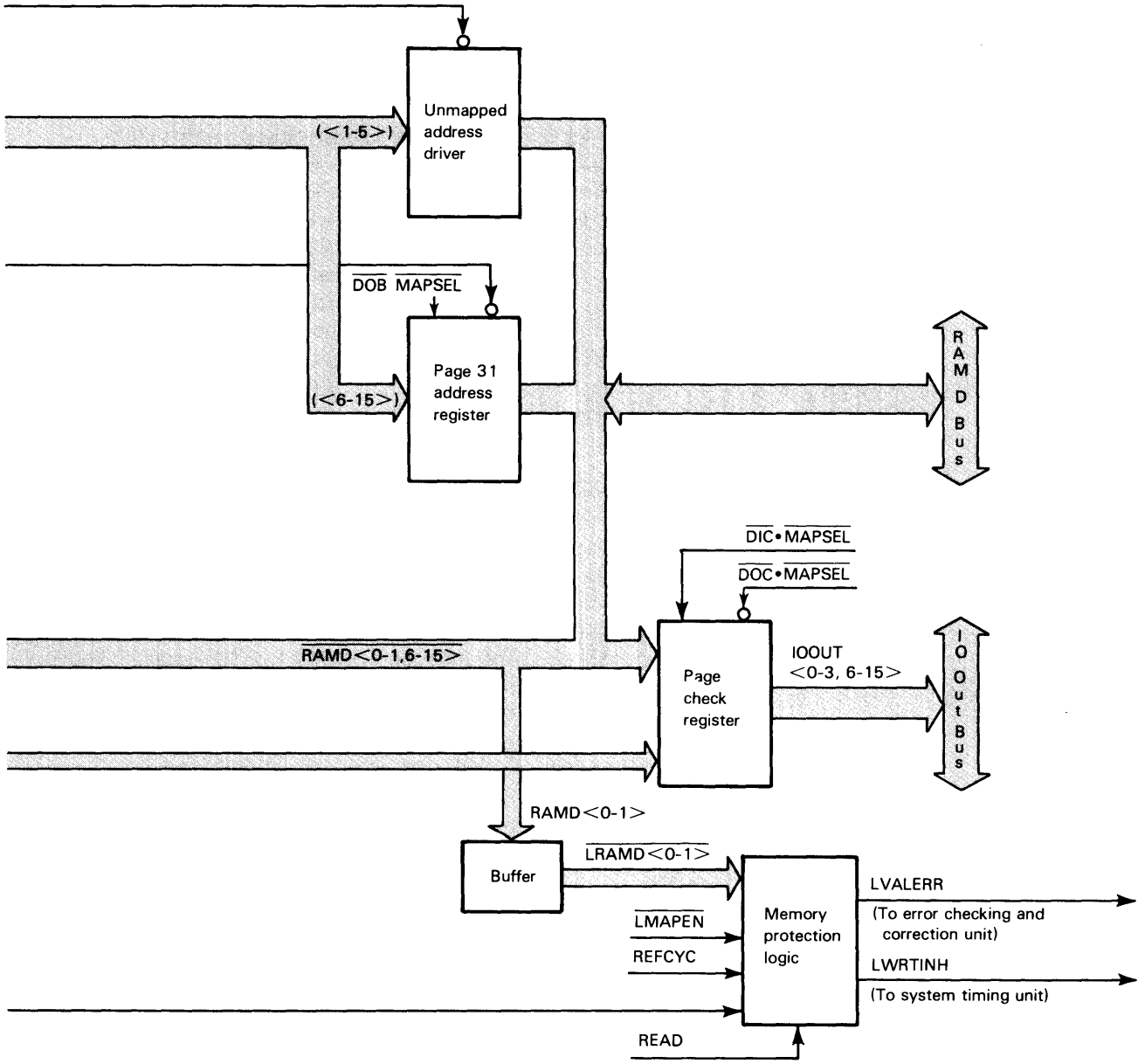
When a *Load Map* instruction is executed, the MAP page assignments are made. Bits **MB<1-5>** on the memory bus represent the logical page number and address the page number in the selected map of the MAP memory. Bits **MB<6-15>** represent the physical page number to be assigned to the given logical page number; these bits are written in the addressed location in MAP memory when system and memory timing asserts **HWRTPLS.** Additionally, two more bits are stored in the same addressed location: the write protect bit, received via **MB0**; and the validity protect bit, **MBIT1**, received via the set logical page invalid decoder. The set logical page invalid decoder detects that the address translation functions for the present word (bits **MB<0,6-15>**) being written into MAP memory are all set to one. This condition declares a logical page invalid.



NOTE: [1] *Signals marked with asterisk(*) pertain to virtual console read/write memory which is physically contained within the same IC's as MAP memory (refer to virtual console in this chapter).*

DG-09051

Figure 2.16 MAP unit


Unmapped address driver
(<1-5>)
DOB MAPSEL
Page 31 address register
(<6-15>)
RAM D Bus
DIC•MAPSEL
DOC•MAPSEL
RAMD<0-1,6-15>
Page check register
IOOUT <0-3, 6-15>
I O Out Bus
RAMD<0-1>
Buffer
LRAMD<0-1>
LMAPEN
REFCYC
Memory protection logic
LVALERR
(To error checking and correction unit)
LWRTINH
(To system timing unit)
READ

Mapped Mode Operation.

A *Load Map Status* instruction (**DOA** *ac*,**MAP**) initiates mapped mode. This instruction sets a data channel and/or user enable bit(s) (bit 14 and/or 15) in the MAP status register to one. A user map is selected by setting the user map selection bits (bits 0 and 13 in the MAP status register), with the same *Load Map Status* instruction. Bits 0 and 13 assert the signals **UMAP0** and **UMAP1**, respectively). In addition, the *Load Map Status* can be used to enable memory-write protection by setting the MAP status register bit 11 to 1. This bit asserts $\overline{\text{WPEN}}$.

A data channel map is selected during a data transfer by two map select bits applied by the device controller performing the transfer. Map select bits $\overline{\text{DATA0}}$ and $\overline{\text{EXDCH}}$ assert the signals DMAP0 and DMAP1, respectively.

During the address phase (**ADREN**) of a system memory cycle (**LMEMCYC, LMB0** low), the MAP memory address selection logic combines the enabled user ($\overline{\text{GRANT}}$ high) or data channel ($\overline{\text{GRANT}}$ low) MAP select bits (**UMAP<0-1>** or **DMAP<0-1>**) and the logical page number ($\overline{\text{BMB<1-5>}}$) to address the MAP memory. The contents of the addressed MAP memory location ($\overline{\text{RAMD<6-15>}}$) are then applied to the memory unit as the physical memory page number that corresponds to the addressed logical page number. The memory unit combines the physical page number with the ten least significant address bits to form the entire physical memory address.

Unmapped Mode Operation.

When the MAP is disabled ($\overline{\text{LMAPEN}}$ is high), user logical pages 0 through 30 (addresses 0 through 75777_8) are not translated. However, user logical page 31 (addresses 76000_8 through 77777_8) is translated by the contents of the special page 31 register. In the unmapped mode with a user logical page in the range 0 through 30 ($\overline{\text{SV31}}$ high), the logical page number bits ($\overline{\text{BMB<1-5>}}$) are applied to the memory unit (via $\overline{\text{RAMD<11-15>}}$) as the physical page number.

MAP Registers

The page 31 and page check registers are discussed in the following paragraphs.

Page 31 Address Register.

The page address 31 register is a 10-bit register containing the physical page number to which logical page 31 (addresses 76000_8 through 77777_8) is to be mapped for user system memory cycles in the unmapped mode. A *Map Page 31* instruction (**DOB** *ac*,**MAP**) loads the Page 31 register from the B memory bus ($\overline{\text{BMB<6-15>}}$) when the I/O decoder asserts $\overline{\text{DOB}}$ and $\overline{\text{MAPSEL}}$.

During the address phase (**ADREN**) of an unmapped ($\overline{\text{LMAPEN}}$) user memory cycle (**USRMEM,** $\overline{\text{GRANT}}$ high), the page 31 control logic asserts $\overline{\text{SV31}}$ when logical page 31 is addressed (bits **MB<1-5>** are all ones). $\overline{\text{SV31}}$ disables the unmapped address drivers and applies the contents of the page 31 register to the memory unit (via $\overline{\text{RAMD<6-15>}}$) as the physical page number.

Page Check Register.

The page check register is an 11-bit register used to return to a processor accumulator the physical page number and write protection characteristic of a specified map and logical page. Two instructions are required to transfer this information to the processor. The first instruction, *Initiate Page Check*, identifies the map and logical page number whose translation data is to be returned to the processor. This instruction also causes the translation data to be loaded into the page check register. The second instruction, *Page Check*, transfers the translation data from the page check register to the processor's accumulator.

An *Initiate Page Check* instruction (**DOC** *ac*,**MAP**) loads the page check register with the translation data from a specified map and logical page, when the I/O decoder asserts $\overline{\text{DOC}}$ and $\overline{\text{MAPSEL}}$. The identity of the logical page number and map to be checked are received via the B memory bus bits 1-5 and 6-8, respectively. The identity code of the map to be checked is loaded into the map selection code field of the MAP status register ($\overline{\text{LMPSL<0-2>}}$). This field and the logical page number ($\overline{\text{BMB<1-5>}}$) are applied to the map memory address lines by the map memory addressed selector logic (**LMEMCYC** low, **LMB0** low). The contents of the addressed map memory location ($\overline{\text{RAMD<0, 6-15>}}$) are latched in the page check register on the trailing edge of $\overline{\text{DOC}}$.

A *Page Check* instruction (**DIC** *ac*,**MAP**) transfers the contents of the page check register to the processor's specified accumulator. This transfer is performed via the I/O out bus ($\overline{\text{IOOUT<0-15>}}$) and the memory bus ($\overline{\text{MB<0-15>}}$) when the I/O decoder asserts $\overline{\text{DIC}}$ and $\overline{\text{MAPSEL}}$.

MAP Faults

The mE674 CPU detects a MAP fault and initiates a MAP fault sequence if any of the following conditions occur.

- A single instruction causes 15 indirect memory cycles when the MAP and indirect protection is enabled.
- A mapped memory reference to a validity-protected page.
- A mapped memory write attempt to a write-protected page when write protection is enabled.

- An I/O instruction is attempted when the MAP and I/O protection are enabled, and the Load Effective Address mode is disabled. (Bit 10 of the MAP status register equals 1 and bit 9 equals 0.)

Memory Protection Logic

This logic provides the memory protection required if a validity-protected or write-protected MAP fault occurs. **LVALERR** is applied to the error checking and correction unit if a validity-protected fault occurs. This signal disables the memory data to memory bus drivers to assure that no memory data reaches the CPU during memory-read cycles which reference a validity-protected memory page. **LWRTINH** is applied to the system timing unit if a write-protected fault occurs. This signal prevents system and memory timing from asserting write-enable pulse (**WEPLS**) to the memory unit when a memory-write cycle references a write-protected memory page.

System Memory

The memory unit provides 128, 256, or 512 Kbytes of dynamic MOS random-access memory (RAM) for the system processor unit. The memory unit utilizes 64K by 1-bit RAM integrated circuits (ICs). Each addressable memory location provides storage for one 16-bit word (2 bytes). An additional six error checking and correction (ERCC) bits are associated with each 16-bit data word in memory. Thus, a memory location is 22 bits wide, requiring 22 RAM ICs. A 128-Kbyte memory unit contains 22 RAM ICs; a 256-Kbyte memory unit contains 44 RAM ICs; a 512-Kbyte memory unit contains 88 RAM ICs.

The system and memory timing unit provides timing for the memory unit. A memory operation either reads or writes a 16-bit data word at the addressed memory location. Six ERCC bits, calculated by an error checking and correction facility, are appended to each word as it is written in the memory. When a memory location is read, the 16 data and 6 ERCC bits are applied to the ERCC facility which, if enabled, checks the validity of the memory word, records error information, and corrects single-bit errors. (Refer to "Error Checking and Correction" for ERCC detail.)

Table 2.3 summarizes the characteristics of the memory unit.

Characteristic	Description
Memory	
Type	Dynamic MOS n-channel RAM
Capacity	131,072; 262,144; or 524,288 bytes with error checking and correction
Cycle Time	
Read	500 nanoseconds
Write	500 nanoseconds

Table 2.3 Summary of memory unit characteristics

The main component of the memory unit is a RAM array containing one, two, or four banks of memory ICs. A 128-Kbyte memory contains one bank (bank 0); a 256-Kbyte memory contains two banks (banks 0 and 2); and a 512-Kbyte memory contains four banks (banks 0, 1, 2 and 3). Each bank contains 22 64K by 1-bit RAM ICs.

Figure 2.17 shows the interconnection of the RAM array and the other components comprising the memory unit, as well as the various signals that control the flow of data between the memory bus and the RAM array.

Initiating a Memory Operation

The memory unit continually receives information from the memory bus. At the start of a memory operation, the processor or data channel interface asserts the **MEMCYC** and $\overline{\textbf{ADREN}}$ signals and sends a 16-bit address over the memory bus. When system memory space is referenced, address bit 0 is set to 0 and address bits 1 through 15 contain a logical memory address. Address bits 1 through 5 (logical page number) are applied to the memory allocation and protection (MAP) unit for translation to a physical page number. (Refer to "Memory Allocation and Protection" for translation detail.)

The physical page number ($\overline{\textbf{RAMD<6-15>}}$), and address bits 6 through 15 (**MB<6-15>**), applied by the MAP unit and requesting logic, respectively, are gated into the memory address register ($\overline{\textbf{LXADR<0-4>}}$ and **LMA<1-15>**).The memory address register content is latched on the trailing edge of **TTL01.** At the same time, memory control signals (**MEMCYC**, $\overline{\textbf{WH}}$, $\overline{\textbf{WL}}$, and $\overline{\textbf{MAPEN}}$) from the processor or data channel interface and the memory protection bits ($\overline{\textbf{RAMD<0-1>}}$) from the MAP are latched in a control logic buffer. The 16-bit address contained on the memory bus is also latched in the A register (**A<0-15>**).

The content of the memory address register is applied to the bank selection logic and the row and column address selector. In addition, the memory address register content is applied to the ERCC facility for recording any memory error detected during the operation.

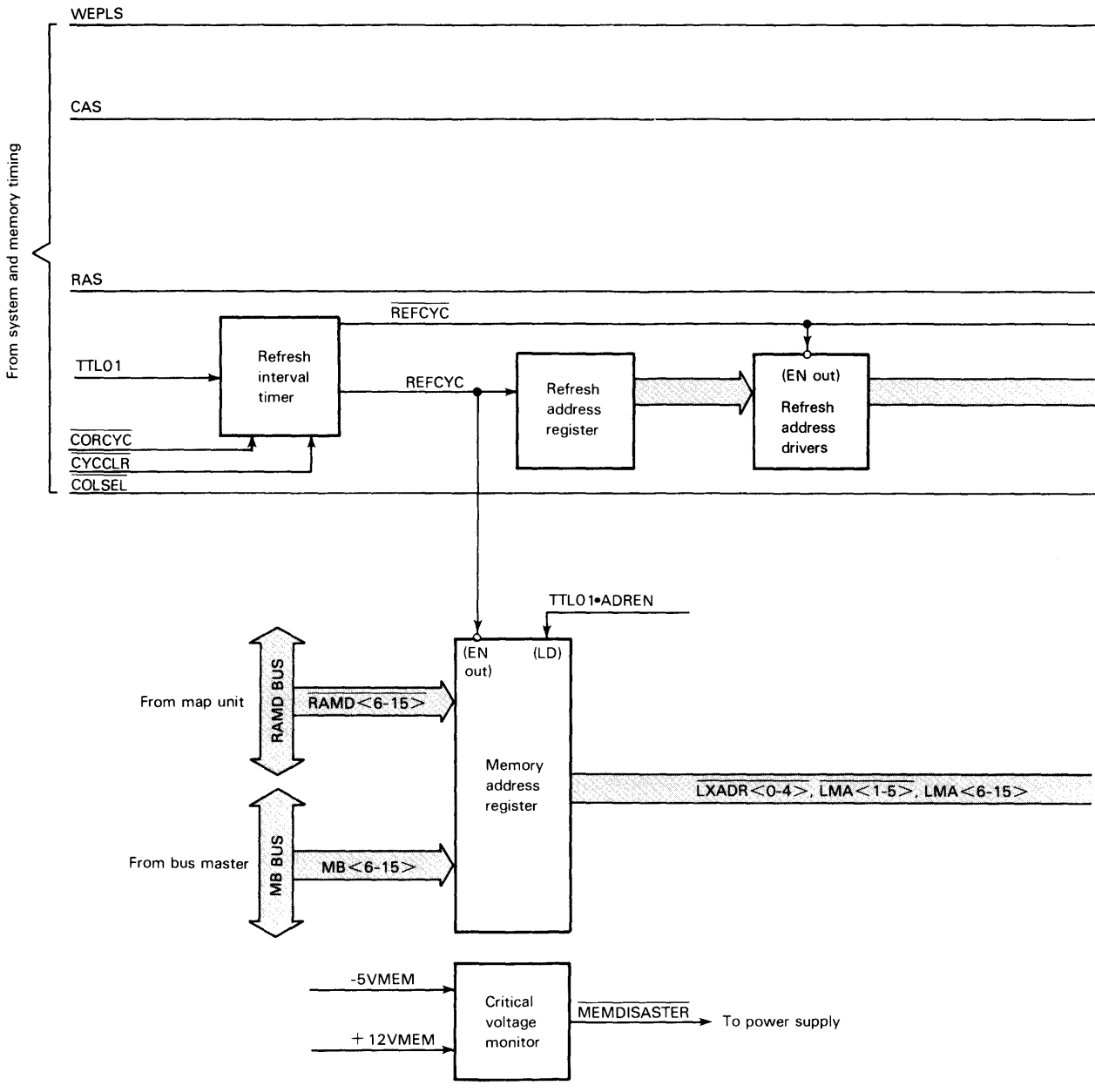
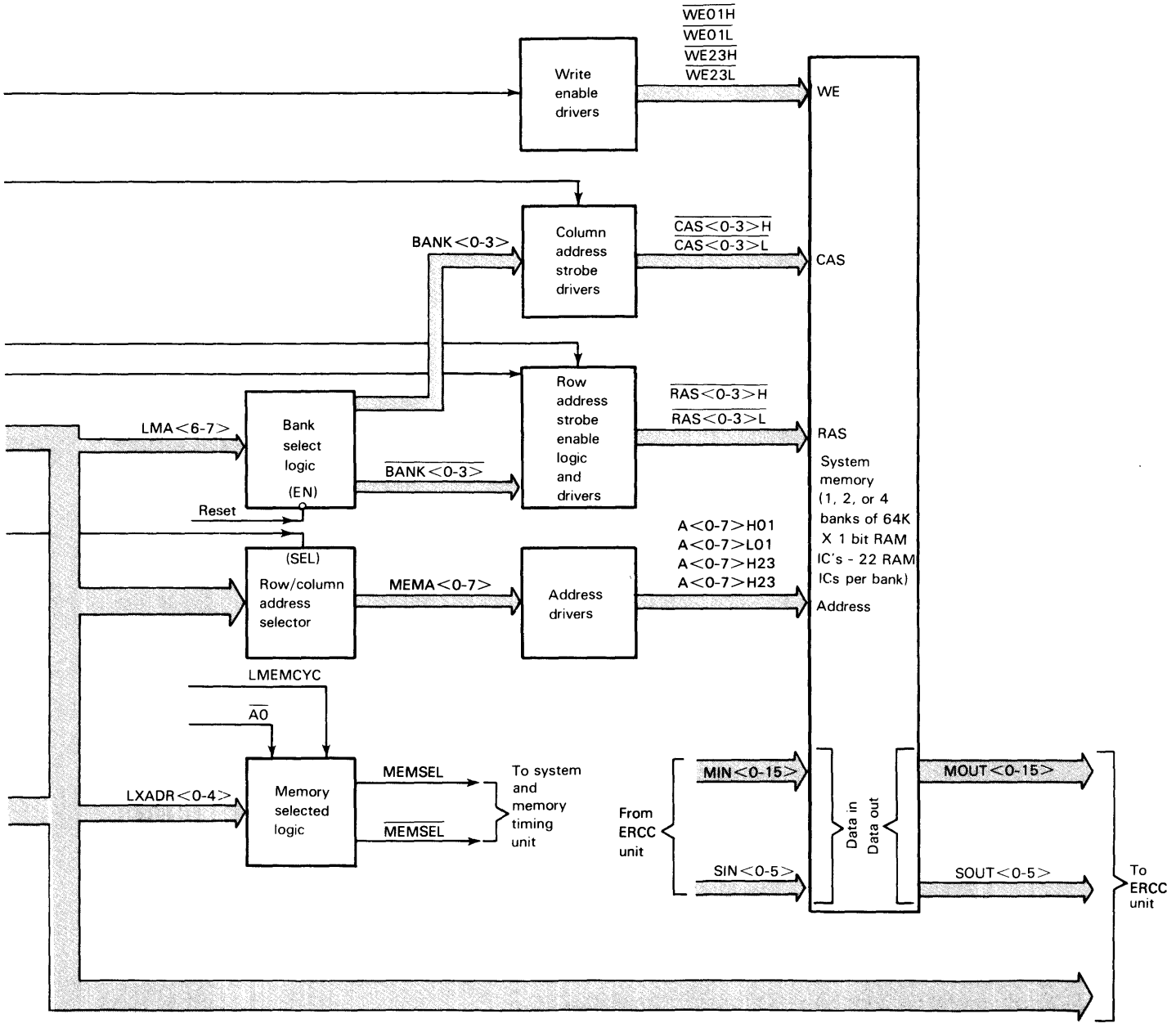


Figure 2.17 Memory unit


WE01H
WE01L
WE23H
WE23L
Write enable drivers
WE
BANK<0-3>
Column address strobe drivers
CAS<0-3>H
CAS<0-3>L
CAS
Row address strobe enable logic and drivers
RAS<0-3>H
RAS<0-3>L
RAS
LMA<6-7>
Bank select logic
(EN)
BANK<0-3>
Reset
System memory (1, 2, or 4 banks of 64K X 1 bit RAM IC's - 22 RAM ICs per bank)
(SEL)
Row/column address selector
MEMA<0-7>
Address drivers
A<0-7>H01
A<0-7>L01
A<0-7>H23
A<0-7>H23
Address
LMEMCYC
A0
LXADR<0-4>
Memory selected logic
MEMSEL
MEMSEL
To system and memory timing unit
From ERCC unit
MIN<0-15>
SIN<0-5>
Data in
Data out
MOUT<0-15>
SOUT<0-5>
To ERCC unit

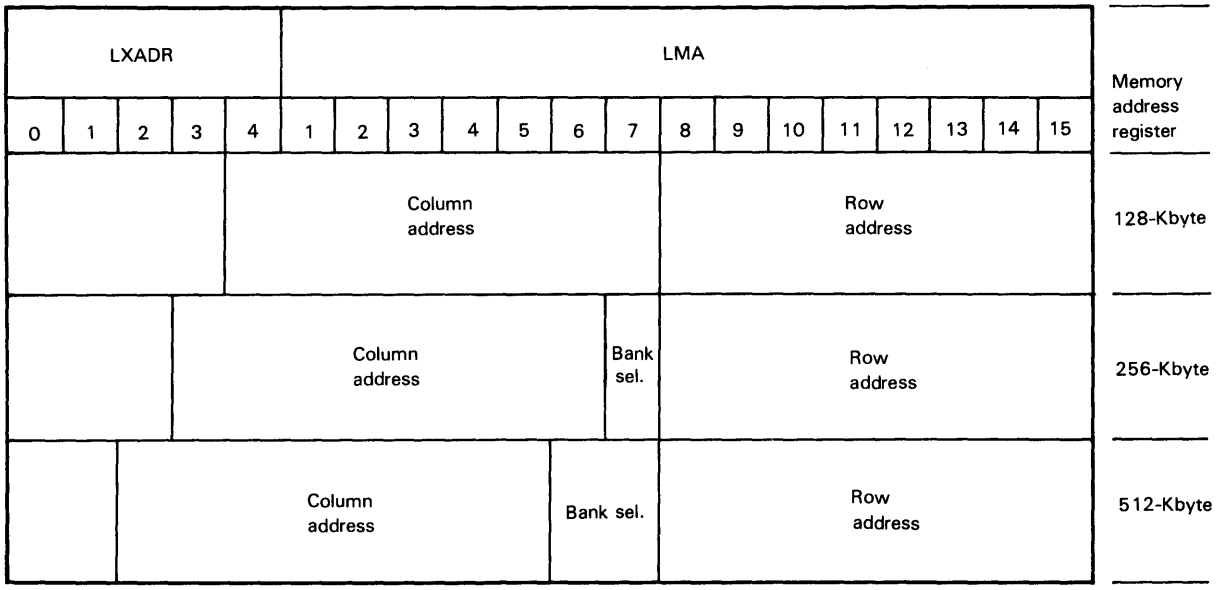


Figure 2.18 Memory address selection

Figure 2.18 details how the memory address register content selects a memory location for each memory capacity.

The memory select logic asserts **MEMSEL** and **MEMSEL**, provided **LXADR<0-4>** indicate the memory address is within the range of implemented memory. The bank select logic decodes **LMA6** and/or **LMA7** to choose the appropriate bank. On 128-Kbyte memory units, bank 0 is always selected; on 256-Kbyte memory units, **LMA7** selects one of two banks (bank 0 or 2); on 512-Kbyte memory units, **LMA<6-7>** selects one of four banks (banks 0, 1, 2, or 3).

The system and memory timing unit initiates a memory operation as soon as it receives **ADREN**, applying the appropriate timing control signals (**RAS**, **COLSEL**, **CAS** and **WEPLS**) to the memory unit.

Row and Column Address Selection

When the address is applied to the row and address selector, the selector applies the appropriate row address bits to the RAM array via the memory address drivers and their associated signal lines (**A<0-7>H01, A<0-7>L01, A<0-7>H23,** and **A<0-7>L23**). Signal lines **A<0-7>H01** are applied to the data bits 0 through 10 RAMs, while **A<0-7>L01** are applied to the data bits 11 through 15 and check bits 0 through 5 RAMs of banks 0 and 1. Signal lines **A<0-7>H23** and **A<0-7>L23** are applied to the corresponding RAM bit positions of banks 2 and 3.

When the system and memory timing unit asserts **RAS**, the row address strobe for the selected memory bank (**RAS0<H&L>**, **RAS1<H&L>**, **RAS2<H&L>**, or **RAS3<H&L>**) is driven low, latching the row address into the appropriate RAMs. Signal line **RAS0H** drives data bits 0 through 10, while **RAS0L** drives data bits 11 through 15 and check bits 0 through 5 when bank 0 is addressed. Signal lines **RAS1<H&L>**, **RAS2<H&L>**, or **RAS3<H&L>** drive the corresponding bits when bank 1, 2, or 3 is addressed.

When system and memory timing asserts **COLSEL**, the row and column address selector applies the appropriate column address bits to the RAM array via the memory address drivers and their associated signal lines. When system timing asserts **CAS**, the column address strobe for the selected memory bank (**CAS0<H&L>**, **CAS1<H&L>**, **CAS2<H&L>**, or **CAS3<H&L>**) is driven low, latching the column address into the RAM array.

System Memory Operations

The system memory read and write operations are discussed and illustrated in the next two sections.

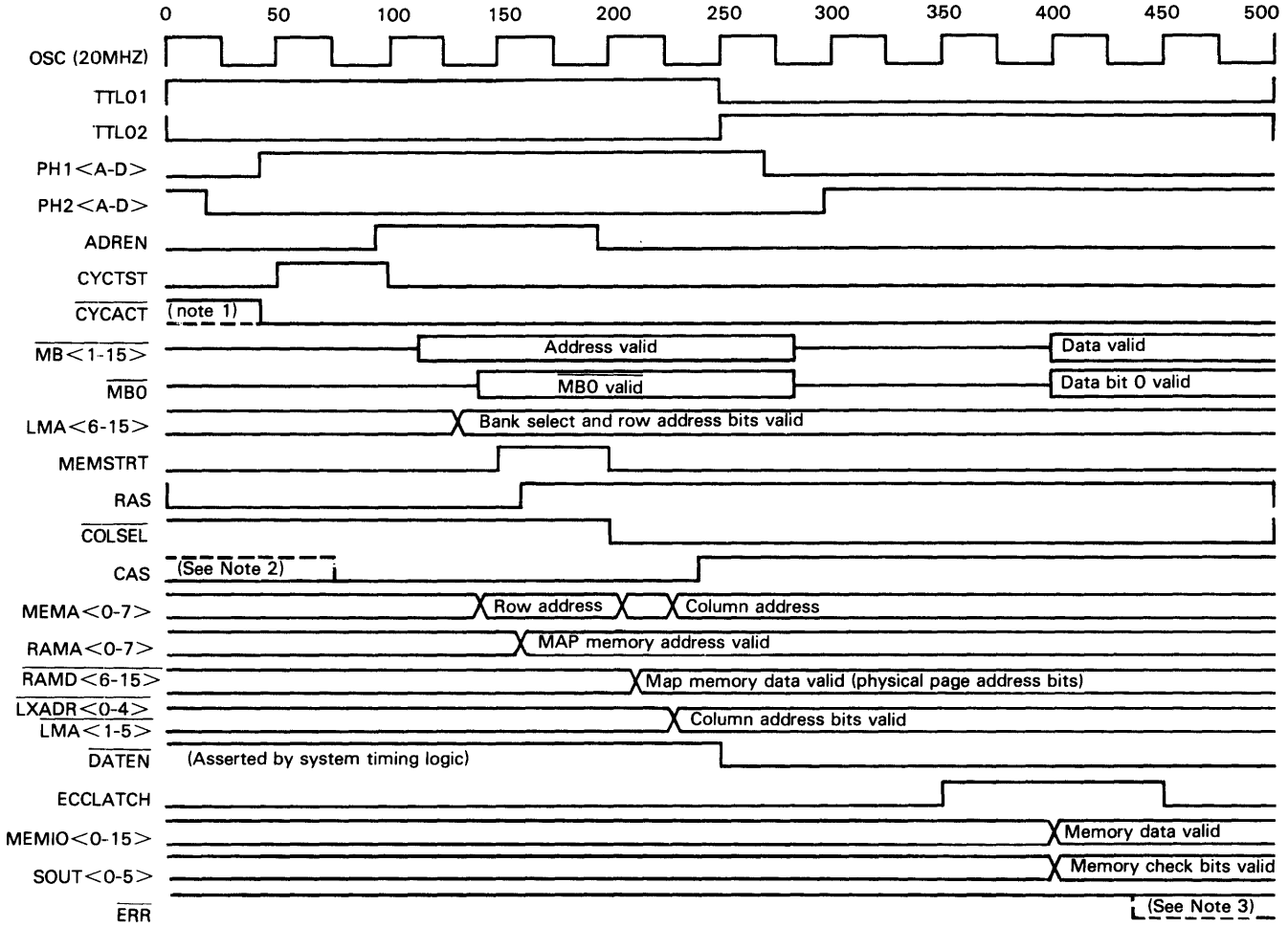
Read Operation

If **WH** and/or **WL** are not asserted during the address phase (**ADREN** asserted), memory initiates a read operation. After the row and column addresses are multiplexed to the RAM array, the selected RAM bank outputs the contents of the addressed location. The data bits (**MOUT<0-15>**) and the check bits (**SOUT<0-5>**) are applied to the ERCC facility. If error correction is enabled, the ERCC component checks the memory word (22 bits) and generates an error code if it is erroneous. If no error is detected, the ERCC facility applies the unaltered data bits to the memory bus drivers.

Provided a validity-protected page was not addressed (**LVALERR** is high), the data bits are output to the memory bus (**MB<0-15>**) during the data phase (**DATEN**) of a system memory-read reference. **DATEN** originates from the system and memory-timing unit during system memory-read operations.

If an error is detected during the read operation and no validity-protected page was accessed, the ERCC error flag (**ERCCERR**) is set. In addition, the generated error code and the physical memory address containing the error are stored in the memory fault registers for later transfer to the processor. An ERCC error interrupts the processor (if enabled). The error also causes system timing to initiate a correction cycle to write the corrected data bits and a new check code into the addressed memory location. Then the corrected data word is applied to the memory bus drivers for transfer to the requesting user. Detected multi-bit errors are reported (**ERCCERR** is set) and subjected to the correction cycle. However, the uncorrected word is retained in memory. (**WEPLS** is inhibited.)

Figures 2.19 and 2.20 show the timing for a normal read operation and a read operation with a pended correction cycle, respectively.



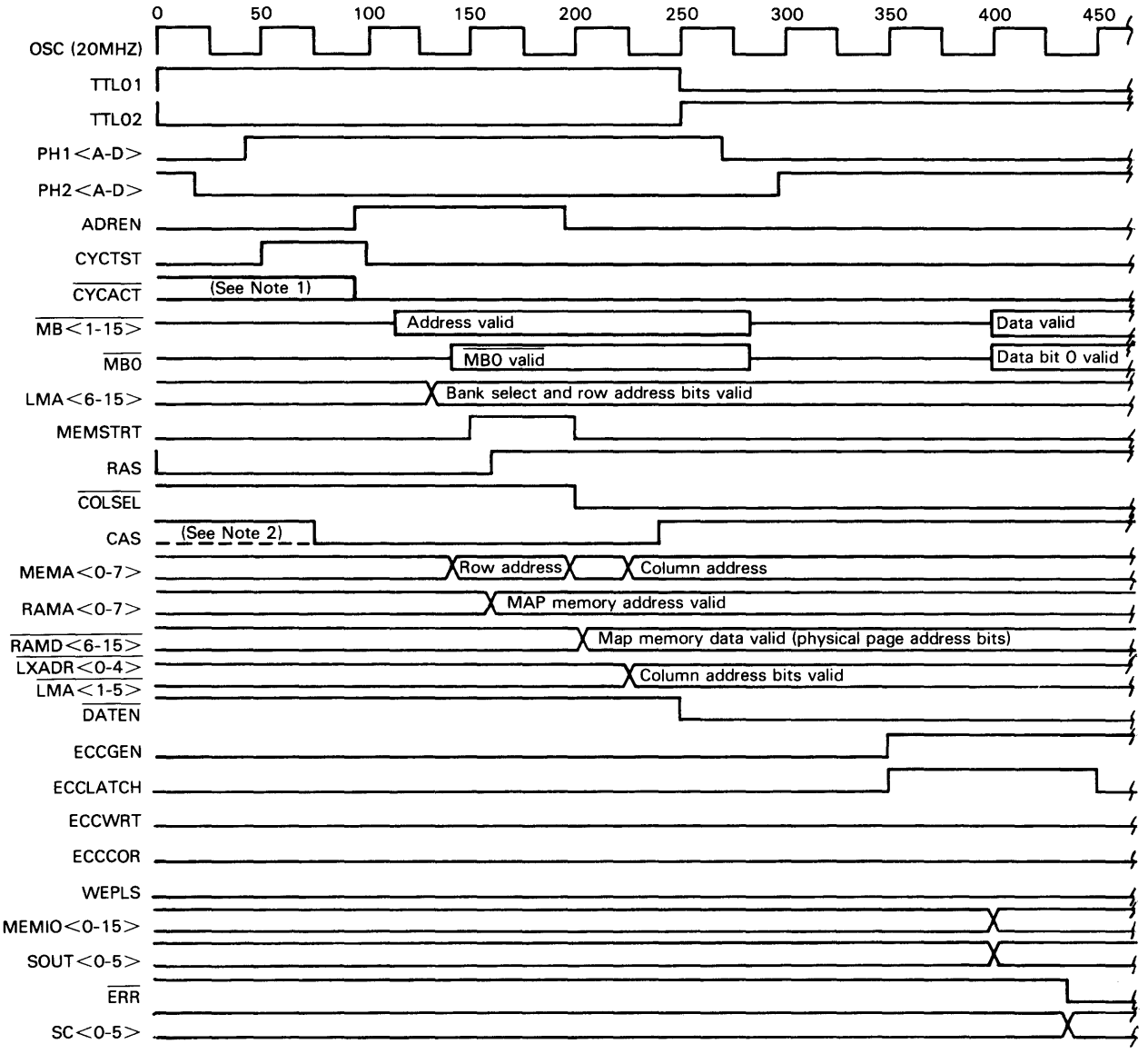
NOTES: *1. CYCACT will remain asserted if previous cycle was a bus transaction and was extended.*

2. CAS is still asserted at this time if previous cycle was a system memory access.

3. ERR will be asserted if a system memory error was detected. ERR will cause a correction cycle to be initiated.

DG-09057

Figure 2.19 System memory read cycle

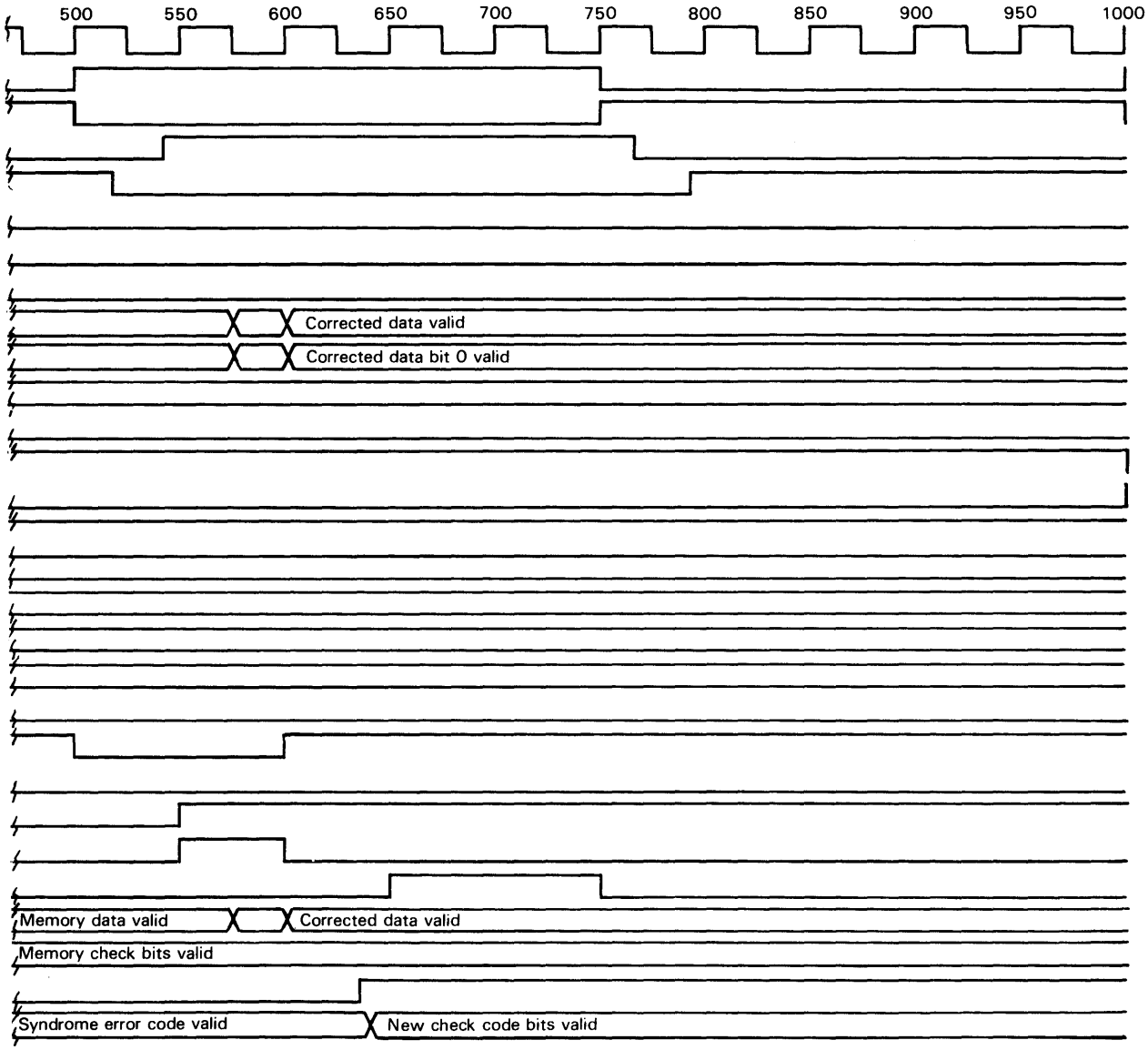


NOTES: *1.* CYCACT *will remain asserted if previous cycle was a bus transaction and was extended.*

2. CAS *is still asserted at this time if previous cycle was a system memory access.*

DG-09058

Figure 2.20 System memory read with a pended correction cycle


500
550
600
650
700
750
800
850
900
950
1000
Corrected data valid
Corrected data bit 0 valid
Memory data valid
Corrected data valid
Memory check bits valid
Syndrome error code valid
New check code bits valid

Write Operation

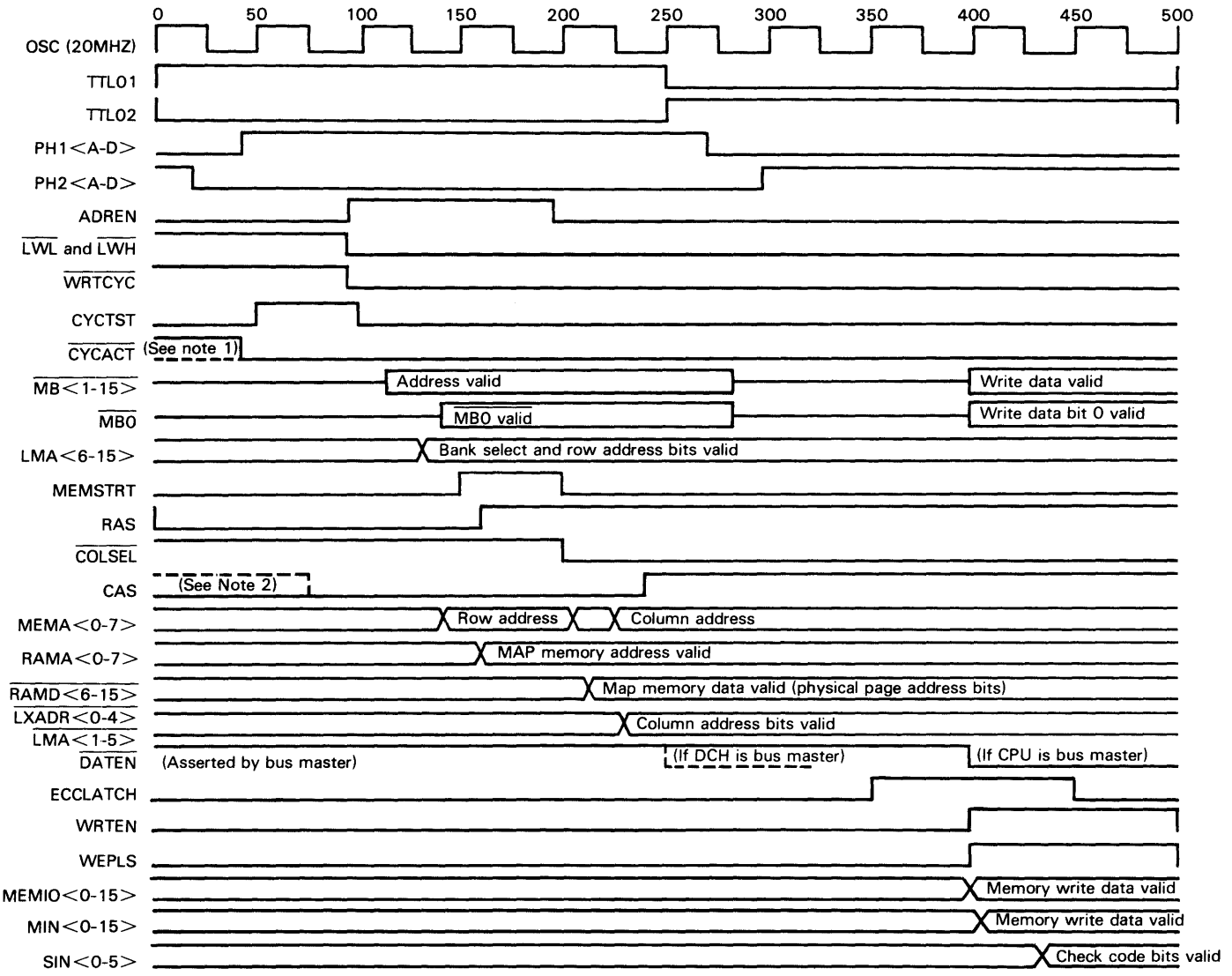
If $\overline{\text{WH}}$ and/or $\overline{\text{WL}}$ are asserted during the address phase ($\overline{\text{ADREN}}$ asserted), memory initiates a write operation. The S/120 memory always writes a word to memory. Therefore, when only one byte is to be written, the other (unchanged) byte must be retained. One-byte writes are performed by reading the addressed location, correcting any single-bit error, and writing the unchanged data byte along with the changed byte. Six new check bits are then calculated and appended to the word.

After the row and column addresses are multiplexed to the RAM array, the selected RAM bank outputs the contents of the addressed location. The data bits (**MOUT<0-15>**) received from the RAM array are applied to the ERCC facility.

During the data phase of a memory-write operation, the data to be written is received from the memory bus (**MB<O-15>**) and also applied to the ERCC facility. Depending on the states of $\overline{\text{WH}}$ and $\overline{\text{WL}}$, the ERCC facility applies the appropriate data bytes to the error detection/correction (EDC) component and to the memory data input signal lines, **MIN<0-15>**. (Refer to "Error Checking and Correction" for more details.)

If a byte is to be retained and correction is required, the EDC component corrects a single-bit error in the byte being retained. It also constructs and outputs the six check bits for the word to be written. These check bits are also applied to the memory data input signal lines, **SIN<0-5>**. When system timing asserts write pulse (**WEPLS**), the memory control logic drives $\overline{\text{WE01H}}$, $\overline{\text{WE01L}}$, $\overline{\text{WE23H}}$ and $\overline{\text{WE23L}}$ to write the entire word and its associated check bits into the addressed RAM location. $\overline{\text{WE01H}}$ writes data bits 0 through 10. When bank 0 or 1 is addressed, $\overline{\text{WE01L}}$ writes data bits 11 through 15 and check bits 0 through 5. When bank 2 or 3 is addressed, $\overline{\text{WE23H}}$ and $\overline{\text{WE23L}}$ writes the corresponding bits.

Figure 2.21 shows the timing for a write operation in which both bytes are written. Figure 2.22 shows the timing when one byte is retained.

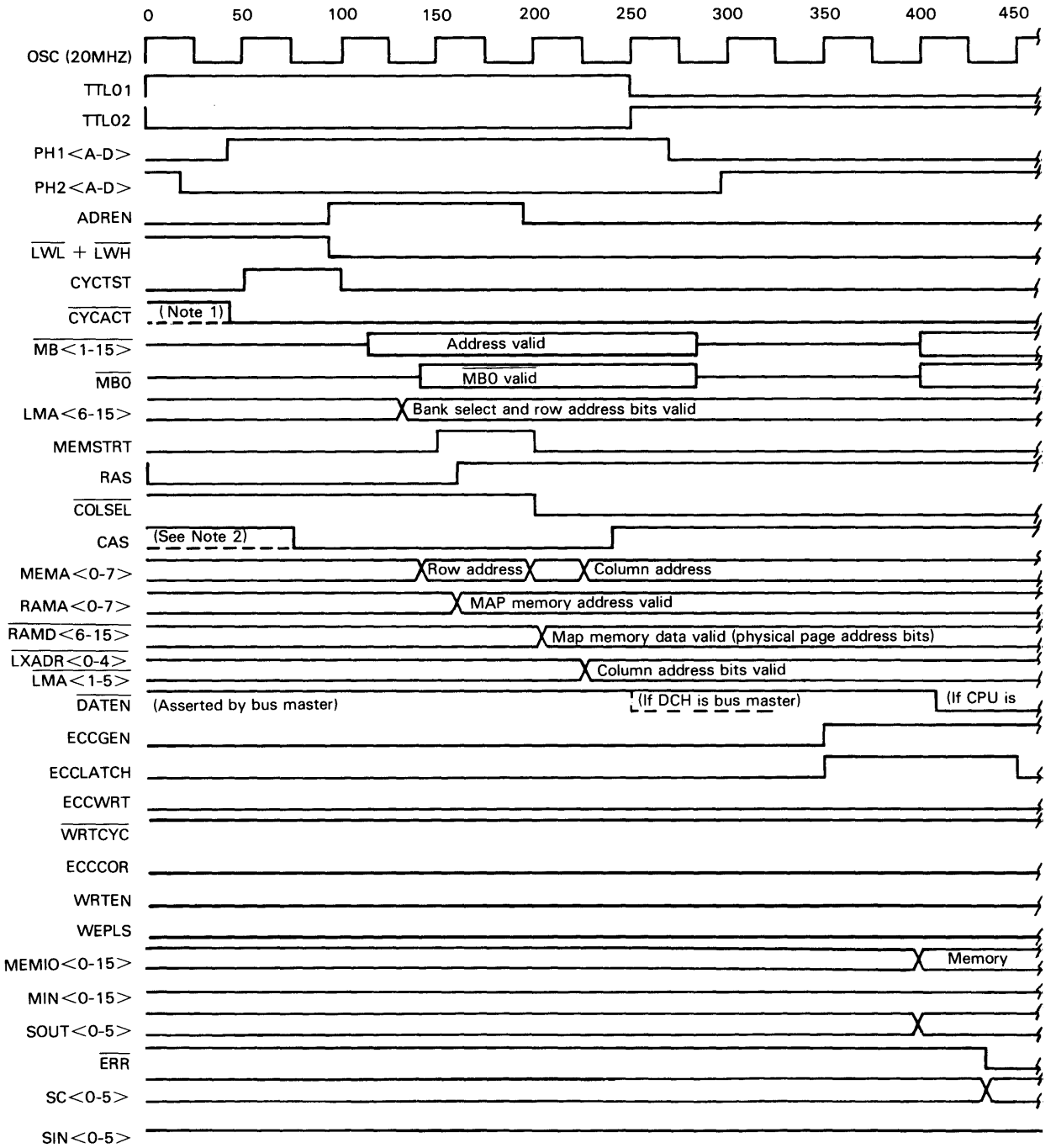


NOTES: *1.* $\overline{\text{CYCACT}}$ *will remain asserted if previous cycle was a bus transaction and was extended.*

2. **CAS** *is still asserted at this time if previous cycle was a system memory access.*

DG-09059

Figure 2.21 System memory word write cycle



NOTES: 1. CYCACT *will remain asserted if previous cycle was a bus transaction and was extended.*

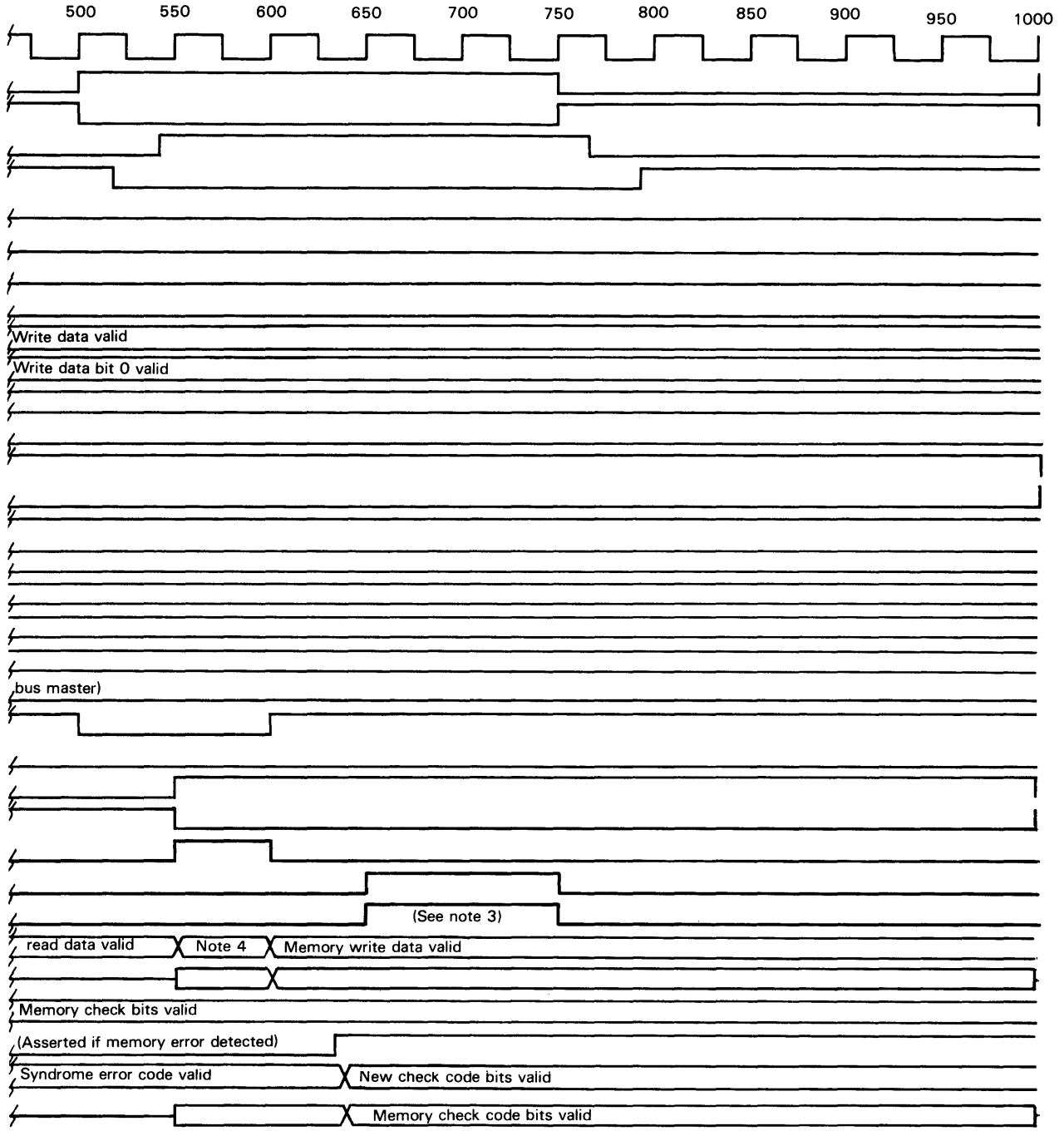
2. CAS *is still asserted at this time if previous cycle was a system memory access.*

3. WEPLS *will not be asserted at this time if a multiple bit error was detected.*

4. *Byte to be written is applied to* MEMIO *bus at this time.*

DG-09060

Figure 2.22 System memory byte write cycle


500
550
600
650
700
750
800
850
900
950
1000
Write data valid
Write data bit 0 valid
bus master)
(See note 3)
read data valid
Note 4
Memory write data valid
Memory check bits valid
(Asserted if memory error detected)
Syndrome error code valid
New check code bits valid
Memory check code bits valid

Refresh and Correction Cycle

The system memory refresh and correction cycles are discussed in the next two sections.

Refresh Cycle

The refresh interval timer asserts the **REFCYC** signal every 15.6 microseconds, directing memory to initiate a refresh cycle. During each refresh cycle, the 256 columns of the addressed row of all RAM ICs contained in memory are refreshed. In addition, if the error checking and correction (ERCC) facility is enabled, the accuracy of the data word addressed by the content of the refresh address counter is verified.

System timing initiates a refresh cycle upon receiving **REFCYC** and applies the appropriate timing control signals to the memory unit. A refresh cycle is similar to a read operation. During a refresh cycle, the content of the refresh address counter drive the memory address lines (**MEMA<0-7>**) via the row/column address selector. Row address strobe (**RAS**) is applied to all memory banks, refreshing the addressing rows (128 or 256 columns) of all RAM ICs contained in memory during one cycle. Column address strobe (**CAS**) is applied only to the addressed bank. Thus, only the addressed bank outputs a data word to the ERCC facility.

When the refresh cycle is complete, memory control logic increments the refresh address counter to the next address. 256 refresh cycles, one occurring every 15.6 microseconds, are required to refresh all rows of the RAMs.

If the addressed memory word generates an ERCC error during the refresh cycle, system timing initiates a correction cycle immediately following the refresh cycle. The correction cycle writes the corrected word into the addressed location. Detected multiple-bit errors initiate the correction cycle; however, the uncorrected word is retained in memory. (**WEPLS** is inhibited.) ERCC errors detected during refresh are not reported to the processor.

When **ADREN** and **REFCYC** signals are asserted simultaneously, the result is a conflict between a memory read/write and a refresh cycle. Memory first performs the refresh cycle and then initiates the memory read/write operation. If a memory read/write operation is requested during a refresh, cycle the read/write operation pends until the refresh cycle is completed. In either case, the read/write address is latched in the memory address register and system and memory timing asserts **READY** low, extending the data transfer phase of the requested read/write operation until the refresh cycle is completed. Figures 2.23 and 2.24 show the timing for a refresh cycle and a refresh cycle with pended correction cycle, respectively.

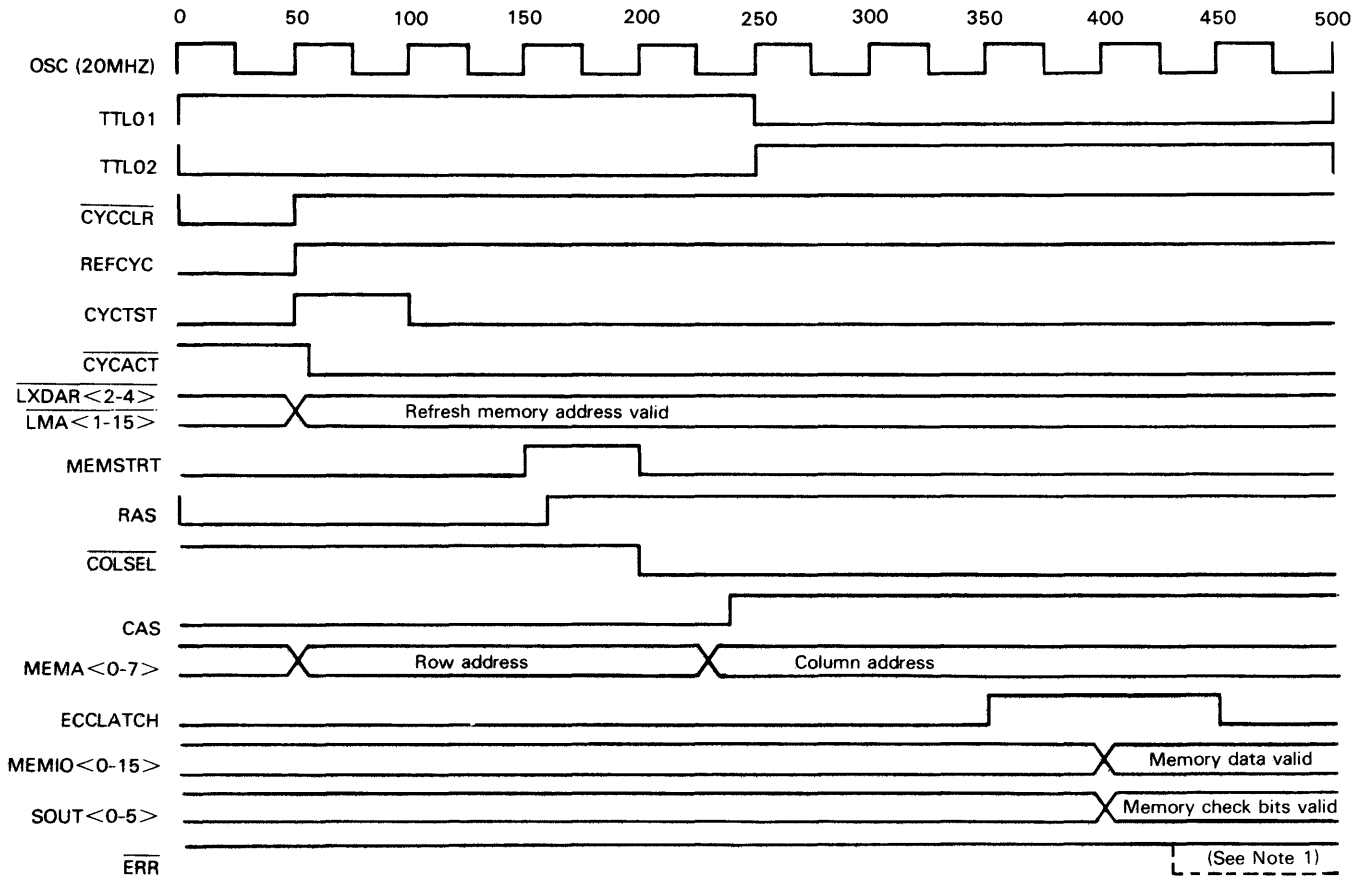
Correction Cycle

When the ERCC facility detects a single-bit error, the error flag is set and notification of the error is sent to system timing. System timing then appends a correction cycle to the memory cycle running when the error is detected. The address of the memory location containing the erroneous data remains latched in the memory address register for the correction cycle.

During the correction cycle, the EDC component corrects the erroneous data bit and calculates a new check code. Both the corrected data word and the new check code are applied to the memory data input signal lines and then written into memory when system and memory timing asserts write pulse (**WEPLS**).

Critical Voltage Monitor

This circuit monitors the −5 and +12 memory voltages (**−5VMEM** and **+12VMEM**). If any malfunction removes the -5 volts while the +12 volts are asserted, the signal **MEMDISASTER** is applied to, and shuts down, the power supply.



NOTE: *1.* ERR *will be asserted if a system memory error was detected.* ERR *will cause a correction cycle to be initiated.*

DG-09061

Figure 2.23 System memory refresh cycle

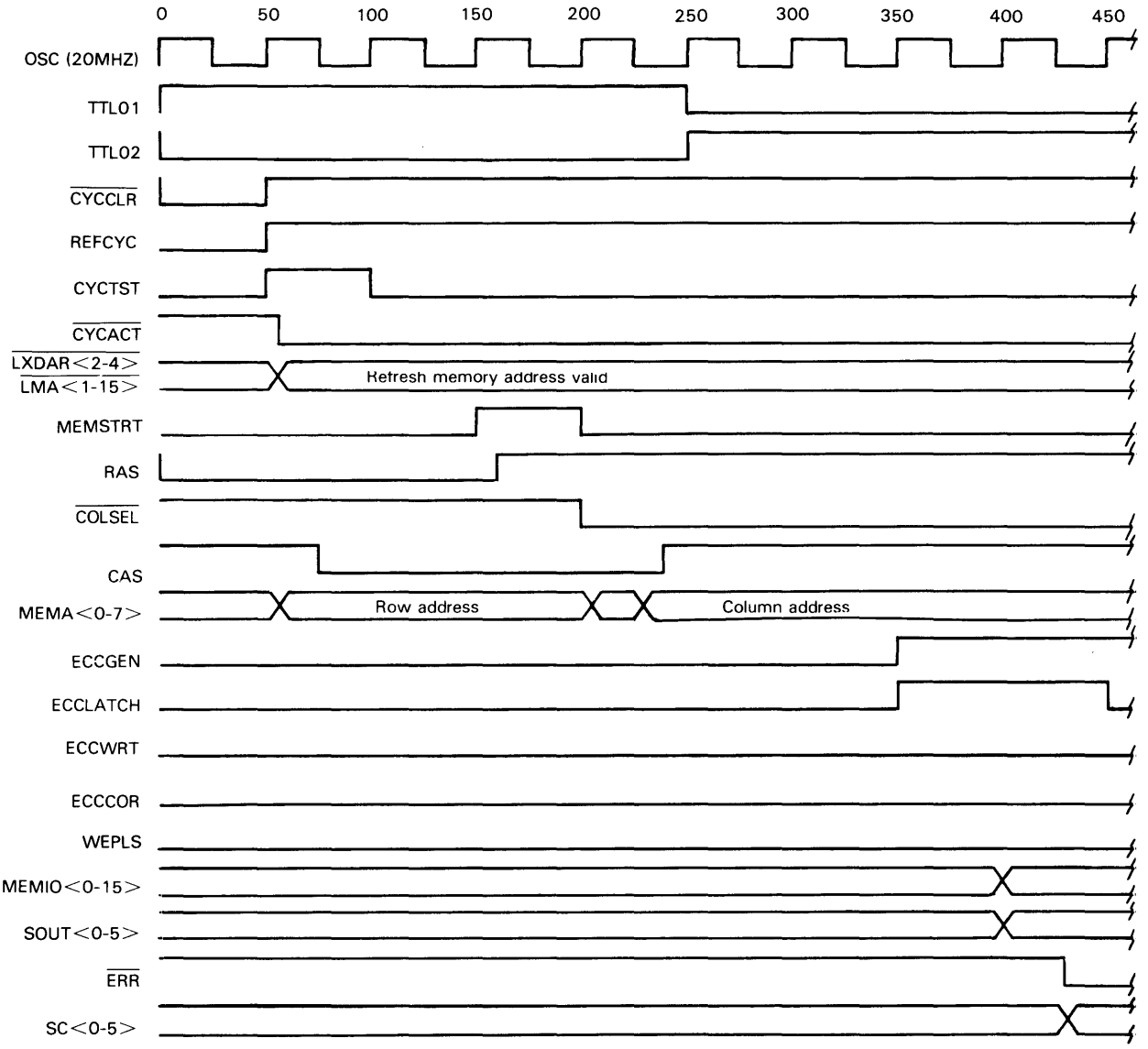
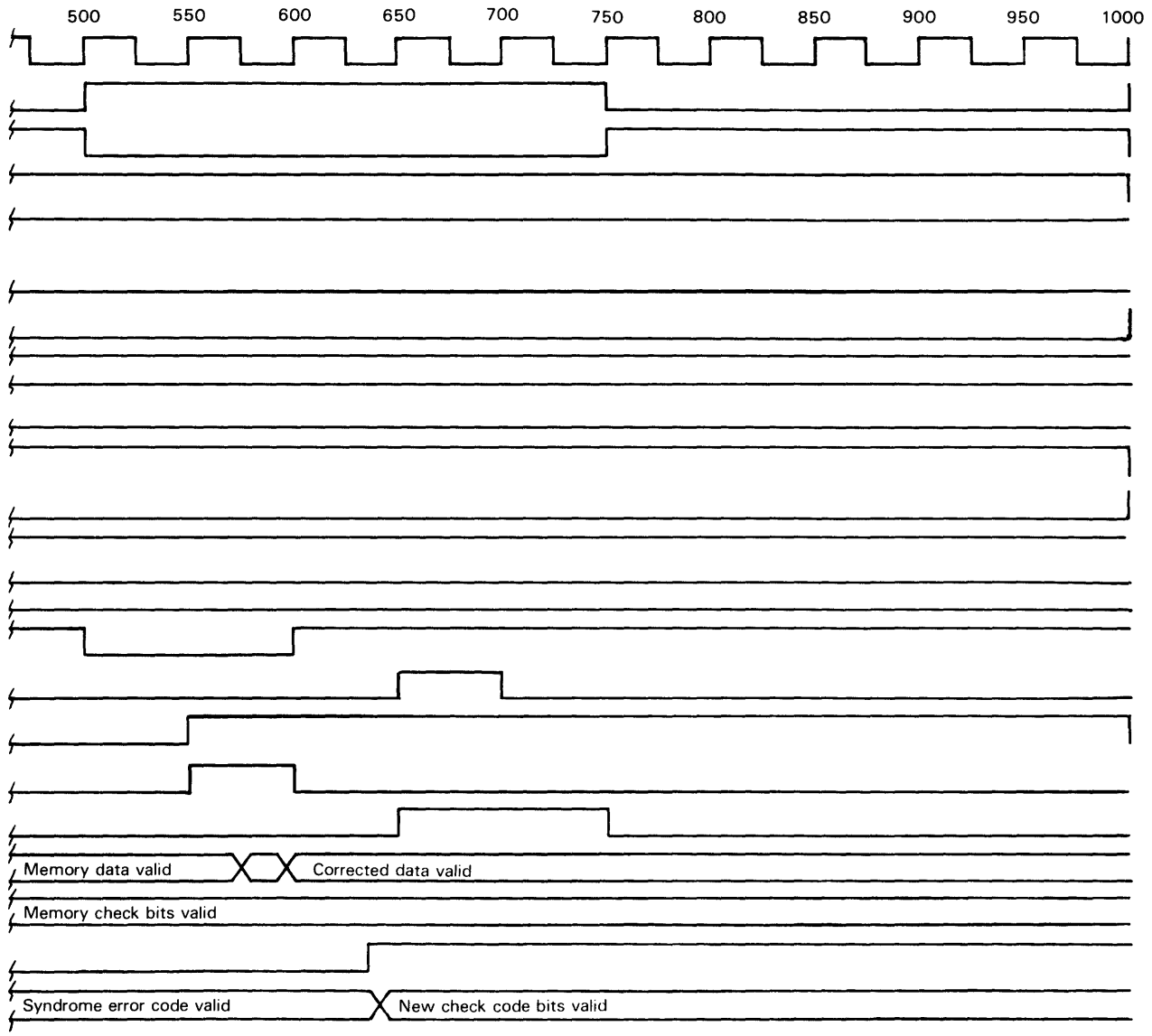


Figure 2.24 System memory refresh cycle with appended correction


500
550
600
650
700
750
800
850
900
950
1000
Memory data valid
Corrected data valid
Memory check bits valid
Syndrome error code valid
New check code bits valid

NOVA/ECLIPSE Input/Output Interface

The NOVA/ECLIPSE input/output interface performs the control, timing, and buffering necessary to pass instructions and data from the

- SPU local busses to I/O devices connected to the S/120 system via the NOVA/ECLIPSE I/O bus.
- On-board I/O devices (MAP, ERCC, APL, and external SPU status register bits), virtual console, and the I/O devices connected to the NOVA/ECLIPSE I/O bus to the CPU bus for transfer to the CPU registers or the system memory.

The interface section performs both programmed input/output and data channel transfers. During programmed input/output, this section buffers the command pulse and the flag control pulse applied to the external I/O bus via the I/O decoder section. It also buffers the device address and output data being applied to the external I/O bus ($\overline{\text{DS}}$<0-5> and $\overline{\text{DATA}}$<0-15>)via the address bus (A<10-15>) and the buffered CPU bus (MB<0-15>).

As previously described, the CPU bus is a multimaster bus. The CPU either controls the bus or relinquishes control for data channel transfers. The I/O interface requests and obtains use of the CPU bus for data channel transfers as follows:

- $\overline{\text{RQENB}}$ is asserted periodically by the I/O interface to synchronize programmed input/output interrupts and data channel requests.
- $\overline{\text{DCHR}}$ is asserted by an I/O device to request a data channel operation.
- $\overline{\text{CNTRST}}$, the timing signal, clocks the request signal into a flip-flop, which then asserts $\overline{\text{DCREQ}}$.
- $\overline{\text{BREQ}}$ is asserted by data channel state machine to the CPU.
- If the CPU does not assert **BLOCK** and if **READY** is asserted, then **TTL02** asserts $\overline{\text{BUSOK}}$, which gives control to the data channel. (The CPU asserts $\overline{\text{BLOCK}}$ when it is engaged in an operation requiring uninterrupted bus accesses, such as the **ISZ** instruction; **READY** is pulled low by a device that has not completed the current data transfer.)

The data channel state machine removes $\overline{\text{BREQ}}$ to the CPU when the data channel transfer is complete. However, as the I/O interface completes the present data transfer, it issues another request enable ($\overline{\text{RQENB}}$) to enable data channel requests ($\overline{\text{DCHR}}$). If another data channel request is immediately applied, the state machine keeps $\overline{\text{BREQ}}$ active.

The I/O interface section, shown in Figure 2.25, consists of

bus interface
bus interface control logic
CPU bus control signal buffer
state machine
data channel address latch
data channel data latch

Figure 2.25 also shows one of the CPU bus input buffers. This buffer applies data received from the external I/O bus to the CPU bus and also applies data from the on-board I/O devices listed above and the virtual console.

Bus Interface

The bus interface, shown in Figure 2.25, consists of

- Bus buffers that transmit data, command pulses, flag control pulses, and data-channel control pulses to external I/O devices via the NOVA/ECLIPSE I/O bus.
- Bus buffers that apply data to the local I/O bus ($\overline{\text{IOOUT}}$<0-15>). This data is received from external I/O devices during programmed input/output or data channel input data transfers via the external I/O bus ($\overline{\text{DATA}}$<0-15>).
- Data bit selector which selects either external I/O bus data bits 0 and 1 ($\overline{\text{DATA}}$<0-1>) or the external done ($\overline{\text{SELD}}$) and busy ($\overline{\text{SELB}}$) flags to be applied to the local I/O bus ($\overline{\text{IOOUT}}$<0-1>). The external busy and done flags are selected when an *I/O Skip* command is executed.
- Bus buffers and latches that receive and/or latch data channel control signals received from external I/O devices.

Data to be transmitted from the SPU to external I/O devices is input to the bus interface via the buffered CPU bus (MB<0-15>) and applied to the external I/O bus ($\overline{\text{DATA}}$<0-15>). Data received from the external I/O bus is sent to the CPU bus ($\overline{\text{MB}}$<0-15>) via the local I/O bus ($\overline{\text{IOOUT}}$<0-15>).

The incoming data, contained on the local I/O bus, is applied to the CPU bus ($\overline{\text{MB}}$<0-15>) for transfer to a CPU accumulator during PIO transfers or the system memory during data-channel input transfers. These buffers also apply data being transmitted to external I/O devices during data channel output data transfers via the CPU bus and the buffered CPU bus (MB<0-15>) to the local I/O bus. The outgoing data is latched in the data channel data latch to extend the time the data appears on the external I/O bus. (Refer to "Data Channel Data Latch" later in this section).

Command and flag control pulses are input to the bus interface from the I/O decoder section. (Refer to "CPU Support" section of this chapter for a detailed description of the I/O decoder.)

State Machine

The state machine shown in the block diagram (Figure 2.25) coordinates the timing of data channel transfer system operations. These operations are completed within two cycles of the 4-MHz **2XCLK** signal, with I/O bus data channel operations. For input data channel transfers, I/O bus data channel operations take at least eight cycles of the **2XCLK** signal to be completed. For output data channel transfers at least ten cycles are required. When more than the minimum number of cycles of the **2XCLK** signal are required, sets of 2 cycles are added. The actual number of cycles run depends on the time required to obtain use of the CPU bus and upon whether a memory refresh or correction cycle is required during the data channel transfer. However once a data channel transfer has been granted use of the CPU bus, additional data channel transfers (if any) can be processed in four additional cycles of the **2XCLK** signal for input transfers or six cycles for output transfers.

The state machine is driven by the **2XCLK** signal and control logic shown in Figure 2.3, which is produced by the system timing. The conditions of nine signals determine the states of this machine.

- **TTL01**, the system timing signal which was shown in Figure 2.5.
- **DCREQ** and **DCHINP**, two I/O control signals which specify a data channel transfer is required and the direction of the transfer. These signals are derived from the I/O bus control signals, **DCHR** and **DCHM0**, respectively.
- **INTACYC**, a control signal which temporarily delays the data channel operation in the event an *Interrupt Acknowledge* instruction is executing.
- **BUSOK**, a bus arbitration signal which indicates the I/O interface can use the CPU bus, beginning with the present bus cycle, for a data channel transfer.
- Four feedback signals from the state machine.

The outputs of the state machine are a system bus request signal, a system bus request-granted signal, four NOVA/ECLIPSE I/O bus control signals, two data-channel system bus control signals, four I/O interface control signals, and four state machine feedback signals.

The system bus request (**BREQ**) is applied to the CPU when the I/O interface requires the system bus to perform a data channel transfer.

The system bus request granted (**GRANT**) is asserted when the system bus can be used by the I/O interface for a data-channel transfer. This signal enables the I/O interface to drive the system bus control signals **WH**, **WL**, and **ADREN**. This signal is also applied to the MAP unit to allow selection of a data-channel map. (Refer to the "Memory Allocation and Protection" section in this chapter for a detailed description.

The four NOVA/ECLIPSE I/O bus control signals are **RQEN**, **DCA**, **DCHI**, and **DCHO**. Their functions are described in Appendix A, "Summary of I/O Bus Signals." **RQEN** occurs at a 4-MHz cycle rate until a data channel transfer bus cycle is granted. Once a transfer is initiated, **RQEN** is not cycled again until the transfer has completed. **DCA** is also used by the I/O interface to latch the data-channel memory address (**DATA<1-15>**), map selection signals (**DATA0** and **EXDCH**), and the direction of transfer signal (**DCHM0**) when they are returned from the device.

The data channel system bus control signals, **IOADREN** and **IODATEN**, perform the function of the corresponding CPU-issued system bus control signals when the data channel state machine takes control of the system bus for data channel transfers. **IOADREN** also applies the latched data channel memory address to the CPU bus during the address phase of the data-channel memory cycle. **IODATEN** also applies the latched data-channel data to the CPU bus during the data phase.

The four I/O interface control signals and their functions are

- **DLATCH**, which latches the data channel output data when it is applied on the I/O bus (**DATA<0-15>**) via the buffered CPU bus (**MB<0-15>**) during the memory read cycle.
- **DENABLE**, which applies the latched data channel output data to the I/O output bus (**IOOUT<0-15>**). This extends the time output data is available for transfer to the device via the external I/O bus (**DATA<0-15>**).
- **DCHDRV**, which applies the address and data via the I/O output bus (**IOOUT<0-15>**) to the CPU bus (**MB<0-15>**) during the address and data phase of data channel transfers.
- **BUSDRV**, which applies the output data via the buffered CPU bus (**MB<0-15>**) to the external I/O bus (**DATA<0-15>**) during the data phase of a data channel output transfer.

The loops of the data channel state machine are shown in summary form in Figure 2.26. Timing diagrams for data channel transfers are shown in Figures 2.27 and 2.28. These timing diagrams also indicate the content of the external I/O bus, the local I/O bus, and the CPU bus at specific times.

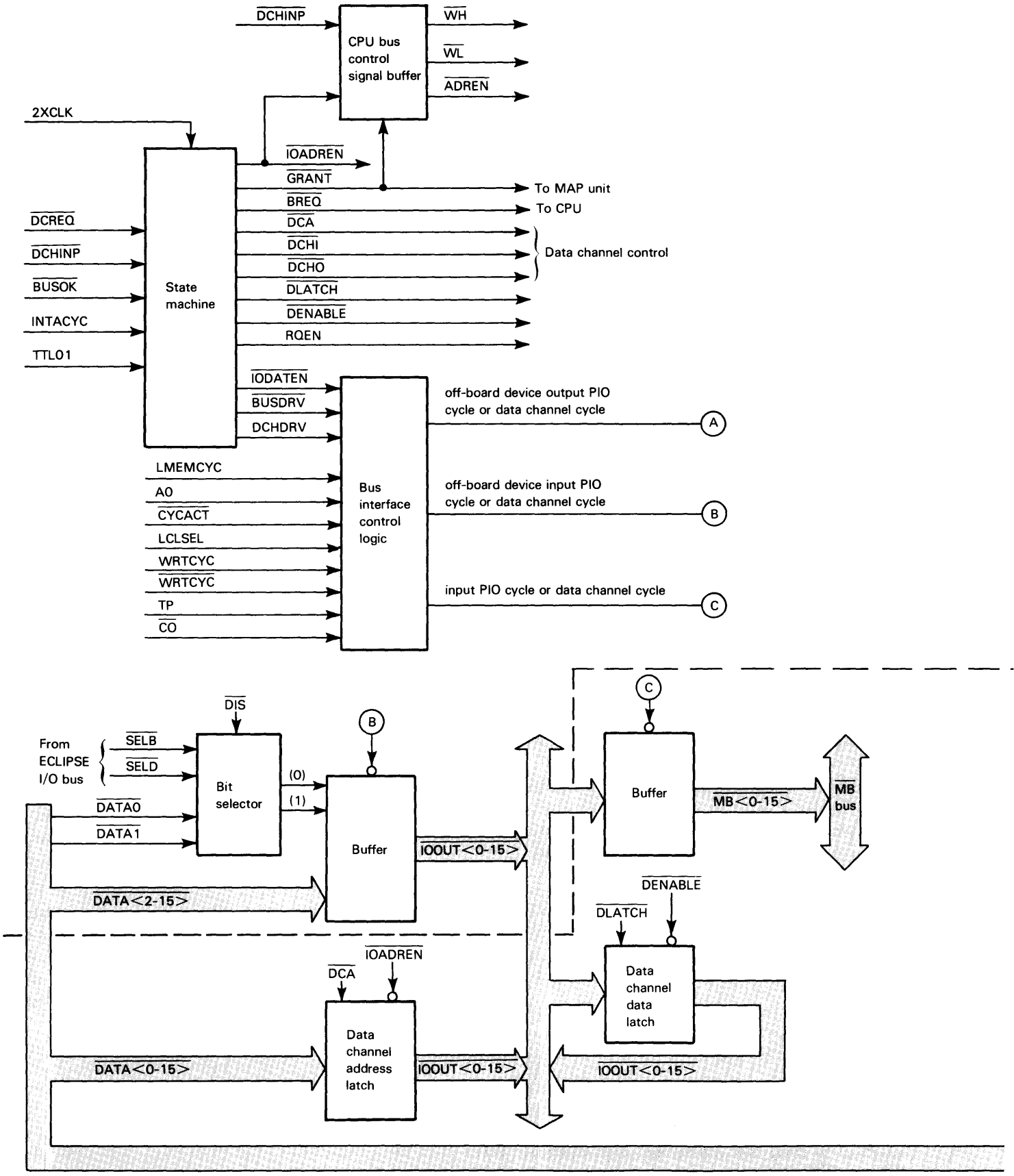
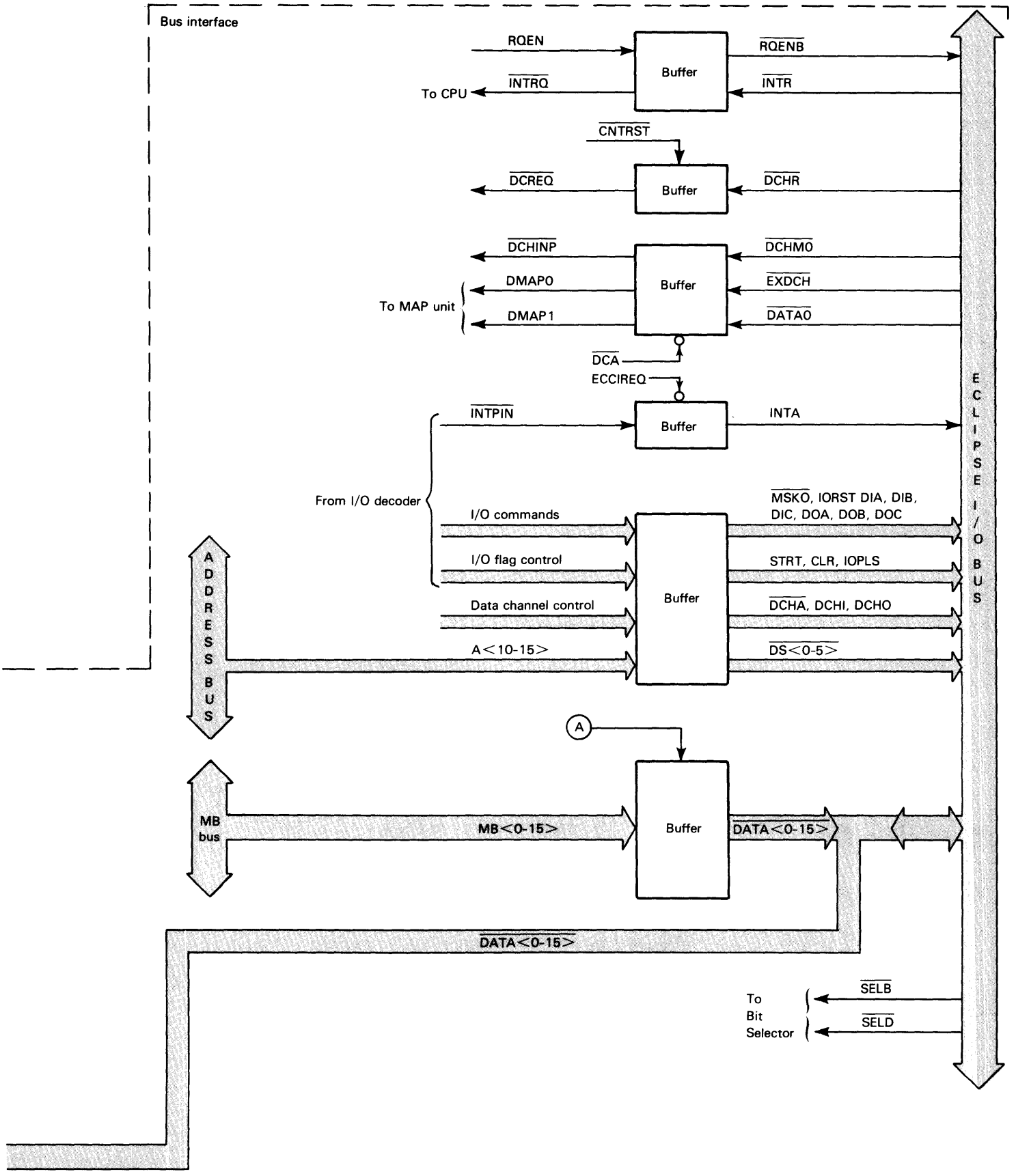
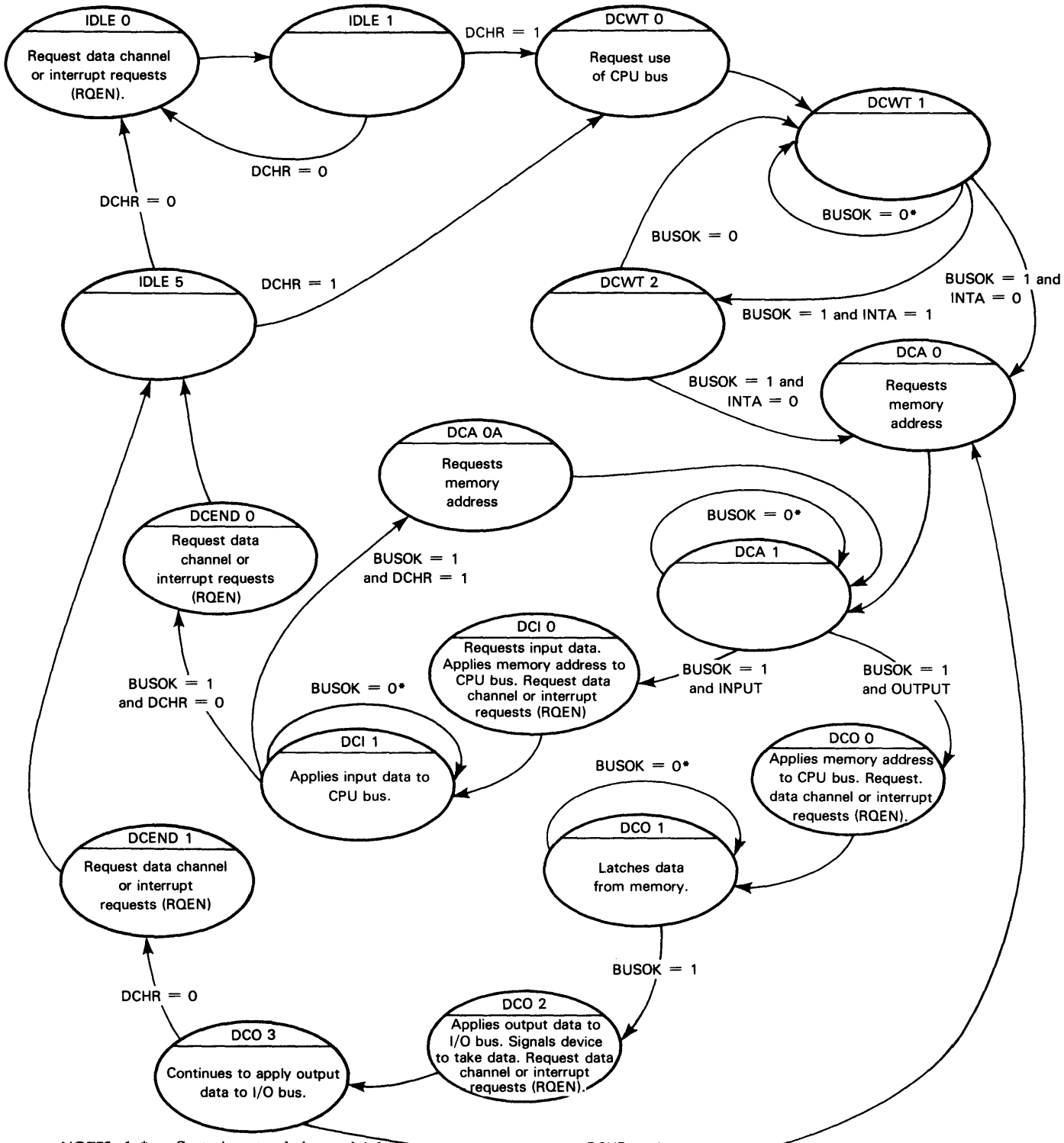


Figure 2.25 NOVA/ECLIPSE I/O interface


Bus interface
RQEN
RQENB
Buffer
To CPU
INTRQ
INTR
CNTRST
DCREQ
Buffer
DCHR
DCHINP
DCHM0
DMAP0
Buffer
EXDCH
To MAP unit
DMAP1
DATA0
DCA
ECCIREQ
INTPIN
Buffer
INTA
From I/O decoder
MSK0, IORST DIA, DIB, DIC, DOA, DOB, DOC
I/O commands
I/O flag control
STRT, CLR, IOPLS
Buffer
Data channel control
DCHA, DCHI, DCHO
A<10-15>
DS<0-5>
ADDRESS BUS
ECLIPSE I/O BUS
A
MB bus
MB<0-15>
Buffer
DATA<0-15>
DATA<0-15>
To Bit Selector
SELB
SELD

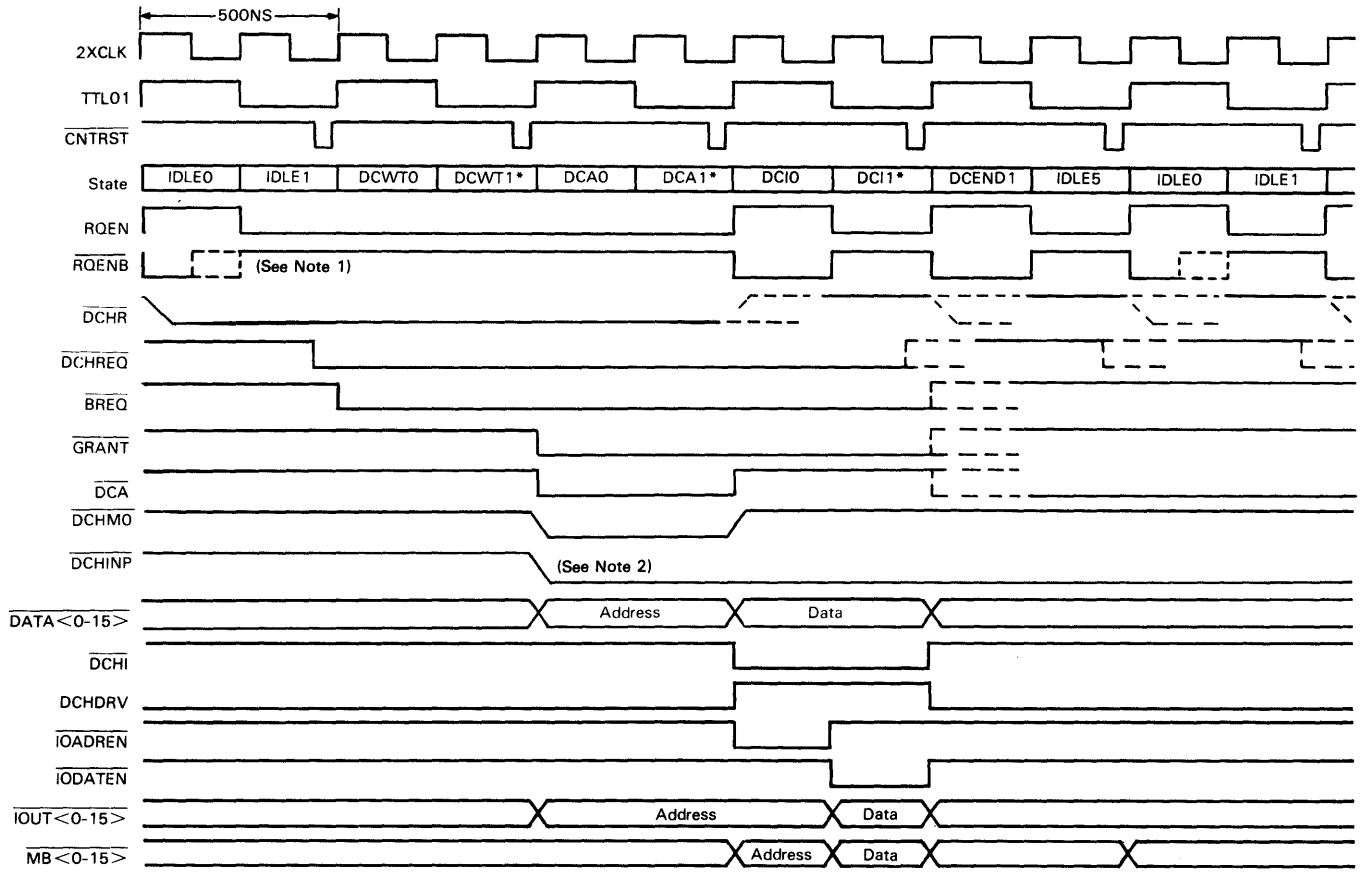


NOTES: *1. * = State is extended a multiple of 500 N seconds if CPU bus cannot be used.*

2. Each state is 250 N seconds in length unless extended as detailed in note 1.

DG-09064

Figure 2.26 Data channel state diagram

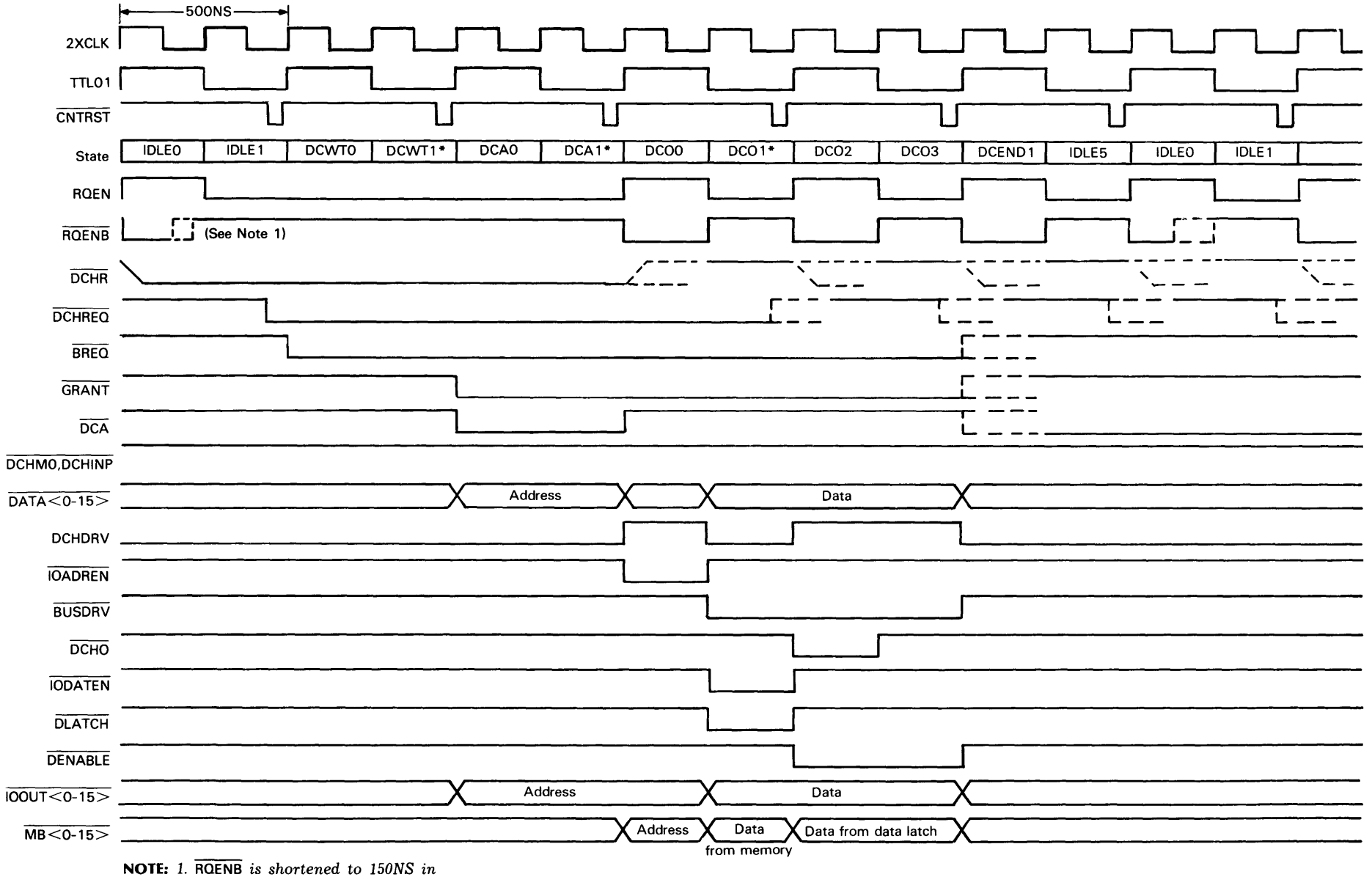


NOTES: *1.* $\overline{\text{RQENB}}$ *is shortened to 150NS in the IDLE0 state if this is the first T period of an I/O instruction.*

2. $\overline{\text{DCHNP}}$ *is latched until next* $\overline{\text{DCA}}$.

DG-09065

Figure 2.27 Data channel input transfer



NOTE: *1.* $\overline{\text{RQENB}}$ *is shortened to 150NS in the IDLE0 state if this is the first T period of an I/O instruction.*

DG-09066

Figure 2.28 Data channel output transfer

Bus Control Logic

This logic combines control signals applied by the CPU, SIO, system and memory timing, I/O decoder, and data channel state machine to apply data to system busses as required to transfer the data. The bus control logic enables the bus buffers as described below.

The external I/O data bus buffer is enabled during off-board device output programmed input/output transfers by the following signals:

LMEMCYC=0
A0=0
$\overline{\text{CYCACT}}$=1
LCLSEL=0
WRTCYC=1

The external I/O data bus buffer is also enabled during input and output data channel transfers by the following signal:

$\overline{\text{BUSDRV}}$=1

The local I/O data bus buffer is enabled during off-board device input programmed input/output transfers by the following signals:

$\overline{\text{LCLSEL}}$=0
$\overline{\text{WRTCYC}}$=0
LMEMCYC=0
A0=0
$\overline{\text{CYCACT}}$=1

The local I/O data bus buffer is also enabled during input and output data channel transfers by the following signal:

$\overline{\text{IODATEN}}$=1

The CPU data bus buffer is enabled during all input programmed input/output transfers by the following signals:

TP=1
$\overline{\text{CO}}$=0
$\overline{\text{WRTCYC}}$=0

The CPU data bus buffer is also enabled during input and output data channel transfers by the following signal:

DCHDRV=1

Data-Channel Address Latch

This latch receives the memory address requested during a data channel transfer sequence. The address is latched at the end of the data-channel acknowledge phase and then applied to the memory unit and the MAP unit via the local I/O bus (**$\overline{\text{IOOUT}}$<0-15>**), the CPU bus (**$\overline{\text{MB}}$<0-15>**), and the buffered CPU bus (**MB<0-15>**).

Data-Channel Data Latch

This latch receives data from memory during the data phase of a data-channel output phase via the memory input/output bus (**MEMIO<0-15>**), the CPU bus (**$\overline{\text{MB}}$<0-15>**), the buffered CPU bus (**MB<0-15>**), the external I/O bus (**$\overline{\text{DATA}}$<0-15>**), a bus buffer, and local I/O bus (**$\overline{\text{IOOUT}}$<0-15>**). The data is latched to extend the amount of time the data appears on the external I/O bus via local I/O bus, CPU bus, and buffered CPU bus.

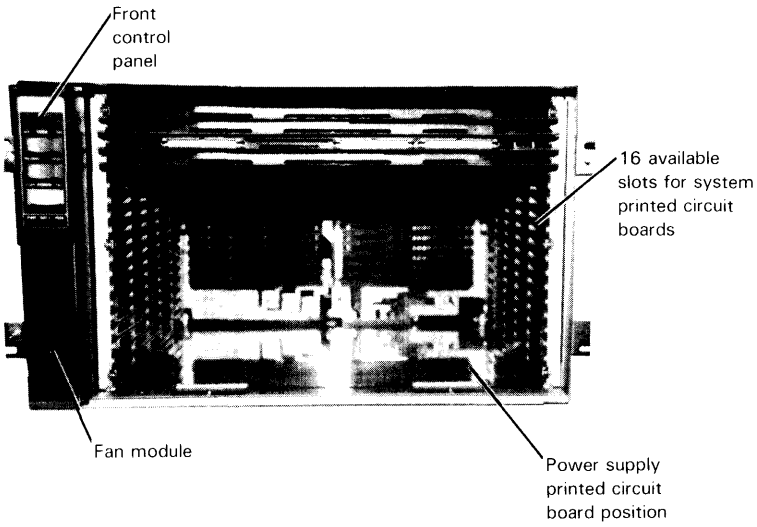
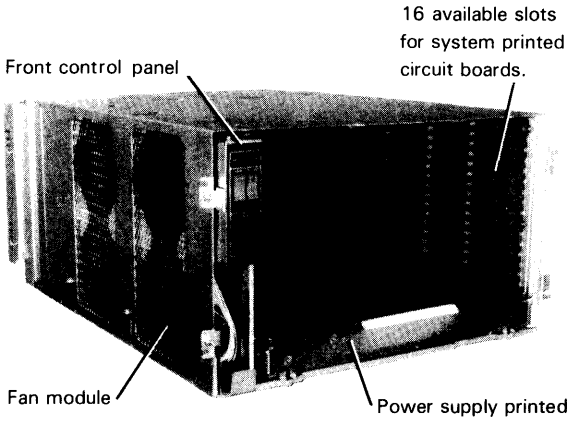


Figure 3.1 Compliant 16-slot chassis



Noncompliant 16-slot chassis

DG-26335

Figure 3.2 Noncompliant 16-slot chassis

Chapter 3

16-Slot Chassis

This chapter describes both the compliant and noncompliant ECLIPSE S/120 16-slot chassis (Figures 3.1 and 3.2) and architecture. The chapter also details chassis-to-device cabling and the system's expansion potential.

Chassis Architecture

The 19-inch, rack-mountable, ECLIPSE S/120 sixteen-slot chassis shown in Figures 3.1 through 3.3, accommodates the S/120 system processor unit and up to 14 input/output interface printed circuit boards. The chassis, together with its power supply, fan module, front console, and front cover, fits all standard Data General rack cabinets as well as 19-inch National Electrical Manufacturers Association (NEMA) standard industrial enclosures.

The power supply consists of a VNR (voltage non-regulated) unit and a slide-in printed circuit board. The VNR unit is vertically hinge-mounted to the rear of the chassis; the printed circuit board plugs into the chassis backpanel. The fan module, located on the left side of the chassis, slides into the chassis from the front of the unit. The fans within the module draw air from outside the cabinet and force it through the chassis. The front console is mounted on the upper-left front of the fan module.

The chassis contains its own backpanel printed circuit board with connectors for up to sixteen 15″ by 15″ system printed circuit boards. These system boards and a power supply board are inserted from the front of the chassis after the front panel has been removed. Figure 3.4 illustrates slot assignments for the system boards. Slot 1 is reserved for the SPU; Slot 2 is reserved for future use.

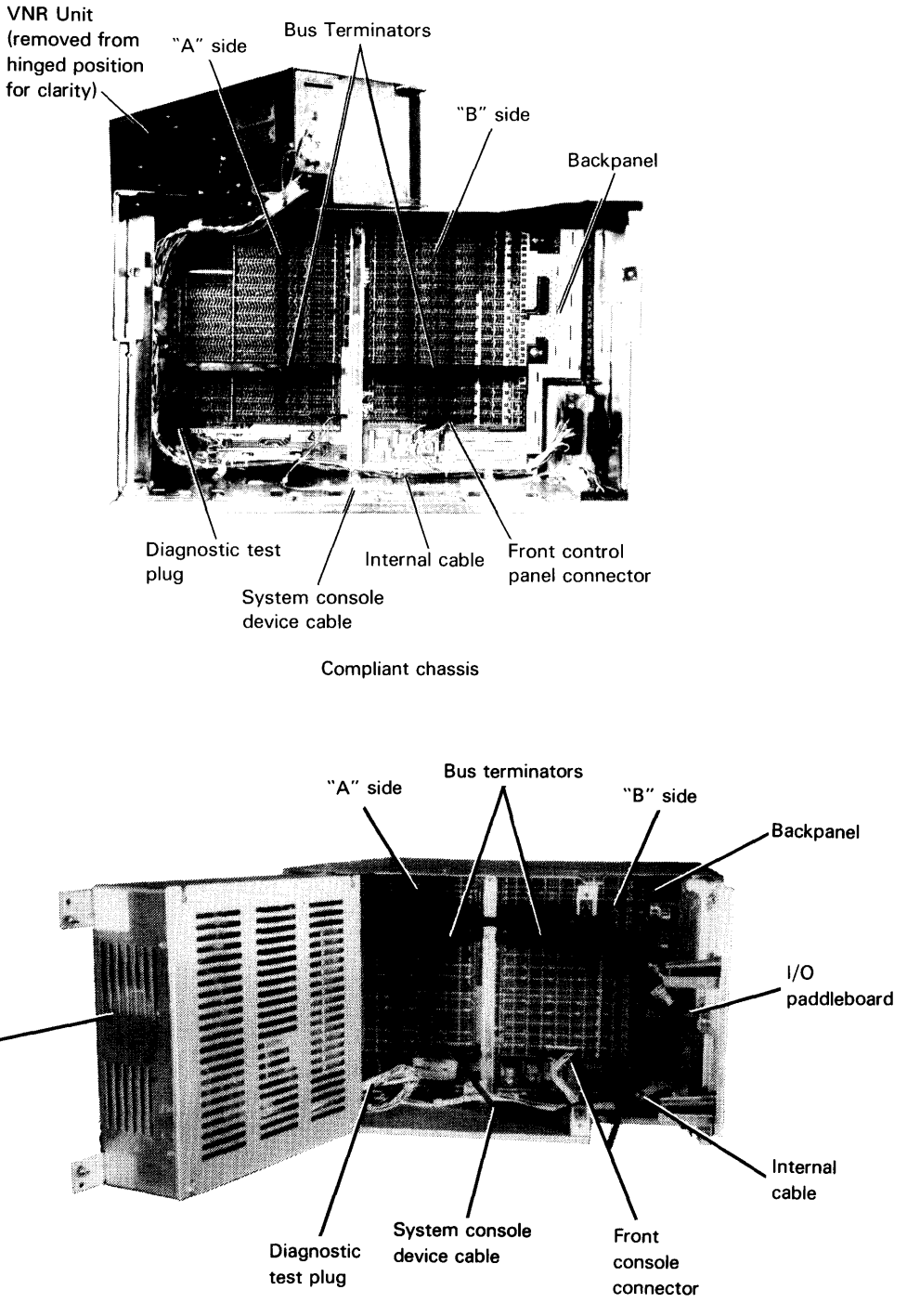
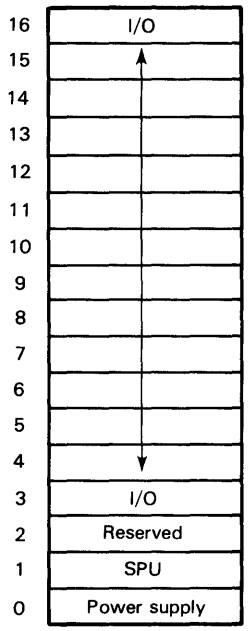


Figure 3.3 Compliant and noncompliant 16-slot chassis: major components (rear view)

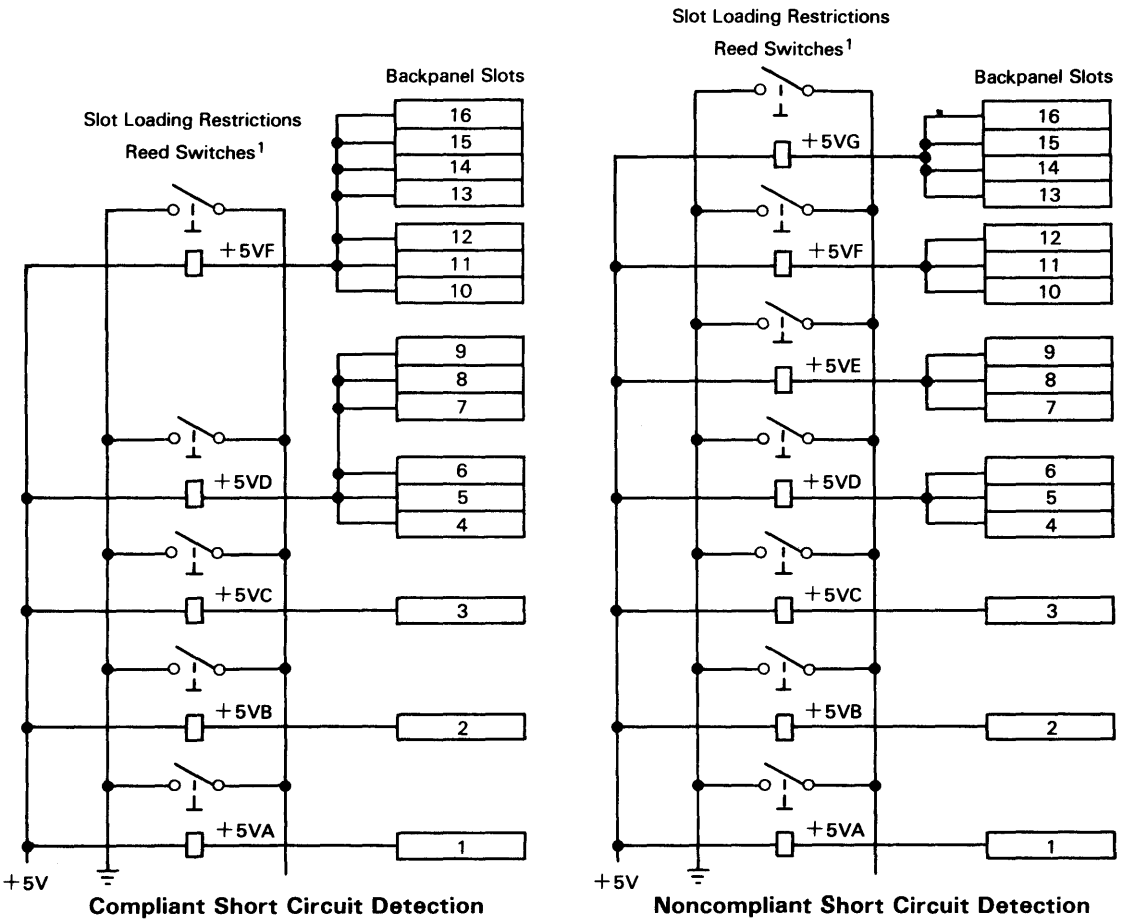
The +5V power on the backpanel is distributed to specific slots, as detailed in Figure 3.5, via backpanel mounted reed switches. These reed switches trip at 22 amps and provide a short circuit detection signal to the power supply fault detection circuits. Connector assignments for the backpanel are given in Figure 3.6.

An etched printed circuit input/output bus connects the ECLIPSE S/120 SPU to all system interfaces except the system console device and extends up the backpanel pins between slots 1 and 16. The input/output bus contains 16 bidirectional data lines and 32 unidirectional control lines. In addition, for system expansion, the input/output bus can be extended to an expansion chassis.



DG-09067

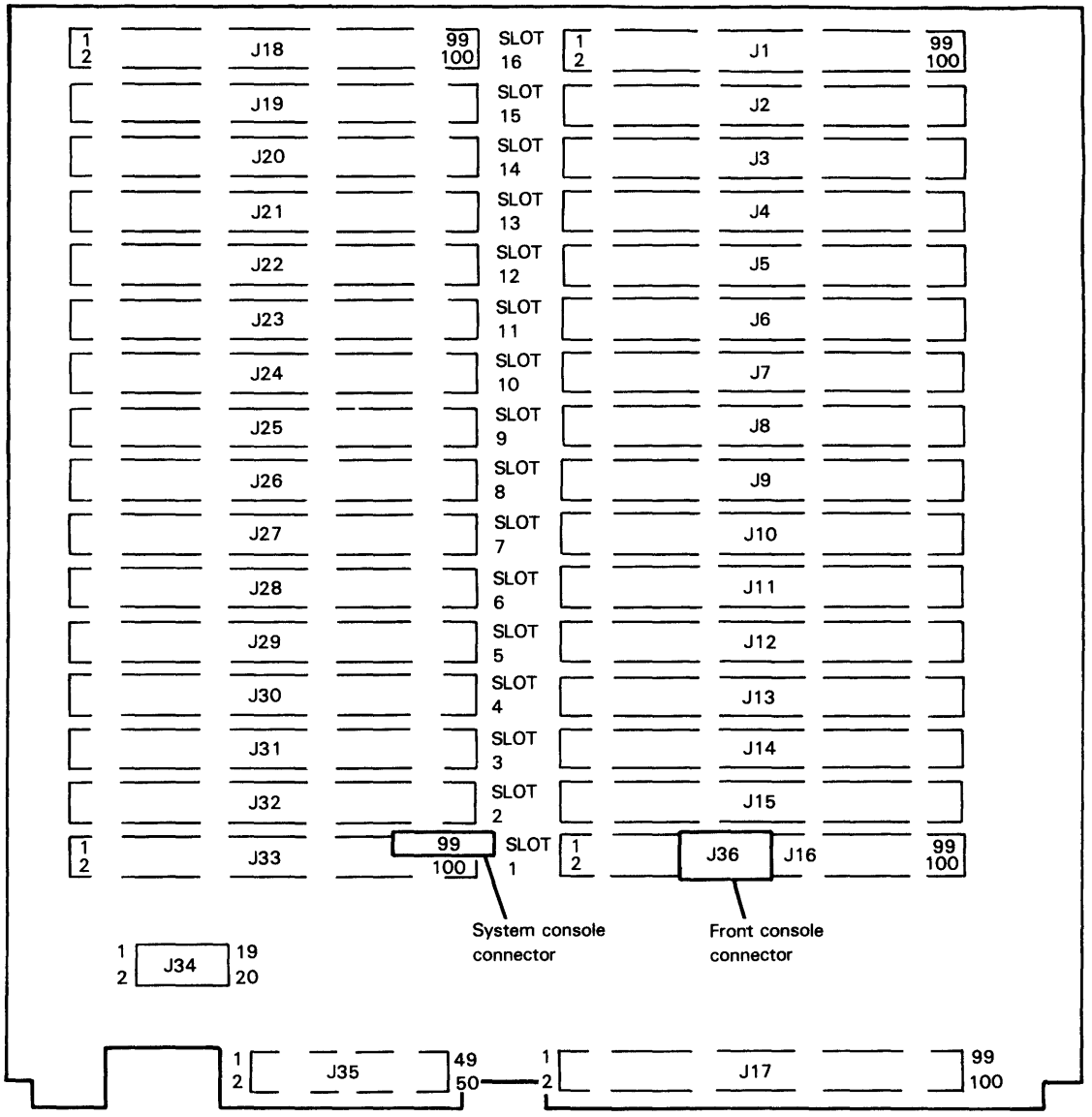
Figure 3.4 16-slot chassis: slot assignments



[1]REED SWITCH TRIPS AT 22 AMPS

DG-26336

Figure 3.5 Compliant and noncompliant +5V power distribution



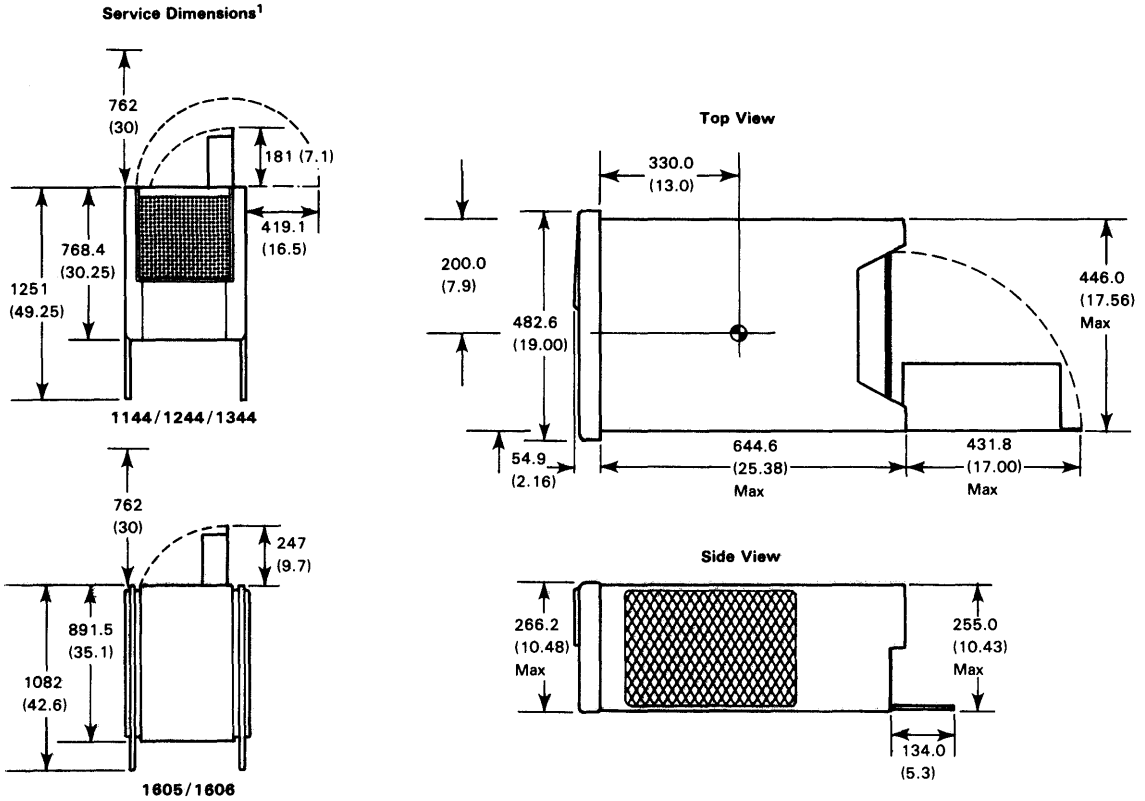
[1]*For complete pin assignments refer to DGC drawing no. 001-003182 for compliant systems, and DGC drawing no. 001-001563 for noncompliant systems.*

DG-09100

Figure 3.6 Backpanel connector assignments[1]

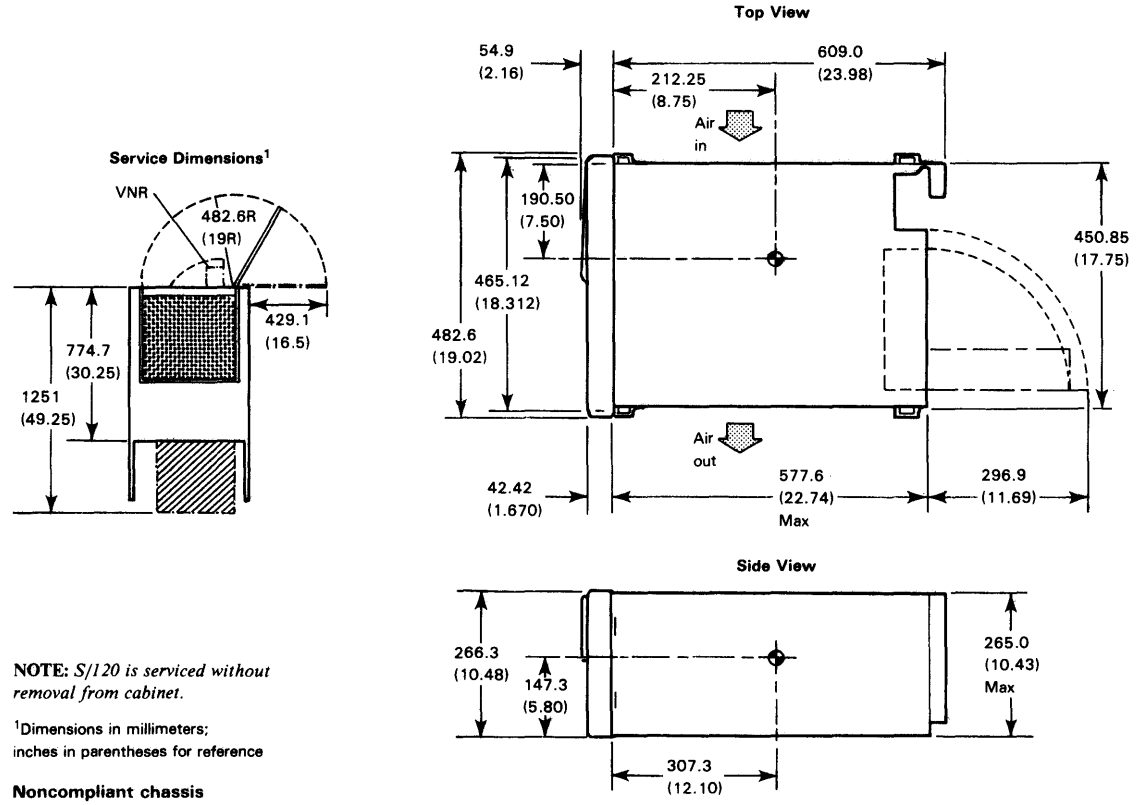
Two bus terminators plug onto the backpanel pins. One terminator is plugged onto one of the backpanel slot connector pins on each of the "A" and "B" sides. The terminators should be placed near the vertical center of the backpanel, at a slot that is not used by a device cable connector.

Figure 3.7 shows the external dimensions of the chassis.



NOTE: *S/120 is serviced without removal from cabinet.*

Compliant chassis



NOTE: *S/120 is serviced without removal from cabinet.*

[1]Dimensions in millimeters; inches in parentheses for reference

Noncompliant chassis

DG-26338

Figure 3.7 Compliant and noncompliant 16-slot chassis: external dimensions

Backpanel Jumpering

Backpanel jumpering is required to facilitate the interrupt and data-channel priority schemes. Priority chains must begin at the lowest numbered slot in the chassis that contains an input/output controller board. The priority chains are originated by connecting a jumper wire between pins A95 (**INTPOUT**) and A100 (**GND**) and another jumper wire between A93 (**DCHPOUT**) and A99 (**GND**) of the slot immediately below the lowest numbered chassis slot that contains an I/O board. (For compliant systems, refer to "Installation Data Sheets," DGC No. 010-000358; for noncompliant systems, refer to "Installation Data Sheets," DGC No. 010-000308.)

Backpanel jumpering is also required when any backpanel slot between input/output controller boards are unused. In this case, the interrupt priority (**INTP**) and data channel priority (**DCHP**) chain must be jumpered across the unused slot(s). The priority chains are jumpered across an unused slot by connecting a jumper wire between pins A96 (**INTPIN**) and A95 (**INTPOUT**) and another jumper wire between A94 (**DCHPIN**) and A93 (**DCHPOUT**) of the slot. (For compliant systems, refer to "Installation Data Sheets," DGC No. 010-000358; for noncompliant systems, refer to "Installation Data Sheets," DGC No. 010-000308.)

Front Console

The front console shown in Figure 3.8 is located on the front upper-left corner of the chassis. It consists of three switches (one of which is double action) and three indicators, defined below.

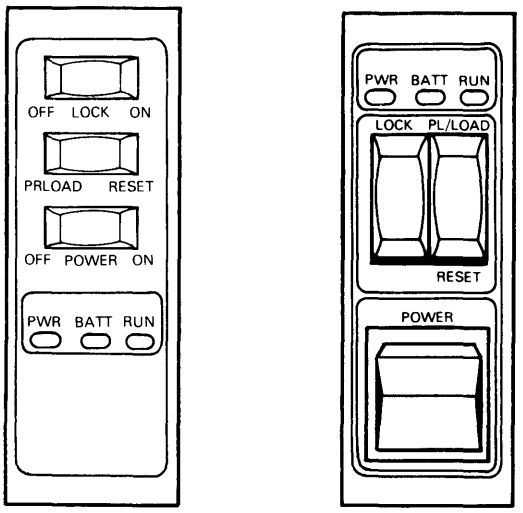


Figure 3.8 Compliant and noncompliant 16-slot chassis: front console

Switches

The three switches on the front console are

POWER
PL/LOAD-RESET
LOCK

The **POWER** switch applies and removes power from the system. When **PL/LOAD - RESET** is pressed to the PL/LOAD position and the S/120 is in the virtual console mode, the SPU performs a program load from the device whose device code is jumpered on the SPU board. (For jumpering information, refer to Installation Data Sheets or "Installation and Jumpering" in Chapter 1 of this manual.)

When this switch is pressed to the RESET position, the SPU performs a system reset and enters the virtual console mode, which is described in Chapter 7 of this manual.

Placing the **LOCK** switch in the LOCK position disables the **PR/LOAD - RESET** switch described above. In addition, placing the **LOCK** switch in the LOCK position:

- Permits automatic program load to be performed when power is applied to the S/120.
- Permits automatic restart to be performed when power is restored following a power failure when battery backup is present and the batteries have not been exhausted during the power disruption.
- Inhibits access to virtual console when the system console **BREAK** key is pressed.

Indicators

The three indicators on the front console are

PWR
BATT
RUN

When illuminated, **PWR** indicates that the system is powered up and the power supply voltages are within operating specifications.

When illuminated, **BATT** indicates that the system memory is on battery-backup power. This is usually the result of a power failure.

When illuminated, **RUN** indicates the system is in the run mode. This indicator is normally not lighted when the system is in virtual console mode.

Additional Indicators

Three additional indicators are located on the front of the power supply board as shown in Figure 3.9. These indicators provide the following status information:

Over-current When illuminated, indicates the power supply detected an over-current condition. To clear the fault indication, power must be removed and reapplied to the power supply. If the fault causing the over-current condition is still present when power is reapplied, the indicator lights again.

Over-voltage When illuminated, indicates the power supply detected an over-voltage condition. To clear the fault indication, power must be removed and reapplied to the power supply. If the fault causing the over-voltage condition is still present when power is reapplied, the indicator lights again.

Memory-disaster When illuminated, indicates the SPU printed circuit board detected dc power failure on **-5VMEM**. To clear the fault indication, power must be removed and reapplied to the power supply. If the fault causing the **-5VMEM** power failure condition is still present when power is reapplied, the indicator lights again.

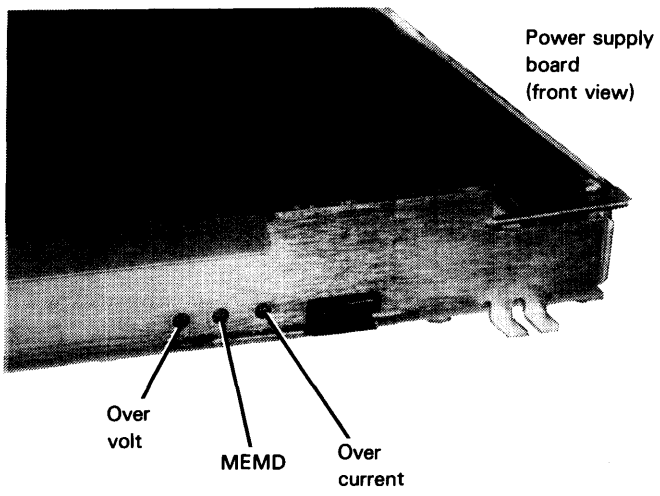


Figure 3.9 16-slot power supply board: indicator lights

Cabling

The line cord, system console cable, and I/O cables connect the chassis to ac line voltage and input/output devices. This cabling is shown in Figures 3.3, 3.10, and 3.11.

Line Cord

In both compliant and noncompliant systems, the line cord plugs into a connector located at the rear of the VNR unit as shown in Figure 3.10. Two line cords are available: one for 100/120 Vac and another for 220/240 Vac.

In compliant systems, one of two external loopback plugs connects ac voltage to the chassis fans, depending on the ac source voltage. In the same manner, one of two internal loopback plugs inside the VNR unit connects the ac source voltage to the VNR printed circuit board. In noncompliant systems, the line cord is provided with internally wired connections for the line voltage available to the chassis.

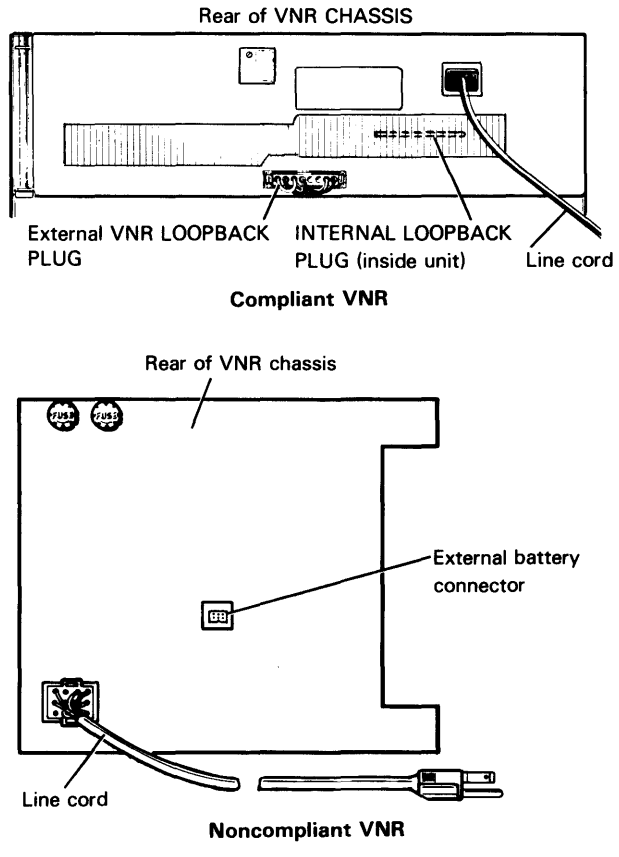


Figure 3.10 Line cord connection

System Console Cable

This cable connects the ECLIPSE S/120 system processor unit directly to a system console device. One end of the cable contains a 12-pin push-on connector that pushes onto designated backpanel pins of Slot 1, connector A, as shown in Figure 3.11. The backpanel connector assignment of the system console connector is shown in Figure 3.6.

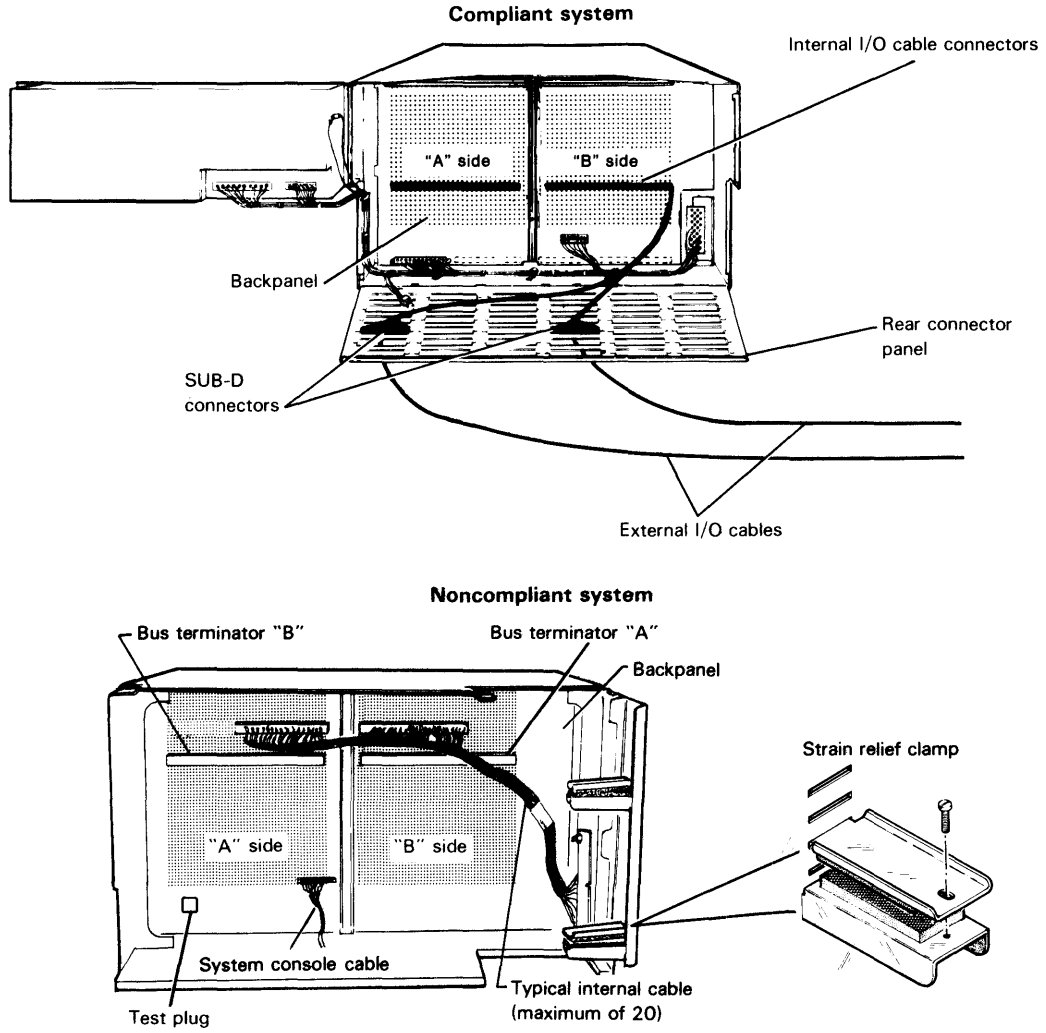


Figure 3.11 Compliant and noncompliant 16-slot connectors

I/O Cables

Connections, in both compliant and noncompliant systems, between an input/output interface and the device it controls are made in two parts: first through an internal chassis cable, then through an external (or device) cable.

In a compliant system, standard length internal cables bring the device signals from the interface backpanel connector pins to a connector panel located at the rear of the chassis. These internal cables are strain-relieved to the side of the chassis and have connectors that push onto backpanel pins at one end and sub-D connectors at the other end which plug into the rear connector panel of the chassis. This allows the chassis connector panel to hinge-open for servicing. External device cables from all peripherals plug directly into the sub-D connectors at the chassis connector panel.

A typical internal cable connection with the rear connector panel open is shown in Figure 3.11.

In a noncompliant system, the internal chassis cable brings the device signals from the interface backpanel connection pins to a convenient connection panel; the external cable takes the device signals from the S/120 chassis to the device. A push-on connector at one end of the internal cable plugs onto the interface backpanel connection pins, and an edge connector (paddleboard) at the other end of the internal cable mates with the external device cable. The device cables enter at the rear of the unit through a cutout in the bottom of the chassis and connect to the internal cable edge connector (paddleboard). A strain relief clamp holds the cables securely in position as they pass from the cutout.

A typical internal cable, the paddleboard location, and strain relief clamp are shown in Figure 3.11.

Diagnostic Test Plug

The diagnostic test plug is reserved for future use and must be installed in the RUN position.

CAUTION: *Use of this plug in other than the RUN position may seriously degrade the system.*

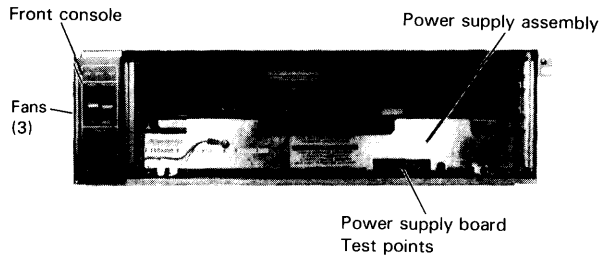
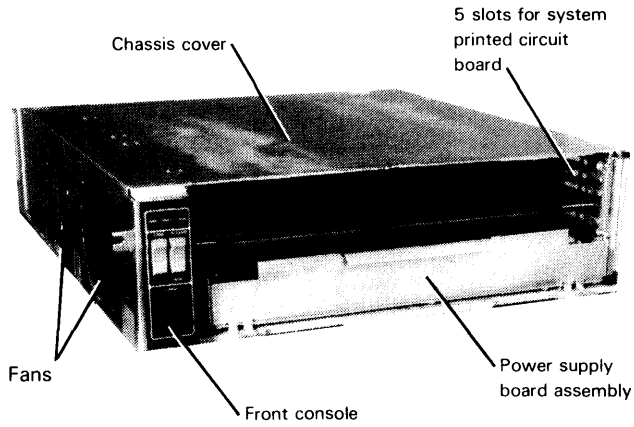


Figure 4.1 Compliant 5-slot chassis



Noncompliant 5 slot chassis

Figure 4.2 Noncompliant 5-slot chassis

Chapter 4

5-Slot Chassis

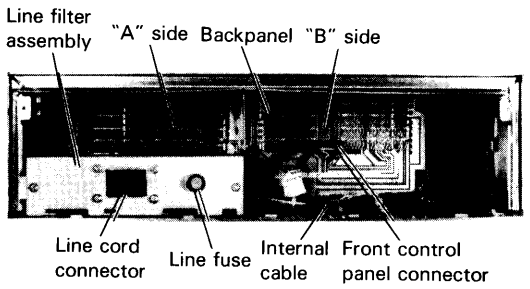
This chapter describes the ECLIPSE S/120 five-slot chassis (Figures 4.1 and 4.2) and its architecture. The chapter also details chassis-to-device cabling and the system's expansion potential.

Chassis Architecture

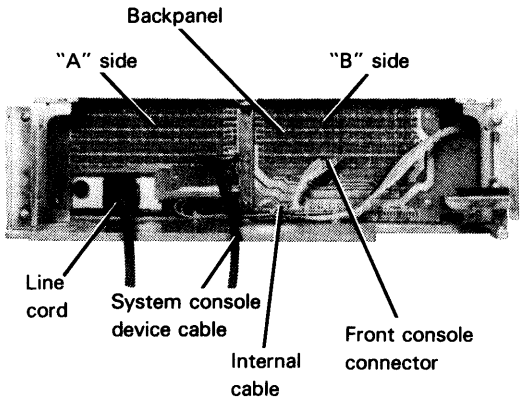
The 19-inch, rack-mountable, ECLIPSE S/120 five-slot chassis shown in Figures 4.1 through 4.3 accommodates the S/120 system processor unit and up to four input/output interface printed circuit boards. The chassis, together with its power supply, front console, and front cover fits all standard Data General rack cabinets as well as 19-inch National Electrical Manufacturers Association (NEMA) standard industrial enclosures.

The power supply is contained on one slide-in printed circuit board that plugs into the chassis backpanel. Two fans, located on the inside left of the chassis cover, draw air from outside the cabinet and force it through the chassis. A front console is mounted on the left side of the chassis cover.

The chassis contains its own backpanel printed circuit board with connectors that accept up to five 15″ by 15″ system printed circuit boards. These system boards and a power supply board are inserted from the front of the chassis after removing its front cover. The backpanel connectors are identified as "A" side and "B" side, as shown in Figure 4.3. Figure 4.4 illustrates slot assignments for the system boards; slot 1 is reserved for the SPU. Connector assignments for the backpanel are listed in Figure 4.6.



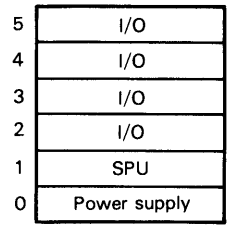
Compliant chassis



Noncompliant chassis

DG-26344

Figure 4.3 Compliant and noncompliant 5-slot chassis: major components (rear view)



DG-09070

Figure 4.4 5-slot chassis: slot assignments

A printed circuit input/output bus connects the ECLIPSE S/120 SPU to all system interfaces, except the system console device, and extends up the backpanel pins between slots 1 and 5. The input/output bus contains 16 bidirectional data lines and 32 unidirectional control lines. In addition, for system expansion the input/output bus can be extended to an expansion chassis. Termination for the input/output bus is provided by a register fence, which is soldered into the backpanel above slot 5.

Figure 4.5 shows the external dimensions of the chassis.

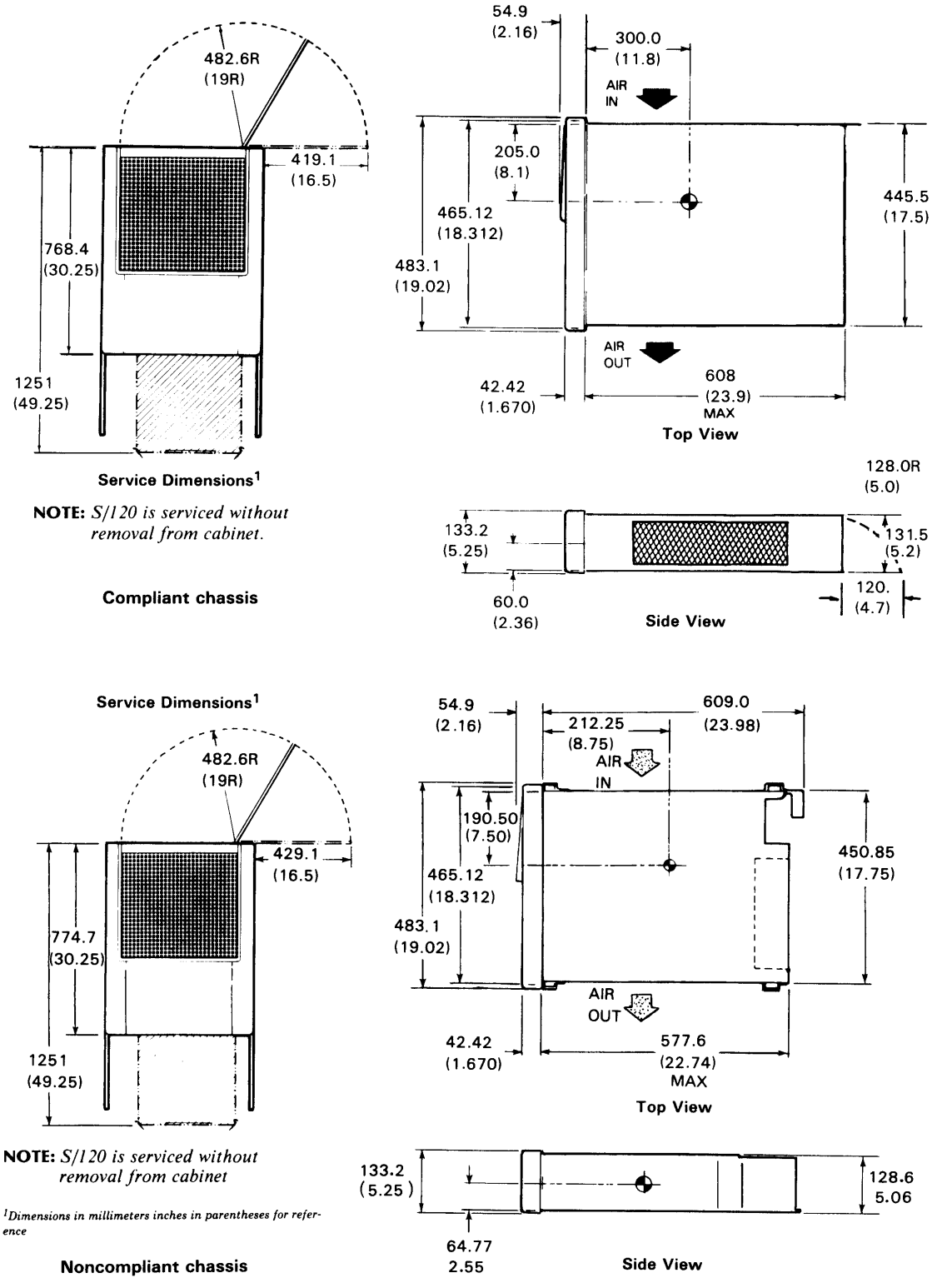


Figure 4.5 Compliant and noncompliant 5-slot chassis: external dimensions

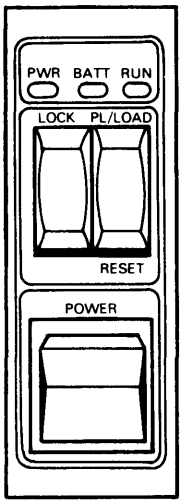
Backpanel Jumpering

Backpanel jumpering is required when any backpanel slot is unused between the SPU board and the highest numbered slot in the chassis containing an input/output controller board. When any slot is unused, the interrupt priority ($\overline{\text{INTP}}$) and data channel priority ($\overline{\text{DCHP}}$) chain must be jumpered across the unused slot(s). The priority chains are jumpered across an open slot by connecting a jumper wire between pins A96 ($\overline{\text{INTPIN}}$) and A95 ($\overline{\text{INTPOUT}}$) and another jumper wire between A94 ($\overline{\text{DCHPIN}}$) and A93 ($\overline{\text{DCHPOUT}}$) of the open slot. (For compliant systems, refer to "Installation Data Sheets" 010-000360; for noncompliant systems, refer to "Installation Data Sheets," DGC No. 010-000298.)

An additional backpanel jumper is required to connect **+5VA** to **+5VMEM** when the battery-backup option is not present. This connection is made by connecting a jumper wire between pins B98 (**+5VA**) and B94 (**+5VMEM**) of slot 1. (For compliant systems, refer to "Installation Data Sheets" 010-000360; for noncompliant systems, refer to "Installation Data Sheets," DGC No. 010-000298.)

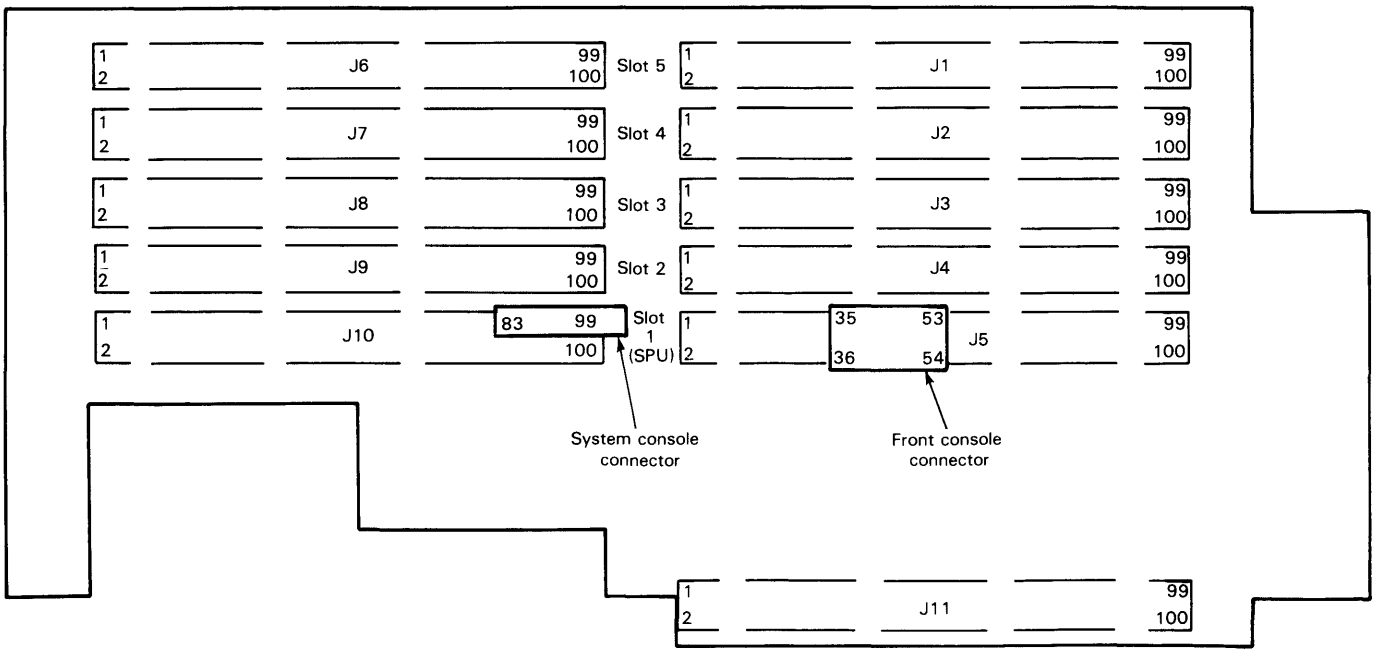
Front Console

The front console, shown in Figure 4.7, is located on the front upper-left corner of the chassis. It consists of three switches (one of which is double action) and three indicators, defined on the following page.



DG-06062

Figure 4.7 Compliant and noncompliant 5-slot chassis: front console



NOTE: For complete pin assignments refer to DGC drawing no. 001-003226 for Compliant systems, and DGC drawing 001-001619 for Noncompliant systems.

DG-09098

Figure 4.6 Connector assignments

Switches

The switches on the front console are

POWER
PL/LOAD-RESET
LOCK

The **POWER** switch applies and removes power from the system.

> **NOTE:** *If* **POWER** *switch of the noncompliant 5-slot chassis is placed in the OFF position and* **LOCK** *(described below) is in the LOCK position, the system memory will go into battery-backup mode if that option is present.*

When the **PL/LOAD - RESET** switch is pressed to the PL/LOAD position and the S/120 is in the virtual console mode, the SPU performs a program load from the device whose device code is jumpered on the SPU board. (For jumpering information, refer to Installation Data Sheets or "Installation and Jumpering" in Chapter 1 of this document.)

When this switch is pressed to the RESET position, the SPU performs a system reset and enters the virtual console mode, described in Chapter 7 of this document.

> **NOTE:** *If the* **LOCK** *switch (described below) is in the LOCK position, the* **PL/LOAD - RESET** *switch has no effect.*

In the LOCK position, the **LOCK** switch disables the **PR/LOAD - RESET** switch described above. In addition, placing the **LOCK** switch in the LOCK position:

- Permits automatic program load to be performed when power is applied to the S/120.
- Permits automatic restart to be performed when power is restored following a power failure when battery backup is present and the batteries have not been exhausted during the power disruption.
- Inhibits access to virtual console when the system console **BREAK** key is pressed.
- Causes the S/120 of a noncompliant 5-slot chassis to go into battery backup mode, if that option is present, when the **POWER** switch is placed in the off position. If battery backup is not present, the S/120 will be powered down.

Indicators

The three indicators on the front console are

PWR
BATT
RUN

When illuminated, **PWR** indicates that the system is powered up and the power supply voltages are within operating specifications.

When illuminated, **BATT** indicates the system memory is on battery-backup power. This usually results from a power failure.

When illuminated, **RUN** indicates the system is in the RUN mode. This indicator is normally not lighted when the system is in virtual console mode.

Cabling

The line cord, system console cable, and I/O cables connect the chassis and its system components to ac line voltage and input/output devices. This cabling is shown in Figure 4.3 and 4.8.

Line Cord

In both compliant and noncompliant systems, the line cord plugs into a connector located at the rear of the chassis as shown in Figure 3.10.

In compliant systems, two line cords are available: one for 100/120 Vac and another for 220/240 Vac. One of two ac line voltage selection plugs connects the ac source voltage to the chassis fans and power supply PC board. This selection plug connects onto the backpanel pins located at the rear of the chassis.

In noncompliant systems, four line cords are available; 100 Vac, 120 Vac, 220 Vac, and 240 Vac. The connector of the line cord is internally wired to provide the line voltage available to the chassis.

System Console Cable

This cable connects the ECLIPSE S/120 system processor unit directly to a system console device. One end of the cable contains a 12-pin push-on connector that pushes onto designated backpanel pins of slot 1, connector A, as shown in Figure 4.8. The backpanel connector used by the system console connector is shown in Figure 4.6.

I/O Cables

Connections, in both compliant and noncompliant systems, between an input/output interface and the device it controls are made in two parts: first through an internal chassis cable, then through an external (or device) cable.

In a compliant system, standard length internal cables bring the device signals from the interface backpanel connector pins to a connector panel located at the rear of the chassis. These internal cables are strain-relieved to the side of the chassis and have connectors that push onto backpanel pins at one end and sub-D connectors at the other end which plug into the rear connector panel of the chassis. This allows the chassis connector panel to hinge-open for servicing. External device cables from all peripherals plug directly into the sub-D connectors at the chassis connector panel.

A typical internal cable connection with the rear connector panel open is shown in Figure 4.8.

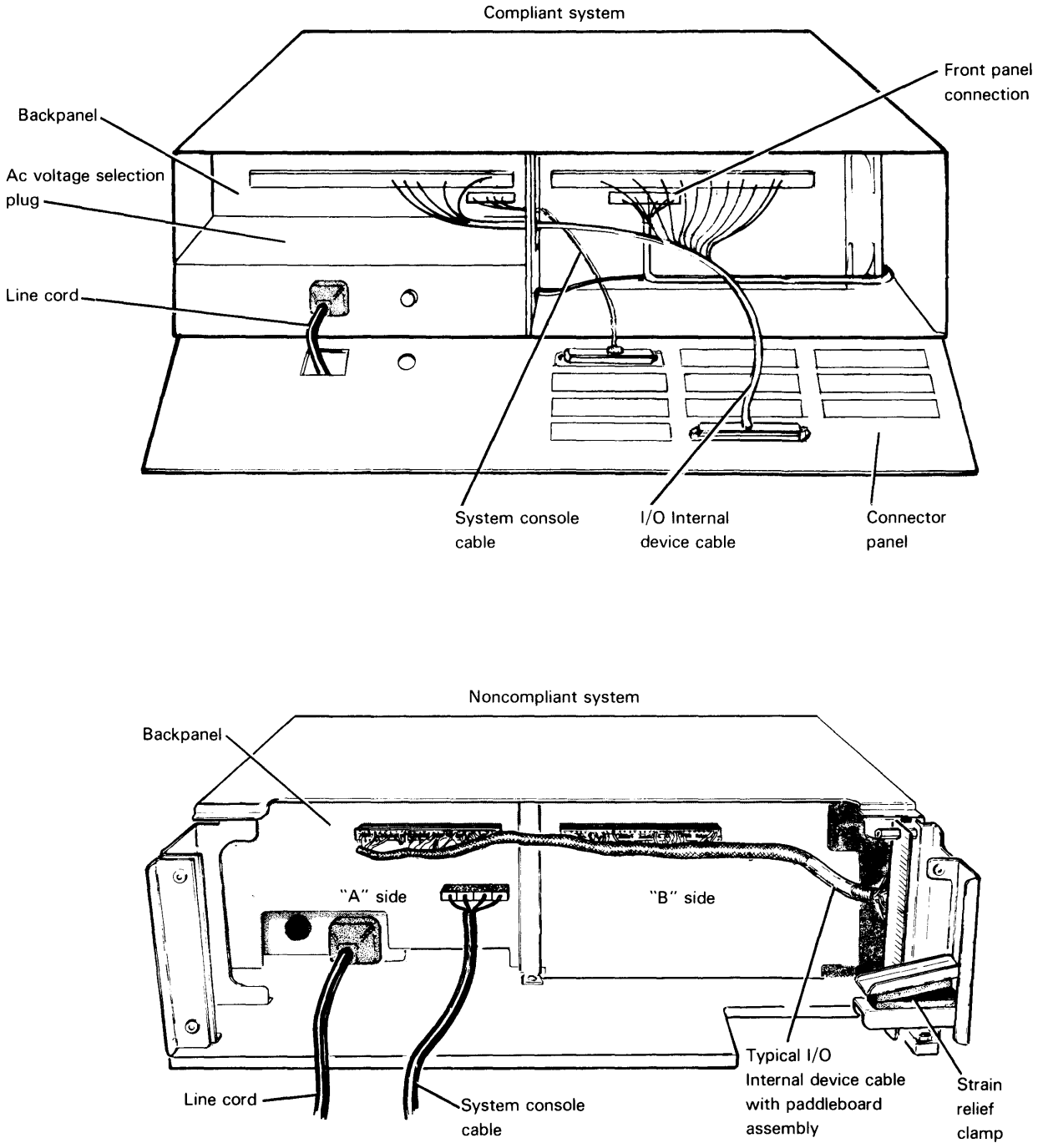
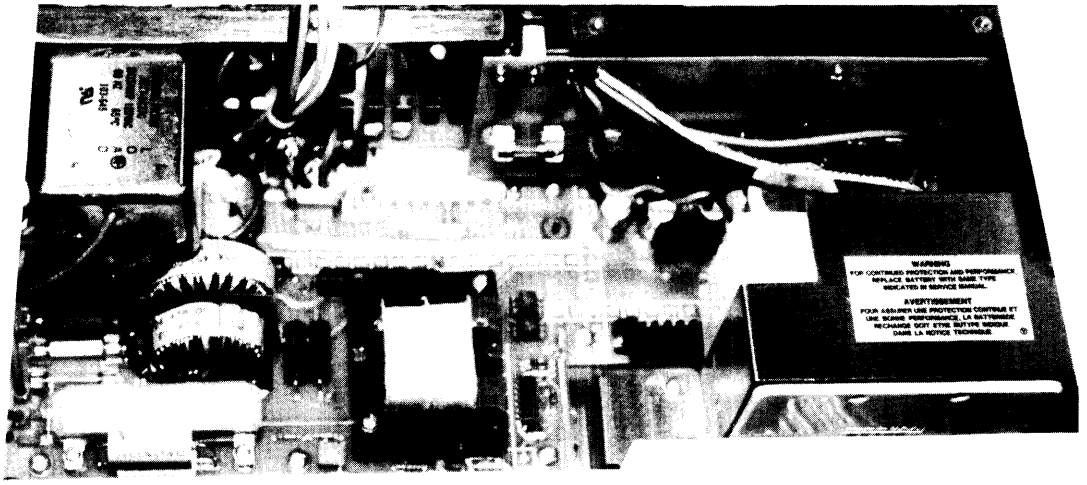


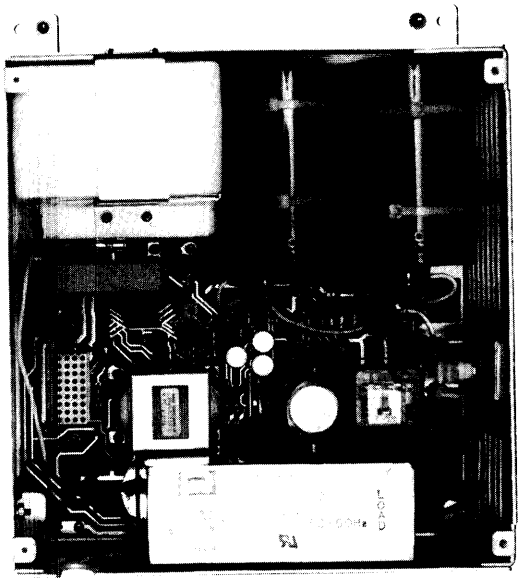
Figure 4.8 Compliant and noncompliant 5-slot connectors

In a noncompliant system, the internal chassis cable brings the device signals from the interface backpanel connection pins to a convenient connection panel; and the external cable takes the device signals from the S/120 chassis to the device. A push on connector at one end of the internal cable plugs onto the interface backpanel connection pins, and an edge connector (paddleboard) at the other end mates with the external device cable. The device cables enter at the rear of the unit through a cutout in the bottom of the chassis and connect to the internal cable edge connector (paddleboard). A strain relief clamp holds the cables securely in position as they pass from the cutout.

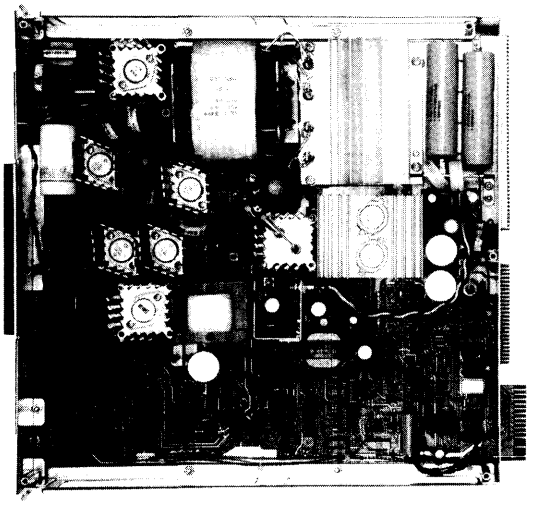
A typical internal cable, the paddleboard location, and strain relief clamp are shown in Figure 4.8.



Compliant VNR unit



Noncompliant VNR unit



Typical 16-slot power supply PCB

PH-0791
PH-0574
PH-0575

Figure 5.1 Compliant and noncompliant 16-slot VNR unit and power supply PCB

Chapter 5

16-Slot Power Supply

The ECLIPSE S/120 16-slot power supply converts a 120 or 220/240 ac voltage source to the five regulated dc voltages required by the ECLIPSE S/120 computers. It also generates the signal necessary to implement the system's real-time clock at line frequency intervals. A battery backup feature generates the regulated dc memory voltages from a +12 volt battery during an ac power failure.

In both compliant and noncompliant systems, the power supply consists of a VNR (voltage non-regulated) unit and a printed circuit board. (See Figure 5.1.) The VNR unit supplies both nonregulated dc voltage to the power supply printed circuit board and ac voltage to the fan module via the internal cable. The VNR unit in compliant systems differs from the VNR unit in noncompliant systems. A discussion of both VNR units is in the "Theory of Operation" section that follows.

The VNR unit connects to the power supply printed circuit board through the internal chassis cable. It also connects to the front control panel power switch and to the chassis fans through the internal chassis cable and the internal fan module cable.

A line cord connects the VNR unit to an ac power source. Compliant line cords have a 3-pin plug and the correct operating ac voltage is selected by VNR unit loopback plugs. Noncompliant line cords have a 12-pin plug at the VNR unit end to select the correct operating ac voltage for the VNR unit. The power supply printed circuit board supplies the regulated dc voltages and a low frequency clock signal to the system printed circuit boards via the backpanel.

Functional Overview

The 16-slot power supply uses a forward off-line switching regulator to produce the high-power outputs required by the ECLIPSE S/120 computers. The major components of this off-line switching regulator are a rectifier and filter, a buck regulator, and a dc to dc converter.

Figure 5.2 shows the interconnection of these components.

The *rectifier* and *filter* convert power from the ac line to a high voltage dc source. The voltage varies with the line, but is typically 300 volts.

The *buck regulator* takes power from the filter and provides a constant voltage for the dc to dc converter. It

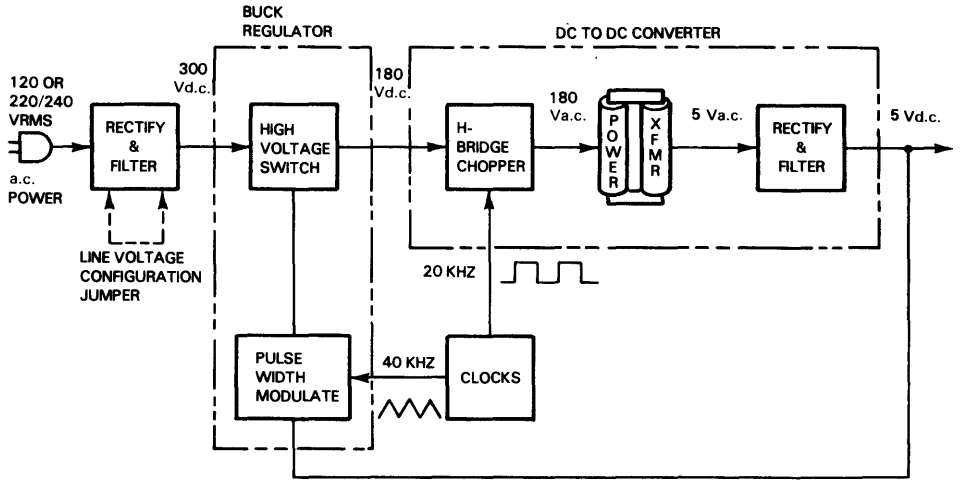


Figure 5.2 Off-line switching regulator

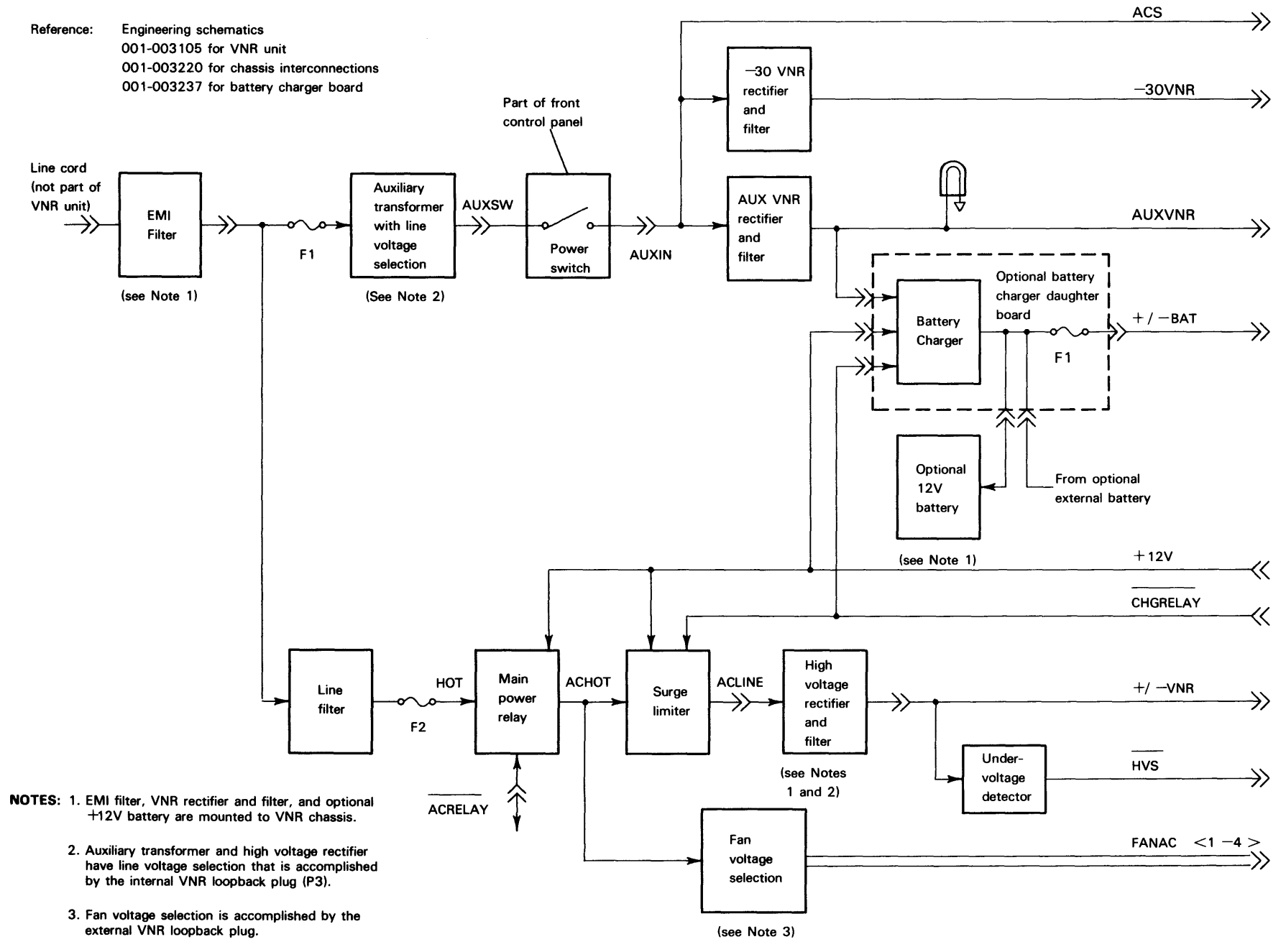


Figure 5.3 Compliant 16-slot VNR unit

monitors the output voltage, delivering more current as the load increases to keep the output regulated. The buck regulator includes a high voltage switch and a pulse width modulator. The switch alternately opens and closes the path between the filter and the dc to dc converter. The pulse width modulator opens and closes the switch at a constant rate of 40 Khz but varies the *duty cycle*, or the ratio between on and off times, to achieve regulation.

The *dc to dc converter* takes power from the buck regulator and produces the low voltage output. It includes an H-bridge chopper, a power transformer, and a rectifier and filter. The chopper converts regulated dc power to a high voltage square wave for the power transformer. It does this by alternately closing a current path in one direction and then the opposite direction between the buck regulator and the transformer's primary winding. The transformer converts the high voltage/low current input to a low voltage/high current output, which the rectifier and filter convert back to dc.

Theory of Operation

This section examines the functional blocks of the ECLIPSE S/120 power supply. These blocks are distributed between the VNR unit and the power supply circuit board. To aid in identifying a failing component, special attention will be paid to the signals that flow in and out of these modules.

Compliant VNR Unit

The compliant VNR (voltage nonregulated) unit:

- provides low and high nonregulated dc power for the power supply printed circuit board,
- distributes ac line power to the fans,
- provides a line frequency signal to the power supply printed circuit board,
- provides battery backup power for the system memory when that option is present.

Figure 5.3 shows the components of the VNR unit. In the text to follow, a description of a power on/off sequence illustrates their operation.

Power On sequence

When the line cord to the VNR unit is plugged in, the power goes through the EMI filter and then branches, passing through fuse F1 to the auxiliary transformer, and through the line filter and fuse F2 to the main power relay.

The auxiliary transformer provides an ac signal of approximately 30 volts (**AUXSW**) to the power switch on the front control panel. When this switch is turned on, the signal **AUXIN** returns to the VNR unit. This voltage is rectified and filtered to provide approximately 30 volts of nonregulated positive and negative dc start-up power to the power supply board (**AUXVNR** and **−30VNR**).

In addition, **AUXVNR** lights an internal VNR unit indicator showing that the unit's low voltage circuits are operating.

AUXIN connects from the VNR unit to the power supply board as signal **ACS**. The power supply board uses this signal to generate a line frequency clock signal for the central processor board.

The control circuits of the power supply board regulate the **AUXVNR** signal to 12 volts for the board's internal use and apply the 12 volts to the VNR unit as a signal to energize the main power relay (**+12V**). (Signal **$\overline{\text{ACRELAY}}$** is grounded as long as the internal chassis cable is plugged into the VNR unit's control connection (J11).

When the VNR switch circuit energizes, the ac power source goes to the chassis fans through the fan voltage selection plug, and to a surge limiter circuit (**ACHOT**). At start-up, the surge limiter circuit sends the ac power through a current limiting resistor to the VNR rectifier. As the VNR filter capacitors charge, high voltage dc power begins to rise, and is applied to an under-voltage detection circuit and to the power supply board (**+VNR, −VNR**). When the high voltage reaches a sufficent level, the undervoltage detector applies signal **$\overline{\text{HVS}}$** to the power supply board. In response to this signal, the power supply board returns **$\overline{\text{CHGRELAY}}$** to the VNR unit. This returned signal causes the surge limiter circuit to bypass the current-limiting resistor, applying full ac power to the VNR rectifier and filter.

Power Off Sequence

When the power switch is turned to the off position, the VNR unit removes signal **AUXVNR**, causing the power supply to remove control signal **+12V** from the VNR unit. This action deenergizes the main power relay, shutting off the high voltage to the power supply board.

Undervoltage Detection

During operation, if the nonregulated high voltage should fall below a minimum acceptable level, the VNR unit's undervoltage detector circuit removes **$\overline{\text{HVS}}$** to notify the power supply of this condition. The power supply board then notifies the central processor of the imminent power failure.

Battery Backup

When there is an optional battery backup, it provides 12 volt dc power from a battery mounted in the VNR chassis to special battery backup circuits on the power supply's printed circuit board. These circuits maintain the voltages necessary to retain system memory data when a power failure occurs. A battery charger is contained on a printed circuit board that plugs into a connector on the VNR

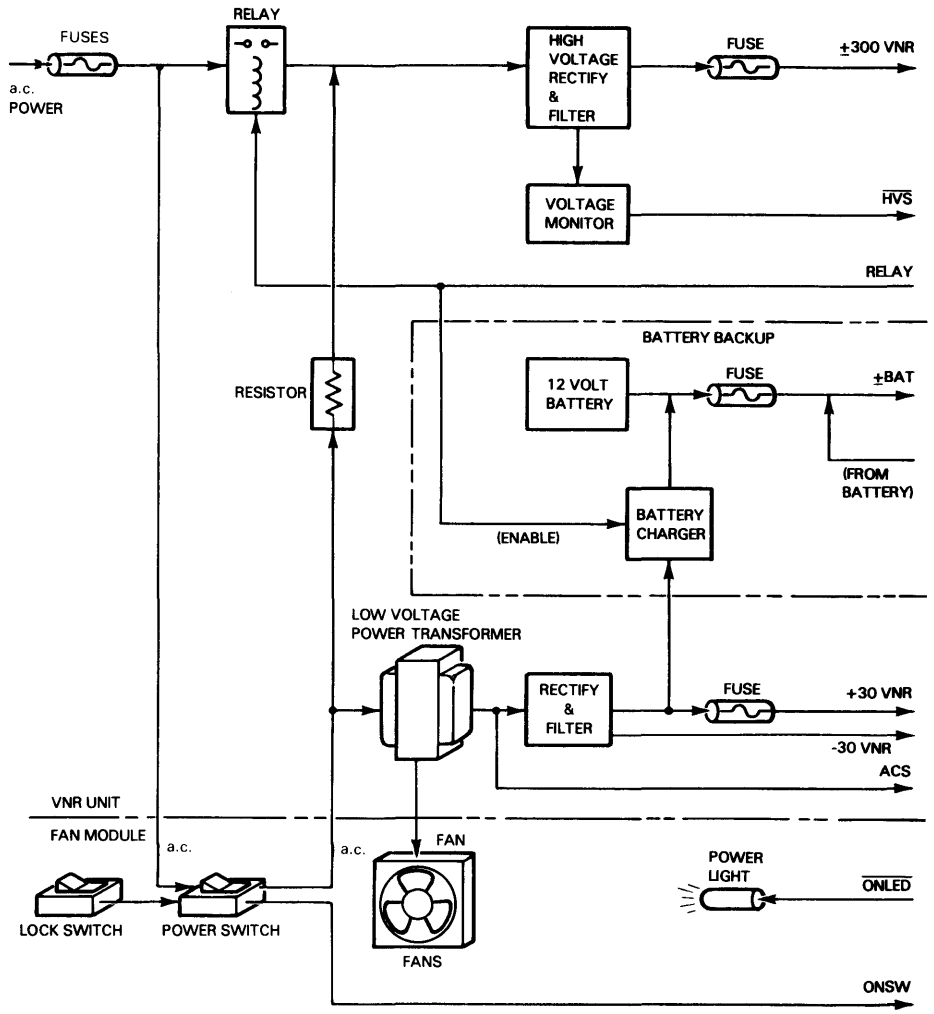


Figure 5.4 Noncompliant VNR unit and fan module

unit's printed circuit board. The battery power (**+BAT, −BAT**) reaches the power supply through the battery charger board, VNR unit board, and internal chassis cable. Fuse F1 protects the battery charger board. The battery charger is powered by **AUXVNR** and is enabled whenever the power supply returns **CHGRELAY**.

How long data can be retained during a power failure depends on the amount of system memory present in the chassis and on the amount of charge in the battery. Battery backup time can be increased by mounting an additional 12 volt battery outside the computer chassis and connecting it to the battery charger daughter board. however, the built-in charging circuit is designed to charge only the internal battery, and its circuit's performance will be reduced if it must charge an additional battery.

Noncompliant VNR Unit

The VNR (voltage-nonregulated) unit provides high and low voltage dc power for the power supply board. It can also generate battery backup power when that option is configured. Figure 5.4 shows the components of the VNR unit. In the text to follow a description of a power on/off sequence illustrates their operation.

Power On/Off Sequence

When the power switch is turned on, current flows from the ac power line, through the line fuse(s), and continues to the low voltage power transformer. To provide **+30VNR** and **−30VNR** for the control circuits on the power supply board, the transformer output is rectified and filtered. The transformer powers the fans and drives a low voltage ac signal (**ACS**) to the high state. This signal goes to the real-time clock circuits.

Current also flows through a very large resistor and into the high voltage rectifier and filter. The resistor limits the current flow while the filter capacitors charge, thus preventing them from blowing the line fuse(s). When the capacitors are sufficiently charged, the voltage monitor drives the high voltage sense (**HVS**) signal to the low state. In response, the supply board closes the relay, which short-circuits the resistor. As a result, full power is

available to the off-line switching regulator and the power-on sequence is completed.

Once the power switch is turned off, the fan module drives the on-switch (**ONSW**) signal to the low state. After a short delay, the supply board opens the relay, which disconnects the VNR unit from the ac power line.

The optional battery backup module generates 12 volt dc power for special battery backup circuits on the power supply board. These circuits maintain critical memory voltages when a power failure occurs, in order to save data stored in main memory. The battery can support main memory for up to 60 minutes. If more time is needed, a larger external battery — an automotive battery, for example—can be connected. The backup option includes a battery charger powered by the low voltage dc supply. The charger is enabled whenever the relay is turned on.

Power Supply Board

The power supply board provides several regulated dc voltages for the computer chassis. It also generates a low-frequency clock and status signals for the CPU. Figure 5.5 shows the components of the power supply board, and Table 5.1 lists the power supply specifications. Neither the figure nor the table describe the battery backup which will be covered later in this chapter, under "Battery Backup."

	Voltages		Current	
Output	Min	Max	Min	Max
+5V	+5.10V	+5.20V	7.5A	120A
+12V	+11.8V	+12.2V	0	12.5A*
+15V	+14.5V	+15.5V	0	1.5A*
−5V	−4.9V	−5.2V	0	3A
−11V	−11.0V	−12.5V	0	0.02A
+5 MEM	+5.10V	+5.20V	0	9.5A
+12 MEM	+11.8V	+12.2V	0	6A*
−5 MEM	−4.7V	−5.1V	0	0.3A

Table 5.1 16-slot chassis power supply specifications without battery backup

**The sum of the currents on +12V, +15, and +12 MEM must NOT exceed 12.5 Amps.*

Support Circuits

A description of some of the support circuits proceeds the discussion of the off-line switching regulators. The support circuits include

- *Auxiliary supply regulators* power all supply circuits except the main voltage regulators. They also provide an adjustable voltage reference, which eliminates the need for individual adjustments.
- *Clock logic* provides a ramp signal for the pulse width modulator, along with switch signals for the H-bridge chopper.
- *Sequencing logic* turns the off-line switching regulator on or off in several steps. The power-up sequence begins after the power switch is turned on (**ONSW**), the clocks start up (**RAMP**) and the high voltage bulk supply stabilizes (**HVS**). First the relay closes. Then the **+11 SW** signal starts the H-bridge chopper. Finally the power-on (**PON**) signal starts the buck regulator. If the power switch is turned off, or if a power or clock fault occurs, the sequencer opens the relay, stops the buck regulator and disables the H-bridge chopper.

Supply Regulators

Each of the circuits described above contributes to the functioning of the main supply regulators. The most important components of the off-line switching regulator are shown at the top of Figure 5.5.

The off-line switcher produces only one regulated output. Consequently, circuits are enlisted to regulate the remaining outputs. Because the power demands on these outputs are relatively low, linear series pass regulators are used. The 10-volt transformer winding powers the **−5V** regulator. The 14-volt transformer winding powers the **+12V** regulator, which regulates ac power from the transformer before rectifying it. The 4-volt transformer winding powers the **3V** regulator, which rides on top of the **12V** regulator to provide a **+15V** output.

The remaining circuits detect various kinds of power faults. They protect the supply from excessive loads, protect the computer from excessive voltages, and flag power failures.

The *over-current detector* monitors the following signals and signal sources:

- **12CS** signal indicates that the **+12V** regulator is overloaded.
- **SC DET** signal indicates a short circuit on one of the supply outputs (except **−5V** or **−5 MEM-P**). (Short circuits on **+5V** are detected on the backpanel.)
- The *primary current limiter* monitors the total current flow into all the loads. If an over-current condition occurs, the limiter immediately drives the **PON** signal to the low state and disables the buck regulator for the remainder of the 40 KHz clock cycle. (The regulator turns on again at the start of the next cycle.) This effectively reduces the duty cycle of the pulse width modulator and thus limits the amount of power delivered. (The supply actually goes into full current limiting when first turned on, in order to charge the output filter capacitors.) If the supply stays in current limiting too long, the limiter circuits drive the **RP8** signal to the high state.

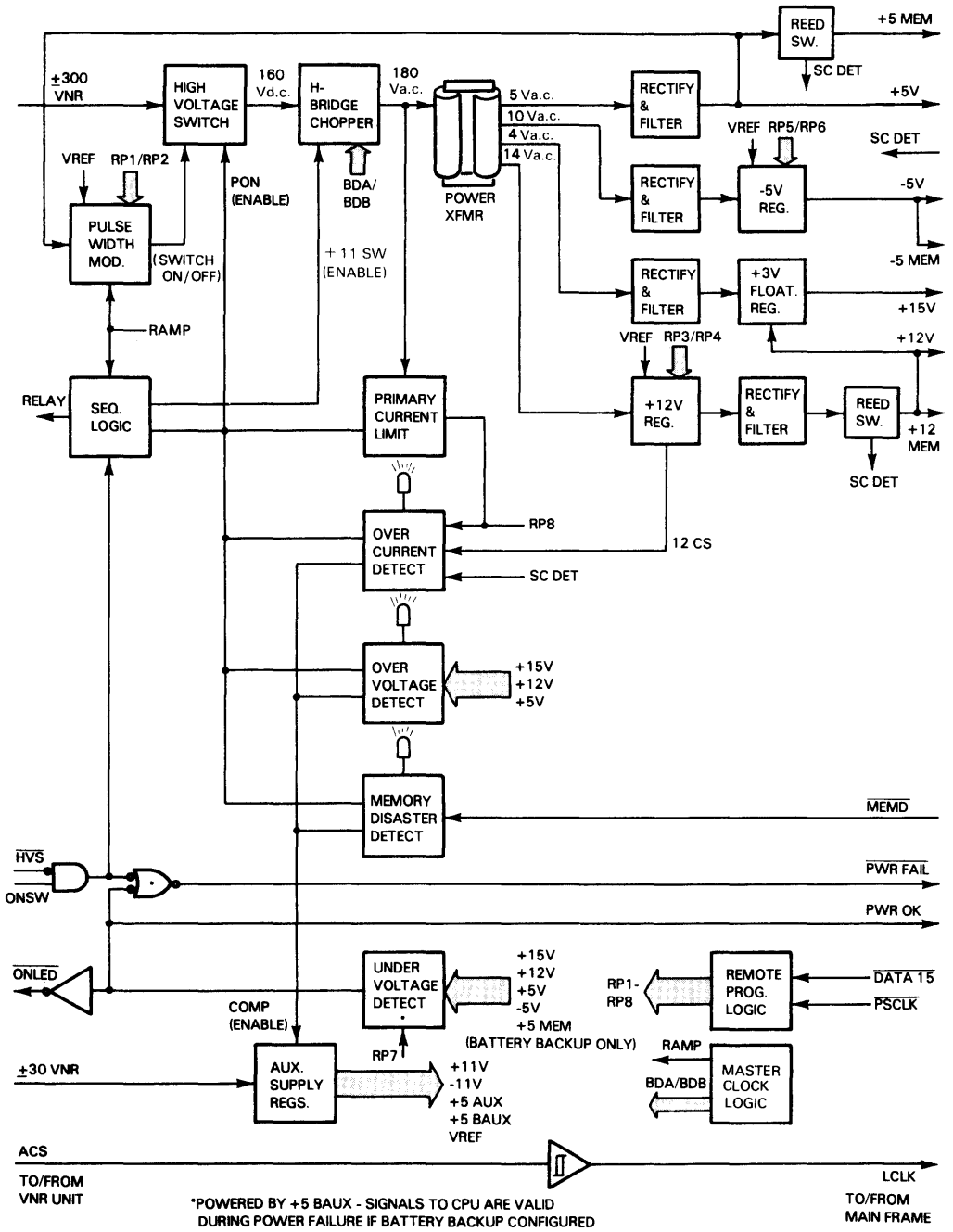
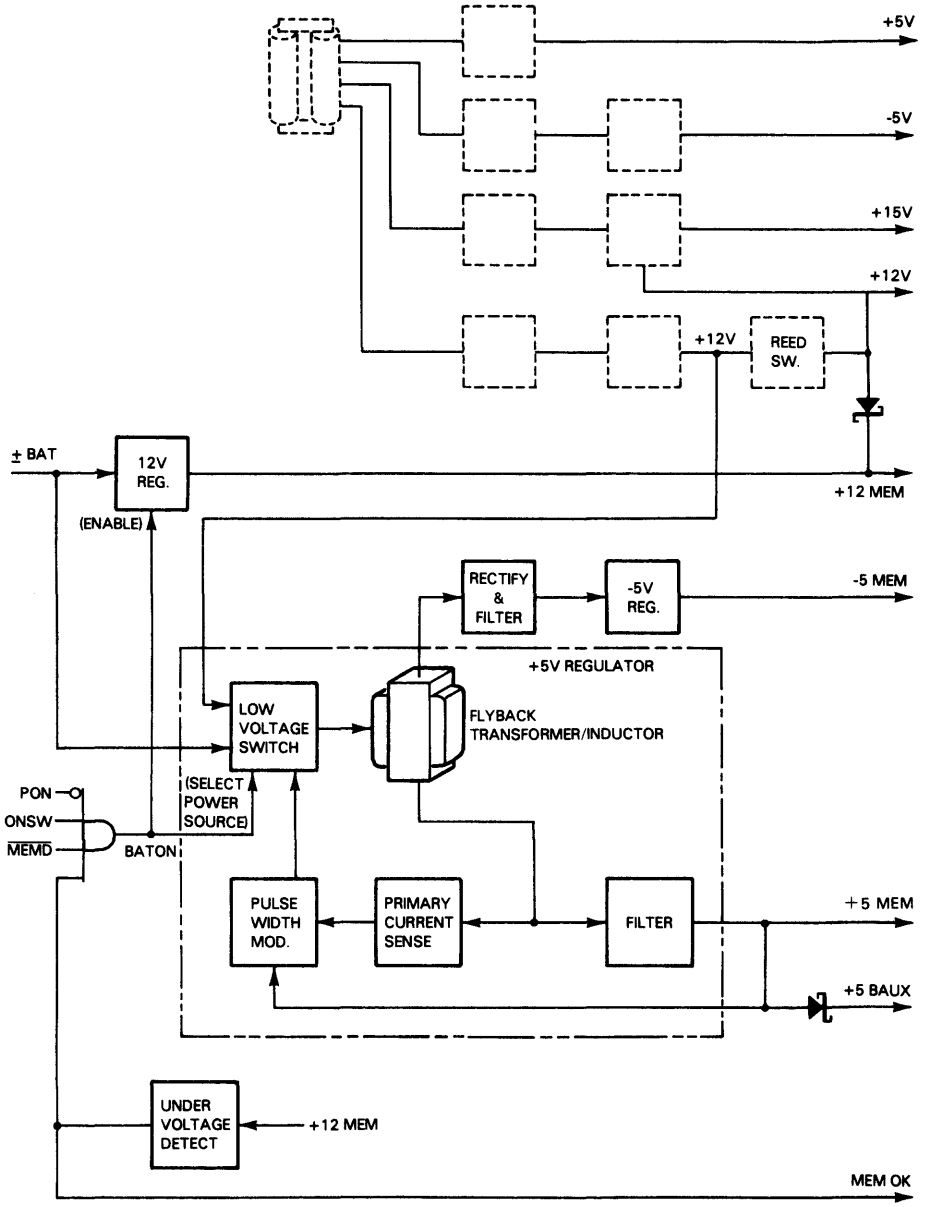


Figure 5.5 Power supply

When an over-current fault occurs, the over-current light turns on and the detector drives the **PON** and **COMP** signals to the low state. The events shut down the entire supply. The detector reenables the supply two seconds later. If the overload is still present, the detector shuts the supply down again. This process can continue for up to five additional cycles. If the overload still exists after the sixth cycle, the supply shuts down and is not reenabled.

The *over-voltage detectors* monitor the **+15V**, **+12V**, and **+5V** outputs. If any of these outputs exceed a preset voltage level, the over-voltage light turns on and the detector shuts down the supply. (The **−5V** regulator has a built-in over-voltage protector. If a fault occurs, the protector clamps the **−5V** bus to less than −0.8 volts.)

The *memory disaster detector* monitors the **MEMD** signal from the main memory on the CPU board. If main memory loses **-5 MEM** a critical memory voltage, the memory disaster light glows and the detector shuts down the supply.



DG-05757

Figure 5.6 Battery backup

The *under-voltage detector* monitors all output voltages. If any fall below a preset level, the detector drives the **POWER OK** signal to the low state, turns off the power-on light, and flags a power failure. A power failure is also flagged when the power switch is turned off or the 300 VNR supply fails.

Battery Backup

The battery backup feature supports the memory voltages when a power failure occurs. It also powers the system clocks and the power status indicators and provides a **MEM OK** status flag to indicate that the memory voltages are functioning properly. Figure 5.6 shows the parts of the battery backup feature that are added to the power supply board.

The battery backup feature generates three voltages: **+12 MEM**, **+5 MEM-P**, and **−5 MEM**. Note that the main supply regulators support these voltages when battery backup is not configured. The battery backup draws power from one of two sources. During normal operation, power comes from the **+12V** main supply output. If a failure occurs, the battery supplies power.

The backup feature includes three voltage regulators along with some control circuits. The following text explains how these circuits work by first examining each voltage regulator and then following a power failure sequence. Finally, Table 5.2 lists the specification for the 16-slot chassis power supply with battery backup present.

Output	Voltages Min	Voltages Max	Current Min	Current Max
+5V	+5.10V	+5.20V	7.5A	100A
+12V	+12.1V	+12.5V	0	12.5A*
+15V	+14.5V	+15.5V	0	1.5A*
−5V	−4.9V	−5.2V	0	3A
−11V	−11.0V	−12.5V	0	0.02A
+5 MEM	+4.8V	+5.1V	0.25A	9.5A*
+12 MEM	+11.7V	+12.1V	0	6A*
−5 MEM	−4.7V	−5.1V	0	0.3A*

Table 5.2 16-slot chassis power supply specifications with battery backup

**The sum of the currents on +12V, +15, +12 MEM, and 0.55 times the sum of the currents on +5 MEM and −5 MEM must not exceed 12.5 Amps.*

Voltage Regulators

The **+12V** regulator draws power directly from the battery. It is designed as a simple, linear series pass *without* built-in protection circuits.

The **+5V** regulator draws power from either the **+12V** main supply or from the battery. It is a buck switching regulator, similar to that present in the main supply. When the low voltage switch turns on, current flows from the source, through the primary winding of the flyback transformer, and on to the output. The pulse width modulator controls the duty cycle of the switch to regulate the output voltage. If too much current flows in the primary winding, the current limiter reduces the modulator's duty cycle.

The **−5V** regulator draws power from the **+5V** regulator via the flyback transformer. It is also designed as a simple, linear series pass, *with* built-in current limiting.

The battery on (**BATON**) signal controls battery backup operation. When the supply operates normally, the **PON** signal is driven high and the **BATON** is driven low. These conditions turn off the **+12V** regulator. Current then flows from the **+12V** supply, through the Schottky diode, and on to the **+12 MEM** output. Simultaneously, the low voltage switch selects the **+12V** supply, which in turn powers the **+5 MEM** and **−5 MEM** outputs.

Power Failure Sequence

When a power failure occurs, the **PON** signal is driven to the low state. The **BATON** signal is driven to the high state, unless the failure occurred because of a memory disaster. These changes cause a switch from regulator to battery operation. As long as the battery retains sufficient charge, the **+12 MEM** output stays in regulation and the **MEM OK** signal remains in the high state. However, when the battery discharges to a dangerously low level, the under-voltage detector drives the **MEM OK** signal to low; **BATON**, in turn, is driven low as well. As a result, the supply shuts down entirely.

Interconnection with the System

The 16-slot power supply board communicates with the rest of the system via jacks 17 and 35 on the backpanel. Tables 5.3 and 5.4, respectively, list the voltage and power supply status signals generated or received by the power supply board, together with the jack locations of the signal. The clock signal **LCLK** is generated from the power supply and sent to the backpanel. It is a 50/60 Hz square wave (ac line frequency) and is located on pin J35-17. Refer to the 16-slot backpanel schematic, DGC No. 001-003152, for the locations of signals on the backpanel in compliant systems, and schematic DGC No. 001-001563 for noncompliant systems.

The last table (5.5) lists and explains the signals that appear on the test points (J1) on the front of the 16-slot power supply board.

Signal	Jack Pin	Source	Destination	Description
GND	J17 even pins J35 pins 1-2, 4, 37-42	Power supply	Backpanel	Power or logic ground
+5V	J17 odd pins	Power supply	Backpanel	+5V source
+5 MEM-P	J35 pins 33-36	Power supply	Backpanel	Same as +5V if battery backup not configured
−5V	J35-27 J35-28	Power supply	Backpanel	−5V source
−5 MEM-P	J35-19	Power supply	Backpanel	Same as −5V if battery backup not configured
−11V	J35-25	Power supply	Backpanel	EIA interface voltage
+12V	J35 pins 43-46	Power supply	Backpanel	+12V source
+12 MEM	J35 pins 29-32	Power supply	Backpanel	Same as +12V if battery back-up not config-ured
+15V	J35-49 J35-50	Power supply	Backpanel	+15V source

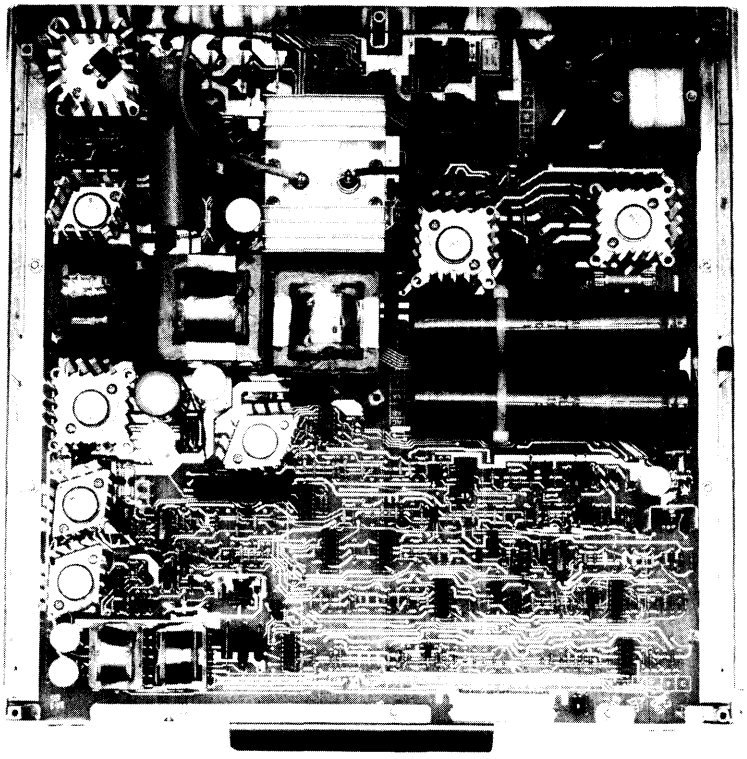
Table 5.3 Voltage signals

Signal	Jack Pin	Source	Destination	Description
DATA 15	J35-3	Power supply	Backpanel	Reserved for future use
MEMD	J35-21	Backpanel	Power supply	Failure on −5 MEM
MEMOK	J35-23	Power supply	Backpanel	+12 MEM voltage ok
ONLED-P	J35-11	Power supply	Backpanel	All dc output voltages ok
PWR FAIL	J35-9	Power supply	Backpanel	Power switch off or dc power failure
PWR OK	J35-16	Power supply	Backpanel	All dc output voltages ok
SC DET	J35-47	Power supply Backpanel	Backpanel Power supply	Short circuit on +5V, +5 MEM, +12V, or +12 MEM

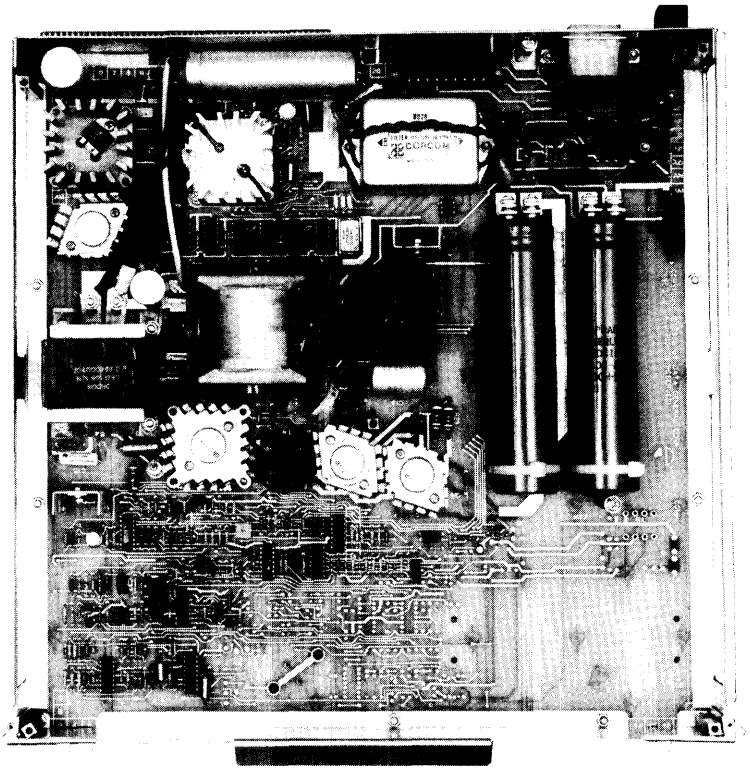
Table 5.4 Power supply status signals

Signal	J1 Pin	Description
+15V	8	Same as +15V supply on backpanel.
+12V	10	Same as +12V supply on backpanel.
+5V	15	Same as +5V supply on backpanel.
+5 AUX	7	Internal +5V supply for the power supply logic. (All supply functions are disabled when there is no +5 AUX. The VNR unit supplies +30V for the +5 AUX regulator.)
−5V	6	Same as −5V supply on backpanel.
VREF	14	Internal +5.80V reference for all regulators. (All internal and external supply voltages go out of regulation if VREF fails.)
GND	9	Power and logic ground.
HVS	4	High voltage from the VNR unit exceeds 260V. (All external supply voltages except −11V shut down if high voltage fails and battery backup comes on-line if configured.)
PON	11	External supply voltage regulators are enabled. (Goes to low state if HVS is in low state, an emergency condition such as a memory disaster occurs, an internal failure occurs, or power switch turned off. Battery backup comes on-line when PON goes to low state if power switch is on.)
POK	5	All external supply voltages ok.
PWR FAIL	3	POK or HVS in low state or power switch is turned off.
PWM	1	40 KHz pulse train from pulse width modulator. Clocks the main switching regulator which powers the external supply voltage regulators.
20 K	16	20 KHz square wave. Clocks the circuits which transform high voltage from the main switching regulator to low voltages for the external supply voltage regulators.
Q30-C	12	20 KHz drive signal in the power transformation circuits mentioned above.
Q29-C	13	20 KHz drive signal in the power transformation circuits mentioned above.
CURRENT	2	Voltage proportional to current flow in the primary winding of power transformer that converts high voltage from the main switching regulator to low voltage for the external supply voltage regulators.

Table 5.5 Test point signals



Compliant power supply PCB



Noncompliant power supply PCB

PH-0775 PH-0579

Figure 6.1 Compliant and noncompliant 5-slot power supply PCB

Chapter 6

5-Slot Power Supply

The compliant and noncompliant 5-slot power supplies differ and are discussed separately.

Compliant 5-slot Power Supply

The EMI compliant 5-slot power supply converts a 100/120 or 220/240 ac voltage power source to the regulated dc voltages required by the computer system boards in a five-slot computer chassis. In addition, the power supply contains circuitry to generate power status and system clock signals. A battery backup option maintains the dc voltages necessary to retain system memory data when a power failure occurs.

The entire power supply, including the battery backup option, is contained on a single printed circuit board surrounded by a metal frame and top cover. The power supply's printed circuit board plugs into two connectors on the backpanel of the computer chassis. These connectors furnish ac source power to the power supply board and, at the same time, deliver status signals and dc power to the system's printed circuit boards. If you have the optional battery backup, a 10-volt battery is mounted inside the power supply cage and connected to the power supply board. This battery provides dc power to the backup circuits when a power failure occurs.

The ac power source reaches the power supply board through the:

line cord,
line filter assembly (including a line fuse),
power switch on the front control panel,
internal chassis cable,
backpanel etch,
backpanel ac line voltage selection plug.

The line cord plugs directly into an ac jack of the line filter assembly. Two line cords are available; one for 100/120 Vac power source and the second for 220/240 Vac when the chassis is mounted in a Data General rack.

The power supply is actually two power supplies, the main supply and the battery-backed supply. The main supply produces +5V, −5V, +12V, and −12V for the system boards as well as +3V for the backpanel terminators. The battery backup supply produces the +5MEM, −5MEM, and +12MEM for the memory. Both supplies operate during normal mode (ac input power above minimum threshold value). The supply voltages come up in sequence; first +5V, −5V, −5MEM, +3V, and −12V; then +12V followed by +5MEM and +12MEM. Table 6.1 provides the dc output specifications for the power supply.

	Voltage		Current	
Output	Min	Max	Min	Max
+5V	+4.95V	+5.25V	5A	40A[1]
+12V	+11.5V	+12.5V	0	5A[2]
−5V	−4.70V	−5.30V	0	2A
−12V	−11.0V	−13.0V	0	0.025A
+3V	+2.70V	+3.30V	0	1.2A
+5MEM	+4.95V	+5.25V	0	5A[3]
+12MEM	+11.5V	+12.5V	0	2.3A[2]
−5MEM	−4.70V	−5.30V	0	0.05A[5]

Table 6.1 5-slot chassis power supply specifications

[1]The sum of the currents on the +5VA, +5VB, and +5VC outputs.

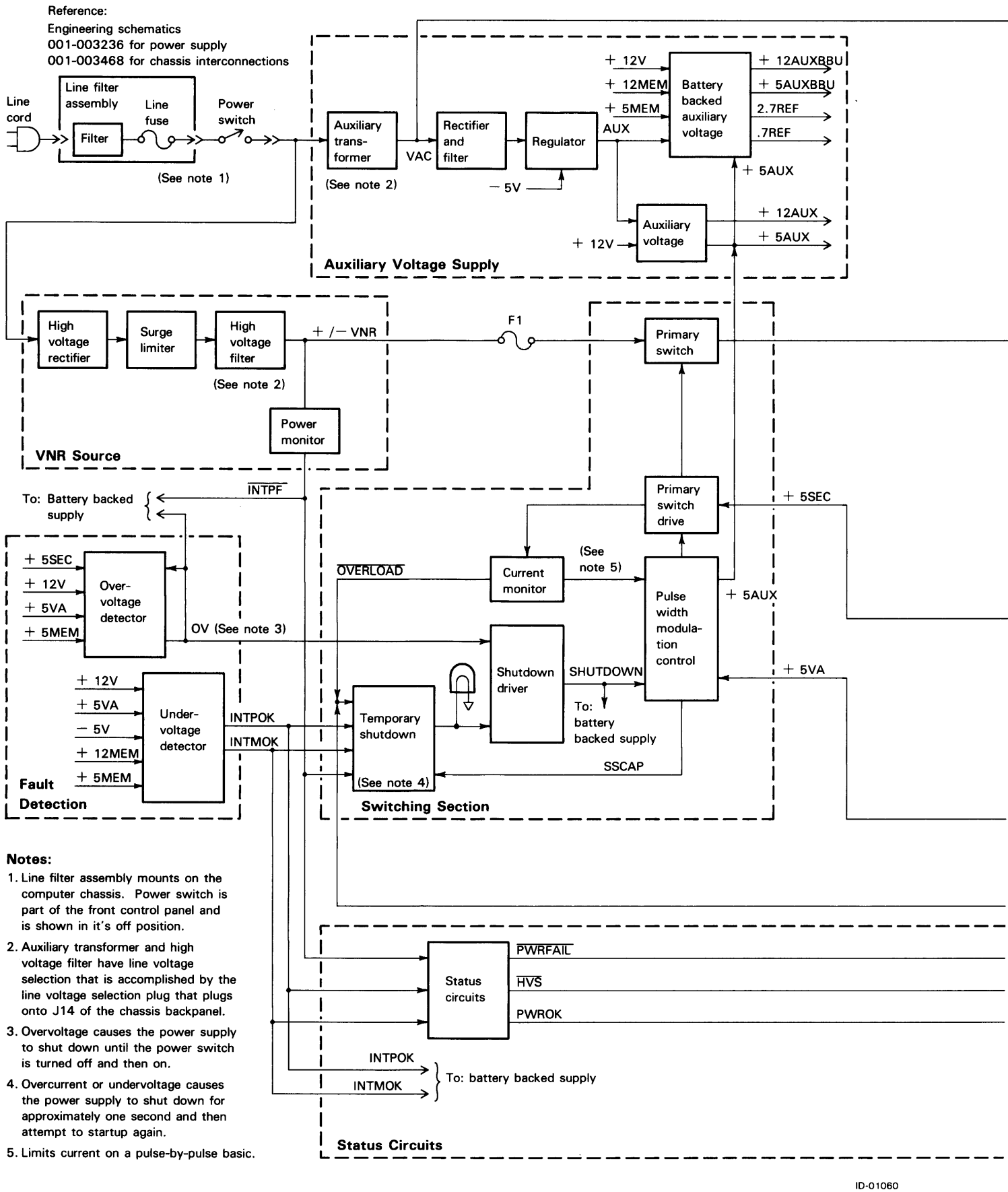
[2]The sum of the maximum currents on the +12V and +12MEM outputs must not exceed 5 Amps.

[3]Maximum current on the +5MEM output during battery backup operation is 5A.

[4]Maximum current on the +12MEM output during battery backup operation is 0.3A.

[5]Maximum current on the −5MEM output during battery backup operation is 0.05A.

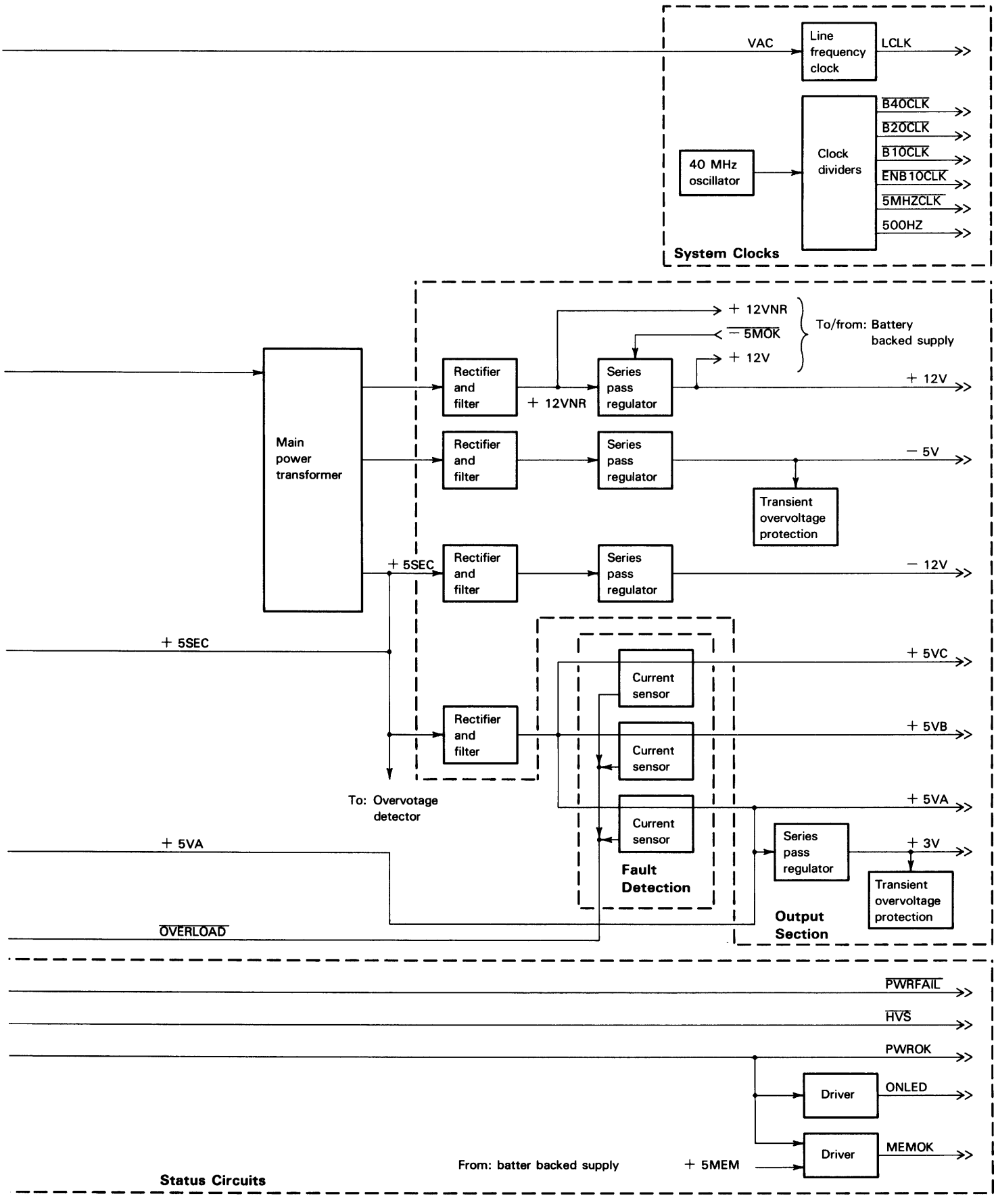
When an input ac power failure takes place, the main supply turns off, leaving only the battery-backed supply running. Naturally, the battery backup will not take over if the power switch on the front control panel is in the off position.



Notes:

1. Line filter assembly mounts on the computer chassis. Power switch is part of the front control panel and is shown in it's off position.
2. Auxiliary transformer and high voltage filter have line voltage selection that is accomplished by the line voltage selection plug that plugs onto J14 of the chassis backpanel.
3. Overvoltage causes the power supply to shut down until the power switch is turned off and then on.
4. Overcurrent or undervoltage causes the power supply to shut down for approximately one second and then attempt to startup again.
5. Limits current on a pulse-by-pulse basic.

Figure 6.2 Compliant 5-slot chassis power supply


VAC
Line frequency clock
LCLK
B40CLK
B20CLK
B10CLK
ENB10CLK
5MHZCLK
500HZ
40 MHz oscillator
Clock dividers
System Clocks
+ 12VNR
− 5MOK
+ 12V
To/from: Battery backed supply
Main power transformer
Rectifier and filter
Series pass regulator
+ 12V
+ 12VNR
− 5V
Transient overvoltage protection
+ 5SEC
− 12V
+ 5SEC
+ 5VC
Current sensor
+ 5VB
Current sensor
+ 5VA
Current sensor
To: Overvotage detector
+ 5VA
Series pass regulator
+ 3V
Transient overvoltage protection
Fault Detection
Output Section
OVERLOAD
PWRFAIL
HVS
PWROK
Driver
ONLED
Driver
MEMOK
From: batter backed supply
+ 5MEM
Status Circuits

When the power supply is operating in normal mode, it monitors all output voltages except +3V and −12V for an undervoltage condition. If it detects an undervoltage, the power supply will temporarily shut down and wait approximately one second before attempting to start again. Overcurrent conditions on the +5V outputs or internal circuitry of the supply will also cause a temporary shutdown. In addition, when it is operating, the power supply monitors the +5V, +12V, and +5MEM output voltages, as well as an internal power source for overvoltage. An overvoltage condition causes the power supply to shut down until the power switch on the front control panel is turned off and then on.

When the supply is operating in battery backup mode, it monitors the three memory voltages, +5MEM, −5MEM, and +12MEM for an undervoltage condition, and the +5MEM for an overvoltage condition. If it detects either condition, the battery-backed supply shuts off until ac power is restored and the main supply is operating in normal mode.

Note that turning off the power switch on the front control panel when the supply is operating in battery backup mode turns off the battery-backed supply.

Functional Overview

The power supply uses two switching regulators and two output sections to produce the dc voltages for the chassis in which it is located. A forward off-line switching regulator and an output section produce the main power supply dc voltages. A low voltage switching regulator and an output section produce the battery-backed dc voltages. In addition, the power supply includes an auxiliary voltage supply section, various fault detection circuits, and the system clock circuitry.

Main Power Supply

The major components of the main power supply's off-line switching regulator are:

- nonregulated high voltage (VNR) source section,
- switching section,
- power transformer,
- output section.

These components and the auxiliary voltage section are shown in Figure 6.2 and described in the following sections. The figure also shows the power supply's fault detection circuits, and the system clock circuitry.

Auxiliary Voltage Supply

The auxiliary voltage supply section:

- powers most of the internal circuits of the power supply,
- generates reference voltages for internal power supply circuits,
- provides an ac signal (**VAC**) to the line frequency clock section of the system clock circuitry.

This supply consists of a transformer, rectifier, filter, +12V regulator, and the auxiliary voltage circuits. The transformer receives power from the ac source through the line filter, line fuse, and the power switch on the front control panel when that switch is in the on position.

During startup, the +12V output of the regulator supplies power to the auxiliary voltage circuits. This produces **+12AUX** and **+12AUXBBU**. In turn, **+12AUX** produces **+5AUX**,and **+12AUXBBU** generates the two reference voltages, **2.7REF** and **.7REF**. **+5AUX** provides **+5AUXBBU**.

Once the power supply is in normal operation, **−5V** shuts the auxiliary regulator off and the +12 auxiliary voltages and reference voltages are produced by the **+12V** output of the power supply. At the same time, the +5 auxiliary voltages are produced by an output of the pulse-width modulator.

During an ac power failure, when the battery backup option is present and in operation, the **+12AUXBBU** and reference voltages are produced by **+12MEM**, while the **+5AUXBBU** is produced by **+5MEM**.

Nonregulated High Voltage (VNR) Source

The VNR section circuitry converts ac line voltage into a high dc voltage source. This section consists of a rectifier, a surge limiter, a filter, and a power monitor. The section receives power from the ac source through the line filter, line fuse, and the front control panel power switch, when that switch is turned on.

During startup, the surge limiters control inrush current while the VNR filter capacitors charge to the VNR voltage level.

For 100/120 volt ac operation, the VNR source section rectifies and doubles the ac line source voltage by connecting the junction of the two VNR filter capacitors to the neutral side of the ac line. For 220/240 volt ac operation, the VNR source rectifies only the ac line source. Ac line voltage is selected by a connector that plugs onto J14 of the chassis' backpanel.

Power Monitor

VNR voltage level is monitored by the power monitor circuitry. During startup, this circuitry applies **INTPF** to the temporary shutdown circuitry of the pulse-width modulation section. This signal, in conjunction with an inactive **SSCAP** from the pulse-width modulation control circuitry, asserts **SHUTDOWN**. **SHUTDOWN** prevents the pulse-width modulation control from starting the main power supply until VNR voltage is high enough. Once it is sufficient, **INTPF** asserts high. After approximately one second, **SHUTDOWN** goes low, allowing the pulse-width modulation control to start the main supply. During startup, **INTPF** also causes the status circuits to assert **PWRFAIL** to the processor.

During normal supply operation (+5V output in regulation) the power monitor applies **INTPF** if VNR voltage falls below a threshold level indicating a low ac source or impending ac source failure. In this case **INTPF** applies to the supply's status circuits. The status circuits assert **PWRFAIL** to notify the processor of the power fault.

Switching Section

The switching section converts nonregulated dc power (VNR) to a pulsed high voltage. It applies this voltage to the primary winding of the power transformer by closing and opening a power path at a 30 KHz rate. This section also controls the amount of power applied to the transformer by varying the time the power path is closed according to variations on the +5V output. For example, as the +5V output load increases, or the ac line voltage decreases, the pulse-width modulation control increases the closed time to transfer more power.

The switching section consists of: the primary switch and its drive circuit, the pulse-width modulation control, the current monitor, and the shutdown circuitry. The current monitor is explained under "Fault Detection" later in the compliant section of this chapter.

During startup, this section is prevented from being started by the **SHUTDOWN** signal until VNR reaches a sufficient voltage level. (When both **INTPF** and **SSCAP** are low, **SHUTDOWN** is applied to the pulse-width modulation control.) An indicator, located at the front edge of the power supply board, glows during a temporary shutdown. Approximately one second after VNR reaches a sufficient level, the following sequence takes place:

(**INTPF** goes high),

(**SHUTDOWN** becomes inactive,

the pulse-width modulation control starts to operate, applying pulses to the primary switch drive circuit at a 30 KHz rate.

The drive circuit turns the primary switch on, closing the power path, each time a pulse asserts. These pulse start narrow and then widen as a soft-start capacitor charges until the +5V output is in regulation.

During operation, the primary switch is prevented from turning on during the power transformer's core reset time. This is accomplished by the **+5SEC** signal from the +5 secondary winding of the transformer.

Main Power Transformer

The main power transformer converts pulsed high voltage at low current to pulsed low voltages at high current. Three secondary windings apply the low voltages to the output section. In addition, the +5V secondary controls the primary switch drive circuit to prevent power from being applied to the transformer primary until the transformer core resets from the previous pulse cycle. Also, the +5V secondary goes to the overvoltage detector (refer to "Fault Detection" in the compliant portion of this chapter for detail).

Output Section

The output section converts three pulsed voltages into five highly regulated low dc voltage outputs: +5V, −5V, +12V, −12V, and +3V. This section consists of four rectifier and filter circuits, and four series pass regulator circuits.

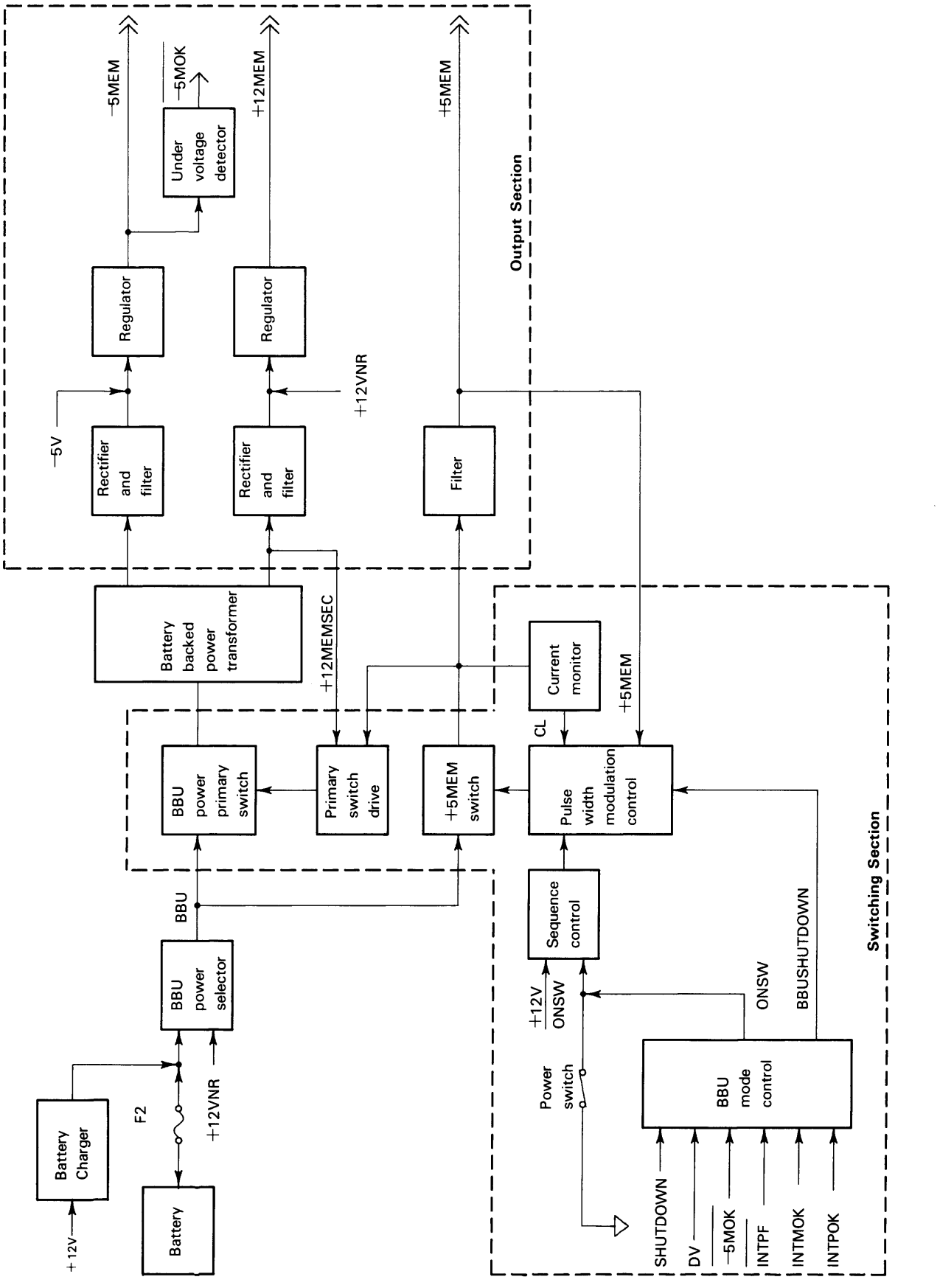
Power from the +5V secondary winding of the power transformer (**+5SEC**) is rectified and filtered and feeds directly to three power supply outputs: +5VA, +5VB, and +5VC. Each of the three +5V outputs feeds through a seperate board etch and a reed switch with a half-turn winding that functions as a current sensor. These current sensors cause a temporary power supply shutdown if the current on their respective outputs exceeds approximately 22 amps. (Refer to "Fault Detection" in the compliant portion of this chapter for details.)

The +5VA output feeds back to the pulse-width modulation control to vary the closed time of the switching regulator's power path, thus regulating the +5V outputs. In addition, +5VA goes to a series pass regulator to produce the +3V output, which is protected against transient overvoltages and only powers the backpanel terminators.

Power from the +5V secondary winding of the power transformer (**+5SEC**) is also rectified, filtered, and regulated to produce the −12V output.

Power from the −5V secondary winding of the power transformer is rectified, filtered, and regulated to produce the −5V output. This output is also protected against transient overvoltages.

Power from the +12V secondary winding of the power transformer is rectified and filtered to produce a nonregulated low voltage (**+12VNR**). This nonregulated voltage provides a VNR source power to the battery-



ID-01061

Figure 6.3 Compliant 5-slot chassis battery-backed power supply

backed supply while the main supply is operating, and is regulated to produce the +12V output. During start-up, the +12V regulator circuit is disabled until −5V is present (**−5MOK**). The regulated +12 volts also goes to the battery charger circuit to recharge the battery as required when the main supply is operating.

Battery-backed Supply

The battery-backed supply produces the memory voltages during normal operation. It also produces the memory voltages during a power failure when the battery backup option is present, and the front control panel power switch remains in the on position. When the power supply board is operating in battery backup mode, the front control panel battery indicator is on. This supply includes a nonregulated voltage source, switching section, power transformer, output section, and battery charger. Figure 6.3 shows the components of the battery-backed power supply.

The length of time data can be retained during a power failure depends on the amount of system memory present in the chassis and on the amount of charge in the battery. Battery backup time can be increased by mounting an additional 10-volt battery outside the computer chassis and connecting it to the backpanel of the chassis. However, the built-in charging circuit is designed to charge only the internal battery, and that circuit's performance may be reduced or it may be damaged by the use of an additional battery.

Nonregulated Voltage Source

This circuitry supplies nonregulated +12 volts (**BBU**) during normal power supply operation, and voltage from the battery during an ac power failure. This voltage powers the battery-backed supply circuits.

Battery-backed Switching Section

Two switching circuits convert nonregulated dc power (**BBU**) to a pulsed voltage. The first circuit — the +5MEM switch — supplies power that is filtered to produce the +5MEM output. This switching circuit supplies the power by closing and opening a power path at a 30 KHz rate. The circuit also controls the amount of power supplied to the +5MEM output by varying the power path's closed time according to variations on that output. The second circuit — the primary switch and its drive circuit — supplies power to the battery-backed power transformer and is controlled by the +5MEM switching circuit.

In addition, the switching section consists of the: battery-backed pulse-width modulation control, sequence control, mode control, and current monitor. The current monitor is explained under "Fault Detection" later in this section on complaint devices.

During start-up, this section is prevented from starting until the +12V output of the main supply is present to the sequence control circuit. When +12V reaches a referenced level, the battery-backed pulse-width modulation control starts to operate, applying pulses to the +5MEM switch circuit. The +5MEM switch closes the power path to the +5MEM filter each time a pulse asserts. These pulses start narrow and then widen as a soft-start capacitor charges until the +5MEM output is in regulation.

Output from the +5MEM switch also controls the primary switch drive circuit, thus controlling the primary switch. When the primary switch turns on, the power path to the battery-backed power transformer is closed. During operation, the primary switch is prevented from turning on until the power transformer's core is reset by the **+12MEMSEC** signal from a secondary winding of the transformer.

The mode control circuitry places the battery-backed supply in battery backup mode when an ac power failure (one not caused by shutting off the power switch) is detected. Placing the front control panel power switch in the off position always shuts down the battery-backed supply. The mode control circuitry also shuts down the battery-backed supply when the following conditions exist:

- The main supply shuts down because of an overvoltage, undervoltage, or overcurrent condition (**SHUTDOWN** asserts high).
- Any memory voltage falls below sufficient level to maintain memory data during battery backup mode (**INTMOK** asserts low).
- Undervoltage on the −5MEM output during battery backup mode (**−5MOK** asserts high).
- Overvoltage on the +5MEM output during battery backup mode (**OV**).

Power Transformer

The battery-backed power transformer converts pulsed voltage at low current to pulsed voltage at a higher current. Two secondary windings apply the pulsed voltages to the output section. In addition, the +12MEM secondary goes to the primary switch drive circuit to prevent power from being applied to the transformer primary. The transformer core resets from the previous pulse cycle.

Output Section

The output section filters the +5MEM output voltage and converts two pulsed voltages into two regulated dc memory voltages.

Battery Charger

The battery charger charges and/or maintains the internal battery while the supply is operating in normal mode. The charger circuit is powered from +12V.

Fault Detection

Several circuits monitor the operation of the power supply, checking for undervoltage, overvoltage, and overcurrent conditions.

Undervoltage Detection

Three undervoltage detection circuits continually monitor the +5V, +12V, −5V, +12MEM, +5MEM, and −5MEM outputs. If any of these outputs falls below a minimum operating level, its respective detector temporarily shuts down the main and battery-backed pulse-width modulation control circuits, thus shutting down both power supplies for approximately one second. Then the supply will attempt to start up again. (An indicator, located at the front edge of the power supply board, lights during a temporary shutdown.)

Undervoltage detected on the +5V, +12V, or −5V output, will cause signal **INTPOK** to assert low; undervoltage detected on +12MEM or +5MEM will cause **INTMOK** to assert low. Either of these signals asserts **SHUTDOWN**, causing the entire supply to shut down temporarily. Either of these signals causes the status circuits to assert **PWROK** low to the processor.

Undervoltage detected on the −5MEM output, will cause signal **$\overline{-5MOK}$** to assert low. This signal asserts **BBUSHUTDOWN**, causing the battery-backed supply to shut down.

Overvoltage Detector

The overvoltage detector continually monitors the +5V, +12V, and +5MEM outputs, and the +5 secondary winding of the main power transformer. If any of these outputs exceeds its respective shutdown level, this detector permanently shuts down the main and battery-backed pulse-width modulation control circuits, thus shutting down both supplies. When the supply shuts down in this manner, the front control panel power switch must be turned off, and then on, to allow the supply to try a restart.

Current Monitoring

Four current monitoring circuits check for overcurrent conditions during the operation of the power supply. Two of these circuits limit current by shutting down the pulse-width modulation control of either the main or the battery-backed power supply. This shutdown occurs only if the current in the primary windings of their respective power transformers exceeds their referenced levels. The other two circuits shut down the entire power supply for approximately one second, after which the supply attempts to start up again. (An indicator, located at the front edge of the power supply board, lights when there is a temporary shutdown.)

Main Power Supply Current Monitor

Current through the primary winding of the main power transformer is sensed by the primary switch drive circuit and monitored by two current monitors. If this current exceeds a referenced level during the closed time of the power path, one current monitor, (a pulse-by-pulse limiter), shuts down the main pulse-width modulation control until its next cycle. The second current monitor asserts **$\overline{OVERLOAD}$** if the current exceeds a reference level higher than that of the first. **$\overline{OVERLOAD}$** causes the temporary shutdown circuit to assert **SHUTDOWN**. This condition shuts down the entire power supply for approximately one second, and then the supply attempts to start up again.

+5V Output Current Sensing

Each of the three +5V outputs is monitored by a current sensor. These current sensors consist of a reed switch with a half-turn winding that connects in series with each respective +5V output.

If the current on any of the three +5V outputs exceeds approximately 22 amps, the respective reed switch asserts **$\overline{OVERLOAD}$**. This signal causes the temporary shutdown circuit to assert **SHUTDOWN**, which shuts the entire power supply down for approximately one second, after which the supply attempts to start up again.

Battery-backed Power Supply Current Monitor

Current on the +5MEM output is monitored by a pulse-by-pulse current limiter. If this current exceeds a reference level during the closed time of the +5MEM switch, the limiter shuts down the pulse-width modulation control until its next cycle. This action opens the power path. until .

System Clocks

The system clock circuitry generates several system clock signals for the processor. One, an ac line frequency clock signal, is generated by the secondary winding of the auxiliary voltage transformer through a pulse shaping circuit. The remaining clock signals are generated by a 40 MHz crystal oscillator through divider circuits, and are synchronized with each other.

Signal	Connector Pin	Source	Destination	Description
PWRFAIL	B17	Power supply	Backpanel (J11)	ac power failure or power switch just turned off
PWROK	B20	Power supply	Backpanel (J11)	All monitored dc output voltages ok
MEMOK	B16	Power supply	Backpanel (J11)	Memory voltages were still valid when ac line power returns following a ac line power failure. (This signal remains low for a minimum of 15 milliseconds after PWROK asserts high on an initial power up or a power up when memory voltages failed to remain powered during an ac line power failure)
ONLED	B18	Power supply	Front control panel through backpanel (J11)	All monitored output voltages ok. (When low, this signal turns on the front control panel power indicator.)
ONSW	A55	Front control panel through backpanel (J12)	Power supply	Specifies the position of the front control panel power switch. (When high, causes the battery backed supplies to remain on during an ac line power failure if the battery backup option is present.)
HVS	A56	Power supply	Backpanel (J11)	When low, specifies VNR source power is below referenced level indicating a low ac line voltage or impending ac line failure. (Presently unused)

Table 6.2 Status and Control Signals

Status Circuits

The status circuitry generates several status signals to the processor. It also generates a control signal to turn on the front control panel power indicator when the monitored dc output voltages are within their referenced levels. These signals are described in Table 6.2. Timing diagrams for the three status signals, **PWRFAIL**, **PWROK**, and **MEMOK** are shown in Figures 6.4 through 6.8.

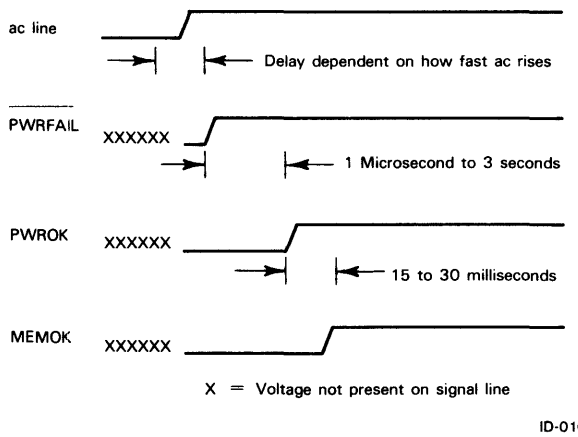


Figure 6.4 Initial powerup status signal

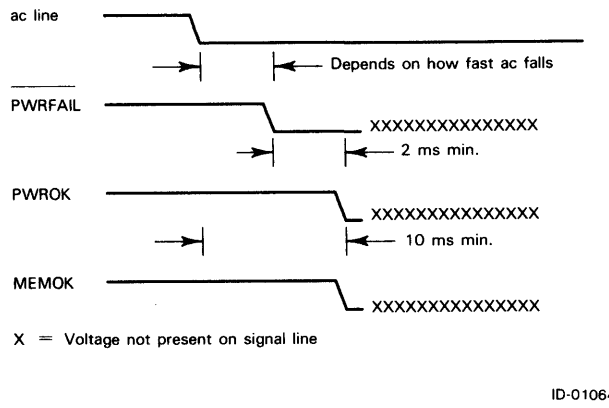


Figure 6.5 Power down or ac power loss without battery backup

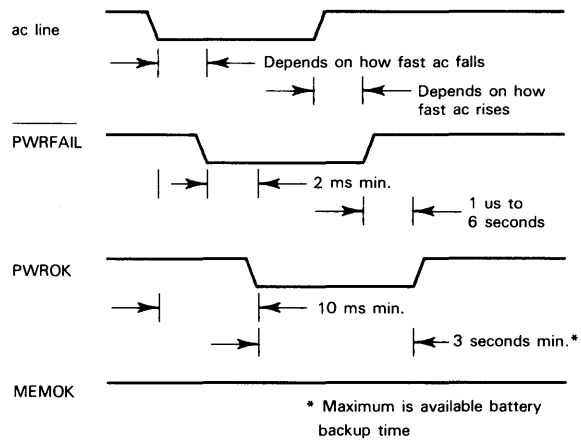


Figure 6.6 Power failure not exceeding available battery backup time

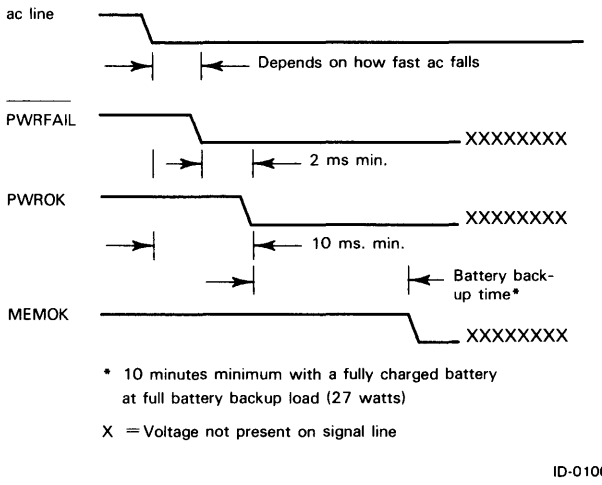


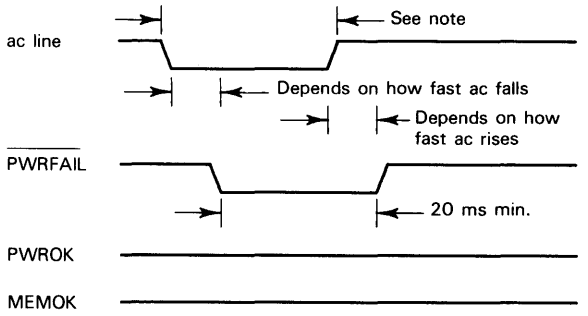
Figure 6.7 Power failure exceeding available battery backup time

Test Points

An eight-pin connector, located on the front edge of the power supply board and accessible through the front of the power supply assembly's metal frame, provides test points to check the dc output voltages of the power supply. From left to right, the test points are: −5MEM, +5V, +5MEM, +12MEM, −5V, +3V, −12V, +12V.

Fault Indications

The fault conditions shown in Table 6.3 can be identified by two indicators. One is located on the front edge of the power supply board, and the other — the power indicator — on the front control panel. The indicator on the power supply board can be seen through the supply's metal frame after the front panel and the RFI shield have been removed.



Note: PWROK will remain asserted for a minimum of 10 millisecond with a complete loss of ac line and a full power load on the supply. With less than full load PWROK will remain asserted for a longer period of time, with approximately 30 milliseconds being the longest with minimum system configuration.

Under a brownout condition(ac line between approximately 70 and 90 VAC) and depending on supply load the time could extend indefinitly. In this case PWRFAIL will assert low as long as ac is below the threshold value, while PWROK will assert high as long as the power supply can provide adequate dc voltage to the load.

ID-01067

Figure 6.8 Brief power failure not requiring battery backup

Power Indicator	Front Control Panel Power Indicator	Condition
OFF	ON	Normal operation (The power supply indicator will be on briefly when the power switch is first turned on.)
Blinking	OFF	Power supply failure or overloaded or shorted output(s)
ON when power switch first turned on, then goes OFF	OFF	Power supply failure, ac line voltage too high, or overvoltage on an output caused by lack of minimum load
ON	OFF	Insufficient ac line voltage or airflow, or ambient air temperature too high

Table 6.3 Fault indications

Interconnections

The compliant power supply board interconnects with the rest of the system through its A and B edge connectors to the chassis backpanel. Tables 6.2, 6.4, and 6.5 list each signal generated or received by the power supply board, together with the connector location(s) of the signal.

Signal	Connector Pin	Source	Destination	Description
+3V	B21	Power supply	Backpanel (J11)	+3V source (used for backpanel terminators)
+5VA	B31-38	Power supply	Backpanel (J11)	+5V source to slot 1
+5VB	B23-30	Power supply	Backpanel (J11) and Front control panel	+5V source to slots 2 and 3
+5VC	B39-46	Power supply	Backpanel (J11)	+5V source to slots 4 and 5
−5V	B97,98	Power supply	Backpanel (J11)	−5V source
+5MEM	B83-88	Power supply	Backpanel (J11) and Front control panel	+5MEM source
−5MEM	B99,100	Power supply	Backpanel (J11)	−5MEM source
+12V	B89-92	Power supply	Backpanel (J11)	+12V source
−12V	B19	Power supply	Backpanel (J11)	−12V source
+12MEM	B93,94	Power supply	Backpanel (J11)	+12MEM source
GND	B47-82	Power supply	Backpanel (J11)	Power ground
RETURN	B2-4, B7-8, B10, B12	Power supply	Backpanel (J11)	Logic ground for all system clock signals except line frequency clock (LCLK)
+VNR	A27-28	Power supply	Backpanel (J12)	+VNR source (Presently unused)
−VNR	A3i-32	Power supply	Backpanel (J12)	−VNR source (Presently unused)
+BAT	A49-52	Backpanel (J12)	Power supply	+BAT source (Presently unused)
PHASE	A1-4	Backpanel (J12)	Power supply	Switched ac hot line
NEUTRAL	A7-10	Backpanel (J12)	Power supply	ac neutral line
JUMPER1	A13-16	Backpanel (J12)	Power supply	ac line voltage selection jumper for the VNR source voltage circuit
JUMPER2	A19-20	Backpanel (J12)	Power supply	ac line voltage selection jumper for the auxiliary voltage transformer
JUMPER3	A23-24	Backpanel (J12)	Power supply	ac line voltage selection jumper for the auxiliary voltage transformer

Table 6.4 Voltage signals

Signal	Connector Pin	Source	Destination	Description
B40CLK	B5	Power supply	Backpanel (J11)	40 MHz square wave
B20CLK	B9	Power supply	Backpanel (J11)	20 MHz square wave
B10CLK	B1	Power supply	Backpanel (J11)	10 MHz square wave
ENB10CLK	B13	Power supply	Backpanel (J11)	10 MHz nonsymmetrical wave
5MHZCLK	B11	Power supply	Backpanel	5 MHz square wave
500HZ	B15	Power supply	Backpanel (J11)	500 Hz square wave
LCLK	B14	Power supply	Backpanel (J11)	50/60 Hz square wave (ac line frequency)

Table 6.5 Clock signals

Noncompliant 5-Slot Power Supply

The ECLIPSE S/120 5-slot power supply converts a 100/120 or 220/240 ac voltage power source to the five regulated dc voltages required by the ECLIPSE S/120 computers. It also generates a low-frequency (50/60 Hz) clock signal for use by the S/120 real-time clock. A battery backup feature, if configured, generates the regulated dc memory voltages from a +6 volt battery during an ac power failure. The power supply is shown in Figure 6.1.

The power supply, including the battery backup feature and its battery when configured, is contained on a single printed circuit board. This board supplies the regulated dc voltages and low-frequency clock to the printed circuit boards via the backpanel; it provides ac power for the fans via the internal cable. A line cord with a 12-pin connector applies line voltage to the power supply.

This chapter discusses the organization and operation of the 5-slot power supply, along with its interconnection with the rest of the system. Reference designators, such as C1 and T1, refer to the 5-slot power supply logic schematic, DGC No. 001-001616. Users may find it helpful to refer to the schematic while reading this chapter.

Functional Overview

The 5-slot power supply uses a forward mode off-line switching regulator to produce the high power outputs required by the ECLIPSE S/120 computers. The major components of the 5-slot power supply's offline switching regulator are a VNR source, a pulse width modulation control, an inverter, and a dc regulator.

In addition to the off-line switching regulator, the 5-slot power supply includes a line filter, an auxiliary voltage supply, soft start logic, several voltage regulators, various fault detection circuits, and the system clock circuitry. Figure 6.9 shows these components in relation to each other.

Theory of Operation

This section explains the operation of each 5-slot power supply component. Topics include the off-line switching regulator, dc regulator, auxiliary voltage supply, soft start logic, and fault detection circuits.

Off-line Switching Regulator

The ac line voltage passes through a line filter to the VNR source where it is converted into a high dc voltage source. Although this dc voltage source ranges from 259 to 337 VNR with the line voltage, it is referred to as **300 VNR**.

For 100/120 volt operation, the **300 VNR** source rectifies and doubles the ac line source by connecting the junction of C1 and C2 (J1-9) to the neutral side of the line (J1-8). In this way, a **300 VNR** ac source is provided. For 220/240 operation, the **300 VNR** source simply rectifies the ac line source.

The pulse width modulation control governs the operation of the inverter, and, in turn, of the dc regulator, by controlling the amount of power through the inverter. It opens and closes the power path once every 50 μs, at a 20 KHz rate, and varies the ratio between the *open* and *closed* times (the duty cycle) according to the variation on the **+5V** output. As the line voltage decreases or the **+5V** output load increases, the pulse width modulation control increases the *closed* time to transfer more power. In this way, the pulse width modulation control regulates the **+5V** output of the dc regulator.

Inverter

The inverter receives power from the **300 VNR** source and transforms it into the ac voltage outputs which power the dc regulator. The inverter's main component is transformer T1. T1 operates at 20 KHz with a variable duty cycle. Its maximum duty cycle is just under 50 percent.

DC Regulator

The dc regulator receives ac voltages from the inverter and converts them into five different low dc voltage outputs: **+5V**, **+15V**, **+12V**, **−5V**, and **−11V**. It consists of rectifiers, an inductor L1, various filter circuits, and two voltage regulators.

The **+5V** output feeds back to the pulse width modulation control to regulate the duty cycle of the inverter, which, in turn, regulates all outputs of the dc regulator.

The **+15V** output derives from T1 in the inverter and the **+5V** in the dc regulator. It has no additional regulation. Since the pulse width modulation control removes all line variations from **+5V**, line variations will not affect the **+15V** output. Although load variations on either the **+5V** or **+15V** outputs do have an effect on the **+15V** output, it is a relatively small one.

The **+15V** output passes through a simple series pass regulator to produce an extremely well regulated **+12V** supply.

The **−5V** and **−11V** outputs receive additional regulation. The **−5V** output is regulated by a **+5V** three terminal regulator with its positive output grounded. The **−11V** output is regulated by a **−12V** three-terminal regulator.

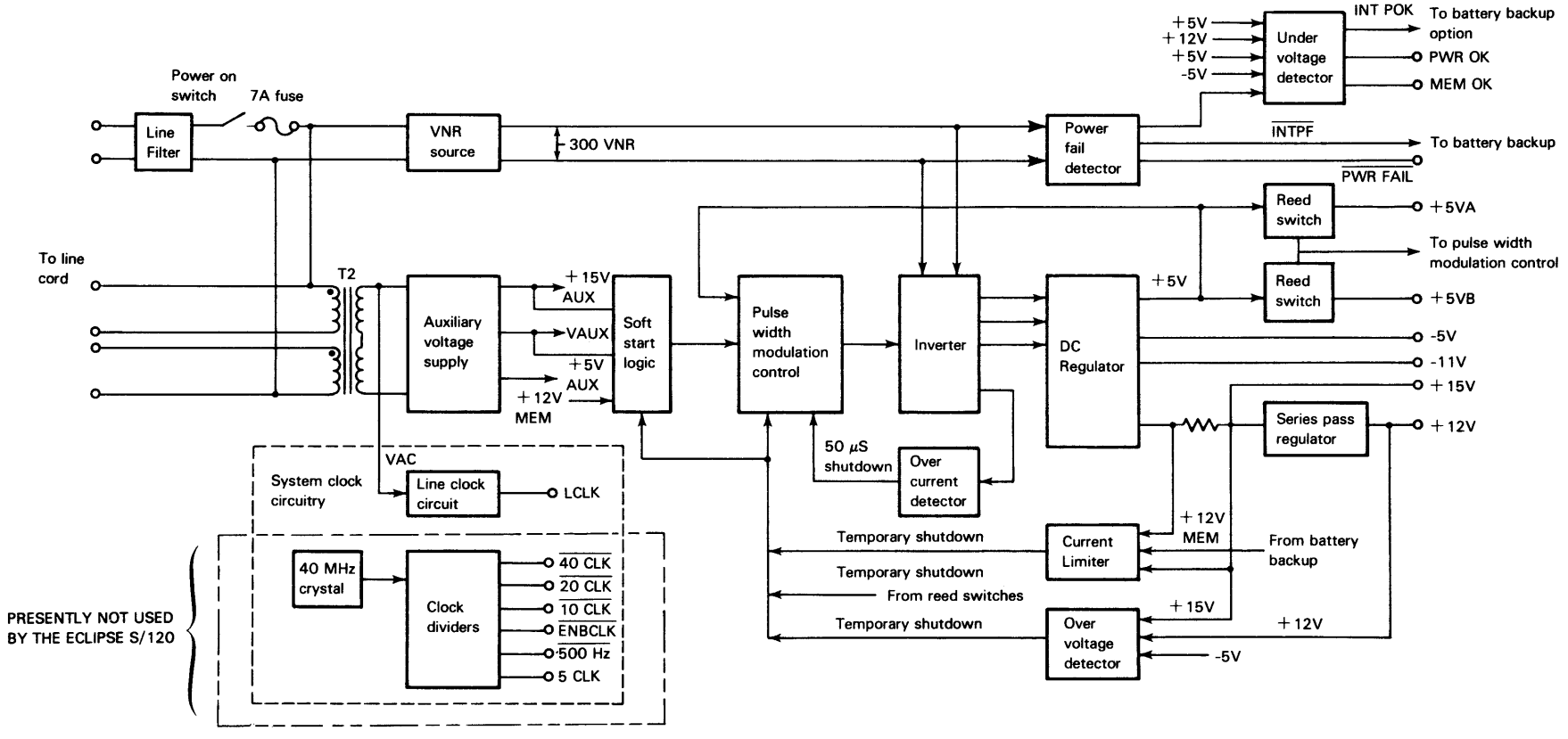


Figure 6.9 Basic 5-Slot chassis power supply

Auxiliary Voltage Supply

The ac line voltage powers the transformer T2 to produce an auxiliary voltage, which functions as a control voltage for the power supply. This voltage goes through a three-terminal voltage regulator to produce **+15V AUX** and **VAUX**. The auxiliary voltage supply also produces **+5V AUX**. Together, these voltages power most of the power supply circuits and also provide voltage references.

Soft Start Logic

The soft start logic enables the output voltages of the power supply to come up slowly, without overshooting when the system powers up. Its main component is a capacitor with a large time constant. This capacitor must charge before the pulse width modulation control can switch full power to the inverter. Most faults that shut down the pulse width modulation control also discharge this capacitor, allowing the output voltages to come up slowly when the fault is removed.

Fault Detection Circuits

Several circuits monitor the operation of the power supply, checking for under-voltage, over-voltage, and over-current conditions.

Under-voltage Detection

The *under-voltage detector* continually monitors the **+5V**, **−5V**, **+12V**, and **+15V** outputs. If any of these outputs falls below the minimum operating level, it drives the **INT POK** signal to a low state. The specifications for these and other power supply outputs are shown in Table 6.6.

	Voltage		Current	
Output	Min	Max	Min	Max
+5V	+4.95V	+5.2V	5A	35A
+12V	+11.7V	+12.7V	0	5A *
+15V	+14.0V	+16.0V	0	5A *
−5V	−4.75V	−5.25V	0	1.5A
−11V	−11.0V	−12.5V	0	0.025A
+5 MEM	+4.95V	+5.2V	0	1A
+12 MEM	+11.3V	+12.7V	0	3A *
−5 MEM	−4.75	−5.25V	0	0.05A

Table 6.6 5-slot chassis power supply specifications

**The sum of the maximum currents on +12V, +15V, and +12 MEM lines must not exceed 5 Amps. In battery backup mode, the maximum current specification for +12 MEM is 0.3 Amps*

The power fail detector monitors the **300 VNR** source for under-voltage conditions on the ac line source. When it detects an under-voltage condition, the power-fail detector pulls the **INTPF** to the low state. This signal drives the **PWR FAIL** signal to the low state to inform the system of an impending ac power failure. Approximately 22 ms later, **INT POK** is driven low and remains low for 155 ms or more, depending on the duration of the power failure.

Whenever **INT POK** goes to a low state, it also drives **PWR OK** low to reset the CPU. If battery backup is not present and the line fails, **PWR OK** remains low and later drives **MEM OK** low as well. If battery backup is present, **PWR OK** remains low and **MEM OK** remains high as long as the battery lasts.

Over-voltage Detection

The *over-voltage detector* protects against over-voltage conditions on the **+5V**, **+12V**, and **+15V** outputs. If any of these outputs exceeds the shutdown level, this detector shuts down pulse width modulation control permanently. The pulse width modulator control will not start again until the power supply is turned off and on again, using the front console power switch. This over-voltage fault also discharges the capacitor in the soft-start logic, allowing the voltage outputs to come up slowly when the system powers up again. Table 6.7 lists the levels at which an over-voltage fault and consequent shutdown occur.

Output	Shutdown Voltage
+5V	5.66 ± 1%
+12V	13.56 ± 5%
+15V	17.3 ± 5%

Table 6.7 Over voltage shutdown levels

Over-current Detection

A cycle-by-cycle current limiter performs the main functions of over-current protection. It consists of a current sense transformer T3 in the inverter and an *over-current detector*. The over-current detector monitors the current in the secondary winding of the T3 transformer. When the current flowing in this winding is too great, the over-current detector shuts down the pulse width modulation control for one cycle (50 μsec).

The *current limiter* is a secondary over-current protection device, which monitors the current on the **+12V**, **+15V**, and **+12V MEM** outputs. When the sum of the current on these three outputs exceeds 5 Amps, the current limiter temporarily shuts down the pulse width modulation control. Shutdown time varies with the load on these outputs, but it is typically 0.5 sec. This over-current fault also discharges the capacitor in the soft start logic, permitting the voltage outputs to come up slowly when the fault condition is removed.

Two reed switches with a one-turn winding around them function as current sensors for the **+5V** outputs. When the current on either of the two **+5V** outputs exceeds 20-25 Amps, the reed switches temporarily shut down the pulse width modulation control. This over-current fault also discharges the capacitor in the soft start logic, allowing the voltage outputs to come up slowly when the fault condition is removed.

Battery Backup

The battery backup option generates 6-volt dc power for the optional battery backup circuitry. This circuitry maintains the following critical memory voltages when a power failure occurs: **+5V MEM**, **+12V MEM**, and **−5V MEM**. These voltages are needed to refresh data stored in the dynamic RAM main memory. In addition, the battery backup circuity powers the system clock and the front console lights, and provides a memory status signal (**MEM OK**) to indicate that the memory voltages are above minimum operating levels.

In addition to the battery, the battery backup option consists of a battery switch, linear regulating circuit, pulse width modulation control, flyback converter, several protection circuits, and battery charger. Figure 6.10 shows these components in relation to each other.

Operation

The *battery switch* monitors the state of the signals **$\overline{\text{INT}}$ $\overline{\text{PF}}$**, **MEM OK**, and **INT POK** to determine when ac power fails and recovers. Power is failing if **$\overline{\text{INT}}$ $\overline{\text{PF}}$** goes to the low state when **MEM OK** is in the high state. In response, the battery switch turns on the battery to supply power (**BATT**) to the rest of the battery backup circuitry. When power recovers, **$\overline{\text{INT}}$ $\overline{\text{PF}}$** returns to the high state. Shortly thereafter, the voltage outputs from the basic power supply come up and drive **INT POK** high. If the battery is on when the outputs recover, **MEM OK** will be in the high state; if the battery is off, **MEM OK** will be driven high. When both **INT POK** and **MEM OK** are high, the battery switch turns off the battery.

The *linear regulating circuit* regulates the battery's output voltage to produce the **+5V MEM** output. This circuit is a series pass regulator with current limiting.

The *pulse width modulation control* governs the amount of power supplied to the flyback converter. It opens and closes the power path between the battery and the flyback converter once every 50 μs, at a rate of 20 KHz, and varies the ratio between the *open* and *closed* times (the duty cycle) according to the variation on the **+12V MEM** output of the flyback converter. In this way, the pulse width modulation control regulates the **+12V MEM** output.

The *flyback converter* transforms the 6V dc output from the battery into a regulated **+12V MEM** output and a **−5V MEM** output.

The *over-current detector* is a cycle-by-cycle current limiter. When it detects an over-current condition, it shuts down the pulse width modulation control for one cycle (50 μs).

The *discharge protection circuit* monitors the voltage level of the battery. When it drops below 5.4 volts, the discharge protection circuit turns off the battery to prevent it from degrading.

The *battery charger* recharges the battery when basic power supply is operating.

Interconnection with the System

The 5-slot power supply board communicates with the rest of the system via the jack J1 to the backpanel. Tables 6.8 and 6.9 list each signal generated or received by the power supply board together with the jack locations of the signal, except for the clock signal, **LCLK**. This signal is generated from the power supply and sent to the backpanel. It is a 50/60 Hz square wave located on pin J1-17. Refer to the 5-slot backpanel schematic, DGC No. 001-0001619, for the locations of signals on the backpanel.

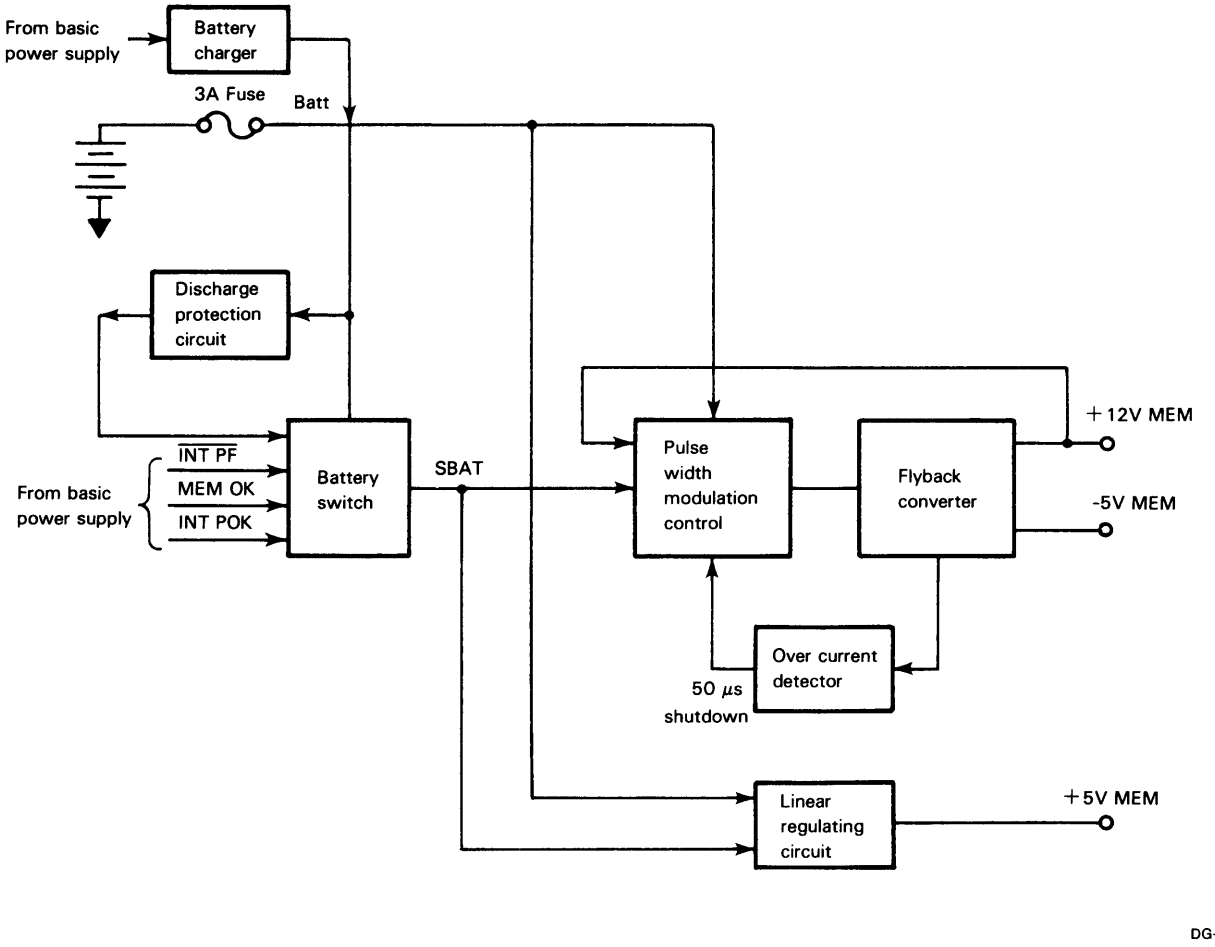


Figure 6.10 Battery backup

Signal	Jack Pin	Source	Destination	Description
GND	J1 pins 1, 2, 4, 8, 35-42, 51, 52, 55-64, 79-88	Power supply	Backpanel	Power or logic ground
+3V	J1-31, J1-32	Power supply	Backpanel	+3V source for backpanel bus terminators
+5VA	J1 pins 91-100	Power supply	Backpanel	+5V source
+5VB	J1 pins 67-76	Power supply	Backpanel	+5V source
+5 MEM	J1-33, J1-34	Power supply	Backpanel	Same as +5V if battery backup not configured
−5V	J1-27 J1-28	Power supply	Backpanel	−5V source
−5 MEM	J1-19	Power supply	Backpanel	Same as −5V if battery backup not configured
−11V	J1-24	Power supply	Backpanel	EIA interface voltage
+12V	J1 pins 43-46	Power supply	Backpanel	+12V source
+12 MEM	J1 pins 47-50	Power supply	Backpanel	Same as +12V if battery backup not configured
+15V	J1-53, J1-54	Power supply	Backpanel	+15V source

Table 6.8 Voltage signals

Signal	Jack Pin	Source	Destination	Description
MEMD	J1-21	Backpanel	Power supply	Unused
MEM OK	J1-23	Power supply	Backpanel	+5 MEM voltage ok
ONLED-P	J1-11	Power supply	Backpanel	All dc output voltages ok
PWR FAIL	J1-9	Power supply	Backpanel	Power switch off or ac power failure
PWR OK	J1-16	Power supply	Backpanel	All dc output voltages ok

Table 6.9 Power supply status signals

Chapter 7

Virtual Console

The virtual console is a program that allows you to interact with the ECLIPSE S/120 computer system via your terminal. Simple commands entered on a terminal keyboard permit you to examine and/or modify any processor register or memory location. A *breakpoint* feature lets you stop program execution at selected places for debugging.

> **NOTE:** *The virtual console inhibits input/output (I/O) interrupts. Also, when the virtual console is executing, the RUN light on the chassis is not lighted. When a user program is executing, the RUN light is lighted. I/O protection is not enabled when the virtual console is executing.*

The virtual console resides in read-only memory (ROM) chips on the ECLIPSE S/120 system processing unit (SPU) board. The virtual console has access to 512 bytes of static read/write memory (RAM), also on the SPU board, which it uses as a *scratchpad*. Since neither virtual console ROM nor scratchpad RAM is part of the normal address space, they are transparent to the user.

Entering the Virtual Console

Upon power up, the virtual console firmware first performs a short (0.75 second) self-test routine; then, if the front panel is locked, it examines the contents of the automatic program load (APL) register. If the APL register contains device code 0, the virtual console retains control. If it contains any other device code, the central processing unit (CPU) performs an immediate program load from that device. (See "Installation and Jumpering" in Chapter 1, and "Program Load Commands" and the Control-G command, below.) If the front panel is unlocked, virtual console retains control.

In addition to power up, the virtual console is entered when

- A **HALT** instruction is executed unless Halt Dispatch is disabled. (Refer to SIO jumpers in the "Installation and Jumpering" section of Chapter 1).
- The user presses the BREAK key of the system console, *if* the Break function on the SPU board has not been disabled by jumpering).

> **NOTE:** *When virtual console is entered as a result of depressing the BREAK key, virtual console tests the front console LOCK switch. If the LOCK switch is in the LOCK position, virtual console immediately returns to the user's program.*

- The user presses the RESET switch on the front console and the console is not locked.
- The program completes execution of an instruction or a number of instructions in the one-step mode. (See "Single Stepping," below.)
- A breakpoint is encountered. (See "Breakpoints" below.)

> **NOTE:** *When the user presses the RESET switch to enter virtual console the contents of the program counter are lost.*

The contents of the ACs, the program counter and the carry bit are displayed each time virtual console is entered, except on power up and when the user presses the RESET switch.

Once the virtual console has been entered and the user presses the PR LOAD switch on the front panel, the virtual console attempts to program load from the device selected by the APL register.

Once called, the virtual console displays a ! on the terminal. This is the virtual console *prompt*; it indicates that the virtual console is ready to accept a command. A single character preceding the prompt indicates the current state of the memory allocation and protection (MAP) unit:

!	The MAP is off; no address translation will occur.
A!	The MAP is on; user A has been selected.
B!	The MAP is on; user B has been selected.
C!	The MAP is on; user C has been selected.
D!	The MAP is on; user D has been selected.

When a particular user has been selected, the current state of that user's map is used in all address translations. You can change the MAP from within the virtual console, as explained in "Changing the MAP or MAP Status" later in this chapter.

Correcting Entry Errors

This section discusses two methods for correcting typographical errors: the Rubout/Delete key and the **K** command. The final topic in the section is virtual console errors, which occur when commands are issued incorrectly. The context of these errors will be clarified when each command is discussed later in the chapter.

Rubout/Delete

The Rubout/Delete key deletes the last character you typed. The virtual console echoes the deletion with an underscore (_). Typing additional Rubouts deletes digits from right to left.

If you type any Rubouts immediately after opening a cell, the virtual console deletes the rightmost digits of the cell's contents as though you had just typed them yourself. You may then type in new values for these digits. Refer to the "Commands" section in this chapter for more information on the Rubout key.

> **CAUTION:** *The Rubout/Delete key has no effect on floating-point accumulators. If you type these characters, virtual console issues a new line prompt without changing data.*

The K Command

To cancel an entire line, type **K**. In response, the virtual console prints a ? followed by a New Line with a prompt and also closes the current cell if it is open. The ? followed by a New Line with a prompt is also printed if you type a character that the virtual console does not recognize.

Virtual Console Errors

If you attempt to open a nonexistent memory cell, the data displayed as its *contents* will be meaningless. To determine whether a location exists, enter a new value in the memory cell and then reopen it. If it does not contain the value just entered, then the location is nonexistent.

The virtual console types a ? followed by a New Line with a prompt under the following conditions.

- An undefined character is typed.
- A command to open a nonexistent internal cell is issued.
- An **R** command is issued without an argument.
- A set breakpoint command specifies an invalid address.
- A delete breakpoint command specifies a number greater than 7.
- A set breakpoint command is issued after all eight breakpoints have been assigned.
- An *n*M or *n*C command is issued and no MAP is on, or *n* specifies a logical page number greater than 37_8.
- An *n*F command is issued and *n* specifies a number greater than 3.

In all of these cases, the command involved will not be executed. In fact, the virtual console does nothing, except discard data that was entered with an erroneous command.

Commands

In the following sections, virtual console commands are categorized into three groups:

- Cell commands
- Function commands
- Miscellaneous commands

A virtual console command consists of a single character. Some commands require a leading *argument*, which is an octal number or an expression. Valid numbers and expressions are

Digits	Ranging from 0 through 7. If the argument is an address, it must range from 0 through 77777.)
Hex digits	Ranging from 0 through 9 or A through F.
Period (.)	Replaces the value of the address last used.
+ or −	+ or − sign may be entered between valid numbers. The virtual console program computes the arithmetic result and replaces the original expression with it.
Delete or Rubout	Deletes any single digit. The virtual console displays an underscore character (_), indicating that the character preceding it has been deleted. The delete key has no effect on the + or − symbols. Used after a period, the delete key deletes the rightmost digit of the last address. Since the virtual console only retains six digits at any time, the delete key will not resolve all errors.

For clarity, all examples in this chapter show data entered by the user in bold type. On the terminal, user input and program response are not differentiated.

Cell Commands

Several virtual console instructions operate on *cells*. A cell is either a memory location (*memory cell*), an internal register (*internal cell*) such as an accumulator, or a floating-point accumulator (*FPAC cell*). Each internal register accessible by the virtual console is assigned an internal cell number. Table 7.1 lists these registers and their numbers.

Number	Cell
0-3	Accumulators AC0 through AC3, respectively.
4	The address of the break instruction at which the program halted, if the virtual console was entered on encountering a break instruction; or the contents of the program counter, if the virtual console was entered in any other way.[1]
5	Carry bit: bit 15 is equal to 0 when carry equals 0 and equal to 1 when carry equals 1.
6	CPU (interrupt and NMI) status.[2]
7	System console status word.[3]
10	Virtual console register.[4]
11	MAP status register.[3]
12	Floating-point status register.[3]
13	Floating-point program counter.[3]
14	Search mask.[5]

Table 7.1 Internal cells

[1]*The virtual console sets bit 0 of this word to 1 when it takes control, regardless of its original value. Bit 0 in cell 4 must be 1 when the user program begins again after a* **P** *command.*

[2]*Refer to the* ECLIPSE S/120 Assembly Language Programmer's Reference Manual *(DGC No. 014-000682), or to Figure 1.2 or Table 2.1 in this manual, for the contents of these registers.*

[3]*Refer to the programmer's reference listed above, or to Figure 1.2 in this manual, for the contents of these registers.*

[4]*Refer to "Program Load Commands" in this chapter for the contents of this register.*

[5]*Refer to "Search Command" in this chapter for the contents of this register.*

To examine or modify any cell, you must *open* it using one of the commands listed in Table 7.2. Opening a cell causes its address and contents to be displayed on your console. Addresses and memory or internal cell contents are displayed in octal format, while floating-point cell contents are displayed in hexadecimal format.

Internal cell number 10, the virtual console register (Table 7.1), can be accessed while in user mode with a **READS** instruction. This call always contains the device code of the last device from which a program load was performed.

Command	Function
*n*A	Opens the internal cell specified by *n*.
*n*F	Opens the FPAC cell specified by *n*.
expr/[1]	Opens the memory location specified by octal number *expr*.
Carriage Return	Closes the current cell[2] and opens the next consecutive cell.
New Line[3]	Closes the current cell[2] but does not open another.
/	Closes the current cell[2] and opens the memory cell whose address is equal to the contents of the current memory or internal cell; after a New Line, opens location 0.

Table 7.2 Virtual console cell commands

[1]*The symbol expr represents any valid octal number or expression, as described in the preceding section "Commands."*

[2]*Current cell means the last cell that you opened.*

[3]*Line Feed on non-ANSI standard keyboards.*

When you open a memory cell, the virtual console interprets the address according to the current setting of the user MAP. That is, the number you enter is interpreted as a 15-bit address and then translated into a physical address. If the memory cell address you enter contains more than five octal digits (15-bits), only the last five digits are used. Leading zeroes are not necessary. If, for example, you want to open logical memory location 5, type

```
5/
```

Once you have opened a cell, you may change its contents. Memory and internal cells are altered by entering octal digits; leading zeroes are not necessary. Floating-point accumulators are altered by entering hexadecimal digits, beginning with the most-significant-position. Therefore, it is not necessary to supply trailing zeroes. Terminate the expression with a Carriage Return, Line Feed, or New Line. Note that if you type Carriage Return, it also opens the next cell. This keystroke command is convenient for entering data into several consecutive locations.

NOTE: *If you open a cell and immediately type* **H** *or* **L**, *the contents of the cell are used as the value of* expr *for that command.*

If you type an expression starting with a + or −, the value of the expression will be added to, or subtracted from, the current contents of the cell.

NOTE: *You cannot type an expression starting with + or − when altering FPAC cells.*

Examples of expression resolution, when the last address entered was 100, follow.

100000_1 will be replaced by 100001.

1000000 will be replaced by 000000 *(Virtual console only remembers six digits).*

.-3 will be replaced by 75.

.7 will be replaced by 1007.

0-7 will be replaced by 177771.

6+.-3 will be replaced by 103 *(6+100-3=103).*

75+_5 will be replaced by 102, *(75+5=102* — the + is not deleted.).

60+._ will be replaced by 70 *(60+10=70* — the _ erased the rightmost 0 of the last address, 100).

NOTE: *When altering FPAC cells, you cannot type an expression starting with + or −, or delete any digits already entered.*

If you enter an illegal digit, virtual console issues a prompt and does not change the cell content. If you enter more

than 16 bits when altering internal cells or memory cells, only the last 16 bits entered are used. If you enter 16 digits when altering floating-point cells, virtual console automatically changes the cell content and issues a prompt.

Examples showing the use of /, New Line, and Carriage Return follow.

A! **3A** 000003A 000100 <**CR**>
AC3 contains 100. Internal cell 4 is opened as shown on next line.

000004A 000704 /0000704 024132 <**NL**>
PC contained 704. Memory location 704 contains 24132. The next memory location is not opened.

A! **5A** 000005A 000000 **1** <**NL**>
User changed the carry bit to 1. The next cell is not opened.

A! **100**/000100 025037 .<**NL**>
Changes the contents of memory location 100 to current address. The next memory location is not opened.

A! **100**/000100 000100 <**CR**>
Confirms the preceding commands. Memory location 101 is opened as shown on next line.

000101 000602 +**1**<**NL**>
Increments contents of 101. The next memory location is not opened.

A! ./000101 000603
Confirms preceding step.

In all memory location examples above, there was no map protection for the logical page. When the logical page is protected, the type of protection is indicated with a character displayed between the memory address and the data as follows.

A! **50**/000050 X DDDDDD

In this example, X represents the type of protection and may be V, W, WP, or a *space*. The meanings of these characters are explained below. DDDDDD Represents data.

V	Indicates that logical page is validity protected and data displayed is invalid.
W	Indicates that logical page is write protected and data cannot be changed.
WP	Indicates that logical page is write protected, user write fault is enabled, and data cannot be changed.
space	Indicates logical page is not protected.

Function Commands

Table 7.3 lists the virtual console function commands. Sections that follow explain these commands in detail.

Command	Function
*expr***A**	Displays contents of the internal cell specified by *expr*. (If no *expr* is entered, AC0 through AC3, program counter, and carry bit are displayed.)
*expr***B**	Inserts a breakpoint at the memory location specified by octal number *expr*. (If no *expr* is entered, all breakpoints will be displayed along with their assigned numbers.)
C	Displays contents of all data channel maps.
*n***C**	Displays and/or alters the currently-selected data channel map page specified by $n(n = 0 - 37_8)$.
*n***D**	Deletes breakpoint number *n* where *n* is a number between 0 and 7. (If no *n* is specified, all breakpoints are deleted.)
↑**G**	Executes the power-up self-test sequence.*
*n***H**	Performs a program load from the data channel device whose device code is *n*.
I	Executes an *I/O Reset* (**IORST**) instruction. Resets all devices.
K	Cancels the entire line just typed and prints a question mark (?).
*n***L**	Performs a program load from the programmed I/O device whose device code is *n*.
M	Displays contents of all user maps.
*n***M**	Displays and/or alters the currently-selected user map page specified by $n(n = 0 - 37_8)$.
*n***O**	Steps through *n* instructions of the user's program. (If no *n* is specified, one instruction is executed.)
P	Starts progrm execution at the memory location specified by the contents of internal cell number 4. (See Table 7.1.)
*n***P**	Same as **P** above, except the next *n* breakpoints are ignored.
*expr***R**	Starts program execution at the memory location specified by octal number *expr*.
*n***S**	Searches the currently-selected logical space for the value specified by *n*.
U	Changes the user map. Displays a colon (:), after which the user must enter A, B, C, or D to specify a map. If any other character is entered, the map is turned off.
↑**V**	Performs a user read/write memory test at all locations.**

Table 7.3 Virtual console function commands

**↑G represents the simultaneous depression of the console CTRL and the G keys.*

***↑V represents the simultaneous depression of the console CTRL and the V keys.*

Breakpoints and Program Control

The virtual console breakpoint facility allows you to place breakpoints at up to eight locations in your program. When the program encounters the breakpoint during execution, it enters the virtual console so that you can examine or modify cells. Breakpoints are an aid in debugging programs, since they enable you to stop your program at locations where there may be problems and then to resume execution without losing data.

NOTE: *Breakpoints will not work and should not be used if the Halt Dispatch is not enabled. Refer to Halt Dispatch jumper in the "Installation and Jumpering" section of Chapter 1.*

Setting Breakpoints. To set a breakpoint, type *expr***B**. This command sets the breakpoint at the address specified by argument *expr*, according to the *current user map*. Breakpoints can be used only in the user map from which the user program will be started.

The virtual console assigns numbers to breakpoints in reverse order — that is, breakpoint 7 is assigned first, then breakpoint 6, and so on. The unassigned breakpoint with the highest number is always assigned first. For example, if numbers 7 and 5 are assigned, the next breakpoint will be 6, not 4.

To delete a breakpoint you must use the number assigned to it as described below. Typing **B** with no specified address directs the virtual console to list all the current breakpoints and their assigned numbers.

Examples

A!**423B**
Places a breakpoint at address 423 in user map A.

A!**B**
Requests list of all current breakpoints.

7 75324
Breakpoint 7 is at address 75324.

3 423
Breakpoint 3 is at address 423.

A!**623B**?
Requests a breakpoint at a valid location when all eight breakpoints are in use. User must delete a breakpoint before setting another.

A!
A prompt always follows a "?".

NOTES: *Do not place two breakpoints at the same location.*

Breakpoints **must never** *be set in addresses that are I/O protected or to be executed when the* Load Effective Address *mode is enabled.*

Deleting Breakpoints. The **D** command deletes a breakpoint. The command's format is

*n***D**

where *n* specifies the break point number. This command deletes the breakpoint, regardless of the current state of the map. If no argument (*n*) is specified, **D** deletes all breakpoints.

Examples

A!**3D**
Deletes breakpoint number 3.

A!**12D**?
Only eight breakpoints (numbers 0-7) are valid; — no other number is allowed.

Encountering a Breakpoint. When a breakpoint is encountered during execution of a user program, the virtual console is entered. The address of the instruction at which the breakpoint was set is displayed and placed in internal cell number 4, and the instruction at the location of the breakpoint is not executed. Then the virtual console displays a prompt, indicating that the user can now inspect and modify any internal cell or memory location.

Single Stepping

Use the command **O** to one-step through a program. Issued when the virtual console has control, the **O** command sets a flag that causes a nonmaskable interrupt (NMI) to occur immediately after the first main (user) program instruction has executed. The virtual console then returns to the main program location specified by the contents of internal cell 4; the main program executes one instruction; and then the virtual console resumes control.

You can cause the virtual console to step through *n* instructions by typing *n***O**, where *n* is an octal number ranging between 0 and 177777. The virtual console then executes those *n* instructions. As each instruction is executed, the virtual console prints the address of the following instruction, the contents of the ACs, and the condition of the carry bit, that is, the contents of the program counter, ACs, and the carry flip-flop at the time of instruction execution. This is a convenient way to locate skips or branches. After the *n* count of instructions have been executed, the virtual console resumes control and issues a prompt.

NOTES: *The fact that a user breakpoint may have been set for an instruction has no effect on the execution of the* **O** *command. When single stepping, interrupts will be honored, if the interrupt system was enabled before entering virtual console.*

The BREAK key interrupts the virtual console before it finishes stepping through all n *instructions. In response, the virtual console displays the address of the instruction following the one last executed and then displays a prompt.*

Resuming Program Execution

Two virtual console commands allow you to resume program execution after the virtual console has been entered through a breakpoint, after single-stepping, or after the BREAK key has been pressed: **P** and **R**.

Typing **P** restarts program execution at the location specified by the contents of internal cell number 4, which specifies the return address. (See Table 7.1). Typing *n***P** restarts program execution in the same manner; however, the next *n* breakpoints are ignored.

NOTE: *When the virtual console is entered through a breakpoint, internal cell 4 contains the address of the location of the breakpoint. This should be the next instruction to be executed to resume normal program flow. When the virtual console is entered any other way, internal cell 4 contains the value of the PC + 1; this should also be the next instruction executed in order to resume normal flow. In either case, the* **P** *instruction produces the required result.*

You can also return to a program by typing *expr***R**. In this case, program execution resumes at the location specified by *expr*. When the **R** command is issued, the virtual console inserts all previously specified breakpoints, clears nonmaskable interrupts (NMIs), and resumes program execution at the logical address <*expr*> in the currently selected MAP. The **R** command does *not* cause an I/O or system reset. The number specified by *expr* must be a valid address in user memory. If this argument is not in the user range, or is not supplied, the virtual console simply displays **?** and then issues a prompt.

Miscellaneous Commands

Changing the MAP or MAP Status

To change user maps, use the **U** command. In response to this command, the console immediately prints a ":". To change the user map, type a single character: A, B, C, or D. Typing *any* other character (including Carriage Return and New Line), turns off MAP. After you type a valid character, the virtual console displays a prompt reflecting the new state of the MAP.

You can change the MAP status from within the virtual console by opening internal cell 11, the MAP status register, with a **11A** command. Type in the new status and then close the cell with a New Line. *Now issue a* **U** *command, whether or not you want to change maps.* (If you do not want to change maps, simply enter the character for the current map.) The new map status is effective only after a **U** command has been issued.

Examples

!U:B
No user map selected; user selected the B user map.

B!**11A**000011A 000000 **77**<**NL**>
MAP status register cleared; user entered 77.

B!**11A**000011A 000077 <**NL**>
Confirms preceding step

B!U:B
New map status takes effect.

B!U:<**NL**>
!
Turns MAP off

Displaying, Altering, and Dumping User MAPs

Using the **M** command displays and/or alters the physical page to which a logical page of the currently selected user map is mapped. This command also displays the physical page to which all logical pages of all user maps are mapped.

To display the physical page to which a logical page of the currently-selected user map is mapped, type *n***M**. In this format, *n* represents the octal logical page number. The physical page is displayed in the following format:

WPPPPP

W Signifies write protection: 1 = write protected, 0 = not write protected.

PPPPP represents the physical page number in octal (0 - 1777).

After the physical page number is displayed, you can alter the physical page for the same logical page by typing the new page number in the same format followed by New Line or Carriage Return.

To display the physical page to which all logical pages of all user maps are mapped, type **M**.

Examples

A!**32M** 000132 **100120**<**NL**>
A!

User map A, logical page 26 ($32_8 = 26_{10}$), is mapped to physical page 90 ($132_8 = 90_{10}$) and is not write protected. The user remapped logical page 26 to physical page 80 ($120_8 = 80_{10}$)and set write protection.

A!**5M**100030 <**CR**>
000006 000077 <**NL**>
A!

User map A, logical page 5, is mapped to physical page 24 ($30_8 = 24_{10}$) and is write protected). User map A, logical page 6, is mapped to physical page 63 ($77_8 = 63_{10}$) and is not write protected.

Displaying, Altering, and Dumping Data Channel MAPs

The **C** command displays and/or alters the physical page to which a logical page of the currently selected data channel map is mapped. It also displays the physical page to which all logical pages of all data channel maps are mapped.

To display the physical page to which a logical page of the currently-selected data channel map is mapped, type *n***C**. The display of the physical page is in the format described above for user maps.

To display the physical page to which all logical pages of all data channel maps are mapped, type **C**.

Program Load

There are three methods of performing a program load while in the virtual console mode: two virtual console commands, **nL** and **nH**; and pressing the **PR LOAD** switch on the front console. The **PR LOAD** was discussed earlier in this chapter. *n*L and *n*H are discussed here.

> **NOTE:** *The S/120 virtual console executes program loads without storing a bootstrap loader in lower user memory.*

Typing *n*L causes the CPU to perform a program load from a programmed I/O device whose device code equals the octal number *n*. Device code 0 is normally not a valid device code, device code 77 is reserved for the CPU, and device codes 1, 2, 3, 10, 11, 14, and 43 are reserved for S/120 SPU resident devices.

Typing *n*H causes the CPU to perform a program load from a data channel device whose device code is equal to *n*. The value of *n* must be within the range specified for the *n*L command.

Once a program load has begun, the virtual console is no longer in control.

After a program load has been performed -- whether initiated by *n*L, *n*H, or **PR LOAD** -- the device code of the device loaded from is placed in the virtual console register. The device code, shifted left one bit position, is also placed in AC0. A **READS** instruction can be used at any time to retrive the device code of the last load device from the virtual console register.

I/O Reset

The **I** command causes the virtual console to immediately execute an *I/O Reset* instruction to all I/O devices. This clears all I/O device controller flags (Busy = 0, Done = 0). Refer to the ECLIPSE *S/120 Assembly Language Programmer's Reference* for information on the effects of the *I/O Reset* instruction.

Example

B!**I**
!

User map B was selected when the user issued an **I** command. **I/O RESET** disables the MAP.

Search

The *expr***S** command is a means of searching the currently-specified logical space (64-kbytes) for the value *expr*. The contents of each searched memory location are "ANDed" with a search mask contained in internal cell 14 before the comparison is made. The address and contents of each location where a comparison is found are displayed as

AAAAAA DDDDDD

In this format,

AAAAAA Is the physical memory address
DDDDDD Is the contents

Example

This example searches the currently-specified logical page for all occurrences of any I/O instruction to device code 22.

A!**14A** 000014A ?????? **160077**<**CR**>

Sets the search mask to 160077, thus limiting the comparison to all I/O instructions. (??????? signify the contents of internal cell 14 when it was opened.)

A!**060022S**

Searches for all I/O instructions to device code 22.

15643 60122
17423 61322
23654 62222

I/O instructions to device code 22 were found in three locations of the current logical space.

Test Commands

Power-up Self Test. The ↑**G** command causes the virtual console to execute the power-up self-test sequence.

> **CAUTION:** *The memory-test portion of the power-up self-test is destructive to user and virtual console memory; therefore, a program load* **must** *be performed after the self-test.*

Once the powerup self-test begins, it will attempt to run to completion even if errors are detected. If an error is found in a section of the test, an appropriate error flag is displayed and the next test section is started.

Errors that occur are flagged as follows.

I/O fault: the virtual console displays an I.

User memory fault: the virtual console displays an M.

Error checking and correction fault: the virtual console displays an E.

Virtual console memory fault: the virtual console displays an H.

If any error is encountered, virtual console stops, and the user must depress the BREAK key to continue.

> **NOTE:** *Pressing the BREAK key after an error code(s) display will allow the virtual console to continue but the error condition remains and may cause unpredictable results. The indicated error(s) should be corrected before continuing.*

If no errors are encountered and the front panel is locked, the virtual console performs a program load as described in Chapter 1. The virtual console will retain control and issue a prompt under the following conditions:

- The user depresses the BREAK key after an error.
- The APL register contains device code 0.
- The front panel is unlocked.

> **NOTE:** *The power-up self-test is* not *a definitive hardware test. Its primary purpose is to detect faults that would prevent the loading of diagnostic programs.*

User Read/Write Memory Test. The ↑**V** command directs the virtual console to execute a test sequence of all user read/write memory locations.

> **CAUTION:** *The test is destructive to user read/write memory; therefore, a program load* **must** *be performed after the* **V** *command.*

The virtual console displays **P** each time all locations of user memory have been tested. Any error that occurs causes the error location to be displayed as shown below, and the test continues.

XXX YYYY

In this display

XXX	Indicates the physical page number that failed. The number ranges from 0 to 377_8
YYYY	Specifies the location within the page that failed. The number ranges from 0 to 1777_8

To stop the user memory test, depress the BREAK key. The test will stop at the completion of the current pass.

Appendix A

Summary of NOVA/ECLIPSE I/O Bus Signals

This appendix presents a summary of the NOVA/ECLIPSE input/output bus as it pertains to the S/120 computer system. The bus structure embodies a single bus connecting the central processor to all interfaces as shown in Figure A.1. Data is transferred on the bus along 16 parallel, bidirectional, data lines. Control signals are carried along dedicated, unidirectional, control lines. In addition to specifying a unique function, each control signal generated by the system processor provides all timing necessary to perform that function. Data transfers are synchronous; no *hand-shaking* occurs between the interface and the system processor. The data channel and program interrupt facilities each use their own single request and priority lines. The two request lines run parallel to all interfaces, so that an interface requiring either data channel or program interrupt service need only assert the appropriate line and wait for the processor to respond. The serial priority lines are independent and are chained from interface to interface, so that priority for service is granted to the interface closest on the chain to the central processor.

Although the interrupt priority chain begins at the interface on the interrupt priority chain closest to the S/120 SPU, the command to acknowledge an interrupt *INTA* begins on the SPU board. If any one of the CPU support devices contained within the SIO controller chip or the error checking and correction unit has an allowable interrupt pending at the time the **INTA** command is executed, the interrupt acknowledge signal **INTA** will not be applied to the external I/O bus. Interrupt priority within the SIO chip is as follows:

Power fail
Programmed Interval Timer
Real Time Clock
Asynchronous interface receiver (TTI)
Asynchronous interface transmitter (TTO)

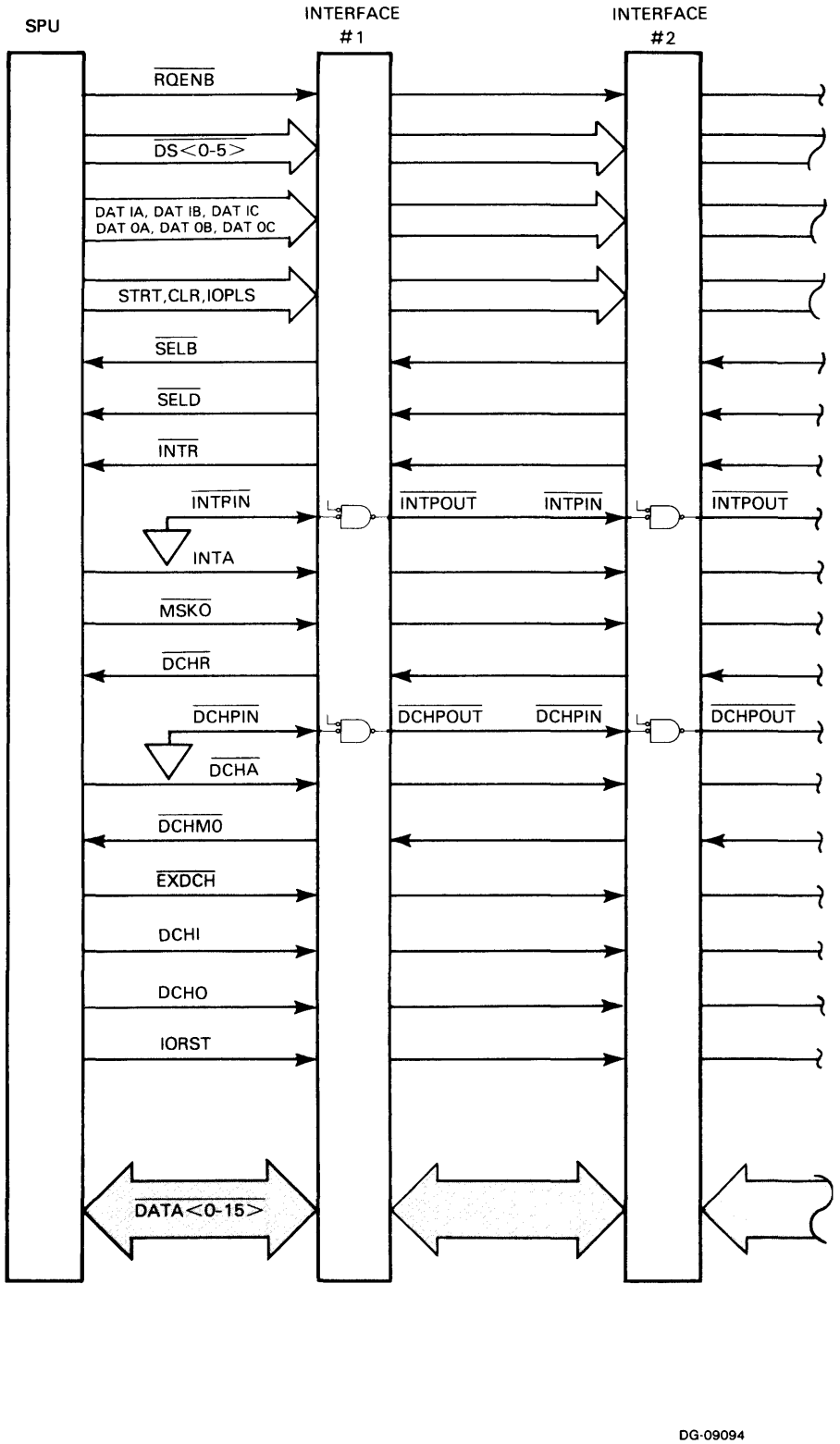
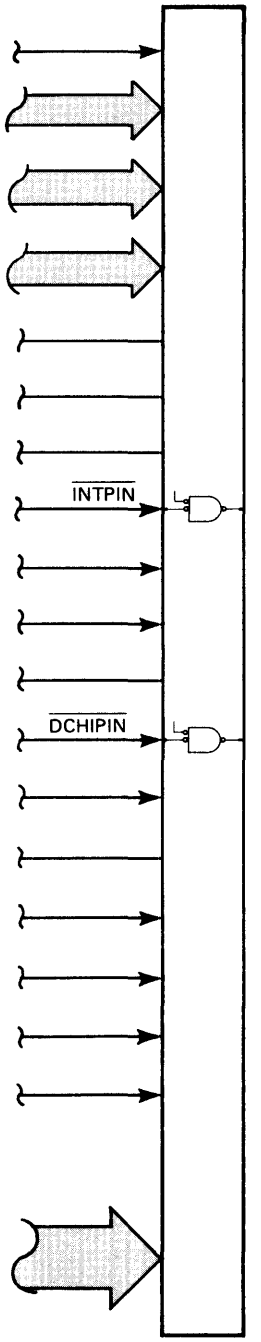
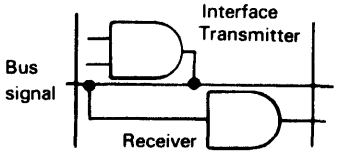


Figure A.1 The NOVA/ECLIPSE I/O bus



GROUP	SIGNALS	DESCRIPTION
ENABLE	RQENB	Enables interrupt request and data channel requests
PROGRAMMED I/O	DS<0-5>	Programmed I/O device select
	DATIA, DATIB, DATIC DATOA, DATOB, DATOC	Data input or data output command strobe
	STRT, CLR, IOPLS	Flag control or flag test command
	SELB	Selected Busy flag
	SELD	Selected Done flag
PROGRAM INTERRUPT	INTR	Interrupt request
	INTPIN/INTPOUT	Serial interrupt priority chain
	INTA	Interrupt acknowledge
	MSKO	Interrupt mask out strobe
DATA CHANNEL	DCHR	Data channel request
	DCHPIN/DCHPOUT	Serial data channel priority chain
	DCHA	Data channel acknowledge
	DCHM0	Data channel mode select
	EXDCH	Data channel map select
	DCHI	Data channel input strobe
	DCHO	Data channel output strobe
SYSTEM CONTROL	IORST	I/O reset command
DATA & ADDRESS	DATA<0-15>	Transfers: Programmed I/O data Interrupt acknowledge device code Data channel address Data channel data

NOTE: All interface to bus connections are parallel connections (see diagram) unless shown otherwise.



Summary of I/O Bus Signals

The forty-eight signals comprising the I/O bus can be divided functionally into five groups:

- Data
- Programmed I/O
- Program Interrupt
- Data Channel
- System Control

The following list shows this grouping and briefly describes the function of each signal. Timing information for these signals is shown in *Interface Designer's manual for* NOVA *and* ECLIPSE *Line Computers*, DGC Ordering No. 014-000629. Note that, with the exception of the two priority lines, all I/O bus lines run parallel to all interfaces.

Data

$\overline{\text{DATA}}$<0-15> *Data.* All data and addresses, for both data channel and programmed I/O, are transferred between the processor and interfaces attached to the I/O bus via these 16 bidirectional lines. The interrupt disable mask and interrupt acknowledge information are also carried on these lines.

Programmed I/O

$\overline{\text{DS}}$<0-5> *Device Select.* These lines carry the low-order six bits of an I/O instruction; that is, the device code. Only the interface whose device code corresponds to that carried on these lines should respond to control signals generated on the I/O bus.

DATIA *Data In A.* Asserted by the processor during the execution of a **DIA** instruction. Should cause the interface selected by $\overline{\text{DS}}$<0-5> to place the contents of its A input buffer on $\overline{\text{DATA}}$<0-15>.

DATIB *Data In B* Asserted by the processor during the execution of a **DIB** instruction. Should cause the interface selected by $\overline{\text{DS}}$<0-5> to place the contents of its B input buffer on $\overline{\text{DATA}}$<0-15>.

DATIC *Data In C.* Asserted by the processor during the execution of a **DIC** instruction. Should cause the interface selected by $\overline{\text{DS}}$<0-5> to place the contents of its C input buffer on $\overline{\text{DATA}}$<0-15>.

DATOA *Data Out A.* Asserted by the processor during the execution of a **DOA** instruction, after the processor has placed the contents of the specified accumulator on $\overline{\text{DATA}}$<0-15>. Should cause the interface selected by $\overline{\text{DS}}$<0-5> to load its A output buffer with the data on $\overline{\text{DATA}}$<0-15>.

DATOB *Data Out B.* Asserted by the processor during the execution of a **DOB** instruction, after the processor has placed the contents of the specified accumulator on $\overline{\text{DATA}}$<0-15>. Should cause the interface selected by $\overline{\text{DS}}$<0-5> to load its B output buffer with the data on $\overline{\text{DATA}}$<0-15>.

DATOC *Data Out C.* Asserted by the processor during the execution of a **DOC** instruction, after the processor has placed the contents of the specified accumulator on $\overline{\text{DATA}}$<0-15>. Should cause the interface selected by $\overline{\text{DS}}$<0-5> to load its C output buffer with the data on $\overline{\text{DATA}}$<0-15>.

STRT *Start.* Asserted by the processor during the execution of any I/O instruction (except an *I/O Skip* instruction) in which bits 8 and 9=01 (i.e., any I/O instruction in which the *Start (S)* control function is specified). Not asserted during **DIA, DIB, DIC, DOA, DOB, DOC** instructions until after the data transfer has occurred. Usually used to initiate peripheral operation by setting the Busy flag to 1 and the Done flag to 0.

CLR *Clear.* Asserted by the processor during the execution of any I/O instruction (except an *I/O Skip* instruction) in which bits 8 and 9=10 (i.e., any I/O instruction in which the *Clear (C)* control function is specified). Not asserted during **DIA, DIB, DIC, DOA, DOB, DOC** instructions until after the data transfer has occurred. Usually used to terminate peripheral operation by setting the Busy and Done flags to 0.

IOPLS *I/O Pulse.* Asserted by the processor during the execution of any I/O instruction (except an *I/O Skip* instruction) in which bits 8 and 9=11 (i.e., any I/O instruction in which the *Pulse (P)* control function is specified). Not asserted during **DIA, DIB, DIC, DOA, DOB, DOC** instructions until after the data transfer has occurred. Usually used to initiate special peripheral operations.

SELB *Selected Busy.* Asserted low by the interface selected by the device select lines if its Busy flag is set to one.

SELD *Selected Done.* Asserted low by the interface selected by the device select lines if its Done flag is set to one.

Program Interrupt

INTR *Interrupt Request.* Asserted low by an interface to request program interrupt service.

MSKO *Mask Out.* Asserted low by the processor during the execution of the **MSKO** instruction, after the contents of the designated accumulator have been placed on **DATA<0-15>**. Used to load the contents of **DATA<0-15>** into the interrupt disable flip-flops of all interfaces using the interrupt system.

INTP *Interrupt Priority.* Seen asserted by the first interface on the I/O bus using the program interrupt facility, and transmitted in series through each successive interface. An interface should not issue an asserted **INTP OUT** unless it is receiving an asserted **INTP IN** and is not requesting interrupt service.

INTA *Interrupt Acknowledge.* Asserted by the processor during the execution of the **INTA** instruction. If an interface receives **INTA** while it is also receiving **INTP IN** asserted and while it is requesting interrupt service, it should place its device code on **DATA<10-15>**.

Data Channel

DCHR *Data Channel Request.* Asserted low by an interface when it requires data channel service.

DCHP *Data Channel Priority.* Seen asserted by the first data channel interface on the I/O bus, and transmitted in series through each interface. An interface should not issue an asserted **DCHP OUT** unless it is receiving an asserted **DCHP IN** and it is not requesting data channel service. Also used by some processors to determine data channel speed.

DCHA *Data Channel Acknowledge.* Asserted low by the processor at the beginning of each data channel cycle. Should cause the interface that is receiving an asserted **DCHP IN** signal and whose DCH REQ flip-flop is set, to set its DCH SEL flip-flop and place the memory address to be used for this transfer on **DATA<1-15>**, the map select bits on **DATA0** and **EXDCH**, and the mode on the data channel mode lines of the I/O bus.

DCHM0 *Data Channel Mode.* Asserted low by the interface whose DCH SEL flip-flop is set to inform the processor of the type of data channel cycle to be performed, as summarized in Table A.1:

DCHM0	Function
0 = H	Output
1 = L	Input

Table A.1 Data channel modes

EXDCH Asserted by the interface whose DCH SEL flip-flop is set to inform the processor to use data channel Map C or D. Used in conjunction with **DATA0**. (This signal is not used by all interfaces.)

DCHI *Data Channel Input.* Asserted by the processor for data channel input (**DCHM0**=1). Should cause the interface whose DCH SEL flip-flop is set to place the contents of its input register on **DATA<0-15>**.

DCHO *Data Channel Output.* Asserted by the processor for data channel output (**DCHM0**=0), after the data word has been placed on **DATA<0-15>**. Should cause the priority-selected interface to load the data from **DATA<0-15>**.

System Control

IORST *I/O Reset.* Asserted by the processor during the **IORST** instruction or when RESET occurs. **IORST** is also issued prior to processor operation at power turn-on. This signal should be used to initialize the machine state of all interfaces in the system.

RQENB *Request Enable.* Asserted low by the processor to synchronize program interrupt and data channel requests from all interfaces. In any interface, **INTR** and **DCHR** should be clocked only on the leading edge of **RQENB**.

Appendix B

Standard I/O Device Codes

Octal Device Code	Mnem	Priority Mask Bit	Device Name
00	—	—	Unused
01	APL	—	Automatic program load register
02	ERCC	—	Error checking and correction
03	MAP	—	Memory allocation and protection
04			
05	BMC	—	Burst multiplexor channel
06	MCAT	12	Multiprocessor adapter transmitter
07	MCAR	12	Multiprocessor adapter receiver
10	TTI	14	System console input
11	TTO	15	System console output
12	PTR	11	Paper tape reader
13	PTP	13	Paper tape punch
14	RTC	13	Real-time clock
15	PLT	12	Incremental plotter
16	CDR	10	Card reader
17	LPT	12	Line printer
20	DSK	9	Fixed-head disc
21	ADCV	8	A-D converter
22	MTA	10	Magnetic tape
23	DACV	None	D-A converter
24	DCM	0	Data communications multiplexor
25			
26	DKB	9	Fixed-head DG/Disc
27	DPF	7	DG/Disk storage subsystem
30	QTY	14	Asynchronous hardware multiplexor
30	SLA	14	Synchronous line adapter
31[1]	IBM1	13	IBM 360/370 interface
32	IBM2	13	IBM 360/370 interface
33	DKP	7	Moving head disk
34	CAS[1]	10	Cassette tape
	DCU[4]	4	Data control unit
34	MX1	11	Multiline asynchronous controller
35	MX2	11	Multiline asynchronous controller
36	IPB	6	Interprocessor bus—half-duplex
37	IVT	6	IPB watchdog timer
40[2]	DPI	8	IPB full-duplex input
40	SCR	8	Synchronous communication receiver
41[3]	DPO	8	IPB full-duplex output
41	SCT	8	Synchronous communication transmitter
42	DIO	7	Digital I/O
43	DIOT	6	Digital I/O timer
43	PIT	11	Programmable interval timer
44	MXM	12	Modem control for MX1/MX2
45			
46	MCAT1	12	Second multiprocessor transmitter
47	MCAR1	12	Second multiprocessor receiver
50	TTI1	14	Second system console input
51	TTO1	15	Second system console output
52	PTR1	11	Second paper tape reader
53	PTP1	13	Second paper tape punch
54	RTC1	13	Second real-time clock
55	PLT1	12	Second incremental plotter
56	CDR1	10	Second card reader
57	LPT1	12	Second line printer
60	DSK1	9	Second fixed-head disk
61	ADCV1	8	Second A-D converter
62	MTA1	10	Second magnetic tape
63	DACV1	None	Second D-A converter
64			
65	IOP1	5[5]	Host to IOP interface
66	DKB1	9	Second fixed-head DG/Disk
67	DPF1	7	Second DG/Disk storage subsystem
70	QTY1	14	Second asynchronous hardware multiplexor
70	SLA1	14	Second synchronous line adapter
71[1]		13	Second IBM 360/370 interface
72		13	Second IBM 360/370 interface
73	DKP1	7	Second moving head disk
74	CAS1	10	Second cassette tape
74[1]		11	Second multiline asynchronous controller
75		11	Second multiline asynchronous controller
76	FPU	—	Floating-point unit
77	CPU	—	Central processor and console functions

[1] *Code returned by INTA and used by VCT.*

[2] *Can be set up with any unused even device code equal to 40 or above.*

[3] *Can be set up with any unused odd device code equal to 41 or above.*

[4] *Can be set to any unused device code between 1 and 76.*

[5] *Micro interrupts are not maskable.*

Appendix C

Locations of Microcode on the microECLIPSE™ Chip Set

The following instructions are resident on the mE674 SPU.

- All ALC instructions: ADC, ADD, ANC, AND, ANDI, COM, INC, MOV, NEG, SUB
- All I/O instructions: DIA, DIB, DIC, DOA, DOB, DOC, INTA, INTDS, INTEN, IORST, MSKO, NIO, READS, SKPBN, SKPBZ, SKPDN, SKPDZ, HALT
- These arithmetic/logic instructions: ADDI, ADI, ANC, ANDI, DIV, DIVS, DIVX, HLV, IOR, IORI, MUL, MULS, SBI, XOR, XORI
- These memory/stack instructions: BLM, ELDA, ELEF, ESTA, LDA, LDB, LEF, LMP, MSP, POP, POPB, POPJ, PSH, PSHJ, RTN, SAVE, STA, STB
- These jump or skip instructions: DSPA, DSZ, EDSZ, EISZ, EJMP, EJSR, ISZ, JMP, JSR, SGE, SGT, SNB, SZB, SXBO
- These miscellaneous instructions: CLM, SYC, XCH, XCT

The ECLIPSE S/120 firmware floating point uses three XMCs. These are designated FXMC (DGC part number 100-000672), GXMC (DGC part number 100-000690), and HXMC (DGC part number 100-000697). The floating point instructions are allocated among them as follows.

- On the FXMC: FAB, FCLE, FCMP, FEXP, FFAS, FFMD, FHLV, FINT, FLAS, FLDD, FLDS, FLMD, FLST, FMMS, FMOV, FMS, FNEG, FNOM, FNS, FPOP, FPSH, FRH, FSA, FSCAL, FSEQ, FSGE, FSGT, FSLE, FSLT FSND, FSNE, FSNER, FSNM, FSNO, FSNOD, FSNU, FSNUD, FSNUO, FSS, FSST, FSTD, FSTS, FTD, FTE
- On the GXMC: FAD, FAMD, FAMS, FAS, FMD, FMMD, FSD, FSMD, FSMS, FSS
- On the HXMC: BAM, CMP, CMT, CMV, COB, CTR, DAD, DHXL, DHXR, DLSH, DSB, ELDB, ESTB, FDD, FDMD, FDMS, HXL, HXR, LOB, LRB, PSHR, RSTR, VCT, XOP, XOP1

Some instructions that are contained on the mE674 are also contained on XMCs. In these cases, the code contained on the XMC is executed.

Appendix D

SPU External Signals

Name	Pin(s)	Source	Destination	Description
NOVA/ECLIPSE I/O bus Signals				
DATA<0-15>	B62,65,82,73,61, 57,95,55,60,63,75, 58,59,64,56,66	I/O bus interface I/O bus	I/O bus I/O bus interface	
DS<0-5>	A72,68,66,46,62,64	I/O bus interface	I/O bus	
RQENB	B41	I/O bus interface	I/O bus	
INTR	B29	I/O bus interface	I/O bus	
DCHR	B35	I/O bus	I/O bus interface	
DCHA	A60	I/O bus interface	I/O bus	
DCHM0	B17	I/O bus	I/O bus interface	
EXDCH	B74	I/O bus	I/O bus interface	
DCHO	B33	I/O bus interface	I/O bus	
DCHI	B37	I/O bus interface	I/O bus	Refer to Appendix A for description.
SELB	A82	I/O bus	I/O bus interface	
SELD	A80	I/O bus	I/O bus interface	
DATIA	A44	I/O bus interface	I/O bus	
DATIB	A42	I/O bus interface	I/O bus	
DATIC	A54	I/O bus interface	I/O bus	
DATOA	A58	I/O bus interface	I/O bus	
DATOB	A56	I/O bus interface	I/O bus	
DATOC	A48	I/O bus interface	I/O bus	
INTA	A40	I/O bus interface	I/O bus	
MSKO	A38	I/O bus interface	I/O bus	
STRT	A52	I/O bus interface	I/O bus	
CLR	A50	I/O bus interface	I/O bus	
IOPLS	A74	I/O bus interface	I/O bus	
IORST	A70	I/O bus interface	I/O bus	
Front Panel Signals				
CONLOCK	B40	Chassis LOCK switch	SPU	Signals the position of the front console LOCK switch
CONPL	B48	Chassis PL switch	SPU	Signals the depression of the PL (Program Load) switch (if the front console is not locked)
CONRSTH,CONRSTL	B51,52	Chassis RESET switch	SPU	Signals the depression of the RESET switch (if the front console is not locked)
CONLED	B49	SPU	Chassis RUN indicator	Signals that user program is executing
System Console Signals				
TTIN	A94	System console	SPU	Serial data input
TTOUT	A85	SPU	System console	Serial data output (RS-232)

Table D.1 SPU external signals (see continuation)

Name	Pin(s)	Source	Destination	Description
ITTO	A83	SPU	System console	Serial data output (20MA- current loop)
CTS	A93	System console	SPU	Enables output data transmission
Power Supply Signals				
PWROK	B79	Power supply	SPU	Signals that dc power is above minimum specifications
PWRFAIL	B54	Power supply	SPU	Signals that ac power is not within specifications
MEMOK	A96	Power supply	SPU	Signals that battery backup (if present) maintained memory data during a power disruption
MEMDISASTER	B96	SPU	Power supply	Signals a failure on −5VMEM dc power
LCLK	A88	Power supply	SPU	A line frequency clock signal
DC Power				
Ground	A1,2,14,25,33,34 37,41,45,65,99,100 B1,2,21,39,50,68, 80,89,92,99,100	----------	SPU, System console, all I/O PCB's	
+5V	A3,4,97,98 B3,4,38,97,98	Power supply	SPU, System console, all I/O PCB's	
+5VMEM	B93,94	Power supply	SPU	
−5V	A6,B81	Power supply	SPU, System console, all I/O PCB's	
−5VMEM	B91	Power supply	SPU	
+12V	B87,88,90	Power supply	SPU, all I/O PCB's	
+12VMEM	B70,71,72	Power supply	SPU	
−12V	B77	Power supply	SPU	
+15V	A10,B46,B84	Power supply	SPU, all I/O PCB's	

Table D.1 SPU external signals (concluded)

Index of Terms

A

Accumulator(s)
  accessing with virtual console, 7-2 to 7-4
  CPU, 1-1, 1-5
  floating point, 1-1, 1-8
  instructions, 1-15
Add instructions, 1-11
Address
  latch, 2-22
  maps, 1-3
  phase, 2-1, 2-3, 2-15, 2-22
  space, logical, 1-5, 1-8, 2-15, 2-16, 2-29
  space, physical, 1-5, 1-8, 2-15, 2-16, 2-29
  translation, 1-8, 1-9, 2-15, 2-19, 2-33
  width, 1-1
Addressing modes, 1-5
Airflow through chassis, 3-1, 3-5, 4-1, 4-2
ALU, see Arithmetic/logic unit
APL, see Automatic program load
Architecture
  chassis, 5-slot, 4-1, 4-2
  chassis, 3-1, 3-5
  system, vii, 1-1, 2-1, 2-2
  system processing unit, 1-1, 2-1
Arithmetic/logic unit (ALU)
  and shifter, 2-4
Asynchronous line controller, (see also TTO/TTI) 1-2, 1-23, 1-26 to 1-28, 2-30, 2-31
  baud rates, 1-2, 2-30
  cables, 1-28
  configuration, 1-23, 1-26 to 1-27
  device code, 2-20, B-1
  modem control signals, 2-20
  priority mask bit, 2-20
  stop bits, 1-26, 2-31
Automatic program load, (see also Power-up response, Virtual console), 1-21, 1-22, 2-22
  device code, 1-21, 1-22, 2-22
  jumpers, 1-26, 1-27
  register, 1-4, 2-5, 2-22, 7-1, B-1
Autonomous prefetch unit, 2-3
Autorestart, see Powerfail/autorestart

B

Backpanel, 5-slot
  cabling, 4-2, 4-5, 4-6
  jumpering, 4-4
  layout, 4-4
  pcb assembly, 4-1, 4-6
  pin assignments, 4-4
Backpanel, 16-slot
  cabling, 3-7, 3-8
  jumpering, 3-6
  layout, 3-4
  pcb assembly, 3-1, 3-2, 3-8
  pin assignments, 3-4
Bandwidth
  I/O, 1-2
  memory, 1-1
Battery backup feature, 1-21
Battery backup, 5-slot, 6-15, 6-16
  battery, 6-15, 6-16
  block diagram, 6-16
  fuse, 6-16
  jumpering, when omitted, 4-4
  operation, 6-15
  powerfail sequence, 6-15
  status signals, 6/15, 6-17
Battery backup, 16-slot, 5-3 to 5-4, 5-7 to 5-8
  battery, 5-4
  block diagram, 5-7
  Compliant, 5-3 to 5-4
  fuse, 108
  Noncompliant, 5-7 to 5-8
  powerfail sequence, 5-8
  specifications, 5-8
  status signals, 5-8 to 5-9
  voltage regulators, 5-7 to 5-8
Batteries
  battery backup feature, 5-4, 5-7 to 5-8, 6-15
Bit addressing, 1-5
Block diagram(s)
  battery backup, 5-slot, 1-28, 3-7, 3-8, 4-5, 4-6
  battery backup, 16-slot, 5-7 to 5-8
  CPU, internal, 2-9
  CPU section, 2-6
  error checking and correction, 2-24
  MAP unit, 2-40
  NOVA/ECLIPSE I/O interface, 2-50

power supply, 5-slot, 6-13
power supply, 16-slot, 5-1, 5-4, 5-6,
SPU, 2-4
SIO, internal, 2-19
system, 1-30
memory, 2-34
virtual console, 2-23
Break enable jumper, 1-26, 1-27, 2-30
Break key, 1-21, 2-20
interrupt, 1-21, 2-20
Bus(ses)
arbitration, 2-2, 2-14, 2-48
address/data (see Bus, memory)
buffer control signals, 2-14, 2-55
console, 2-1, 2-4
CPU, 2-2
internal CPU, 2-3, 2-9
local input, 2-2
local output, 2-2
memory, (see Bus, CPU)
NOVA/ECLIPSE I/O, 2-1, 2-5, A-1 to A-5
system, (see Bus, CPU)
width of, 1-1
XMC, 1-30, 2-4, 2-9
Byte addressing, 1-5

C

Cables
asynchronous interface (system console), 1-28, 3-7, 3-8, 4-5, 4-6
I/O device, internal, 3-7, 3-8, 4-5, 4-6
power, 3-7, 4-1, 4-5
Cell(s) (virtual console), 7-2, 7-3
Cell commands (virtual console), 7-2, to 7-4
Central processing unit, 1-1, 1-2, 2-3, 2-9
ALU and shifter, 2-3, 2-9
autonomous prefetch unit, 2-3, 2-9
device instructions, 1-17
definition, i
done bit, 1-21
integrated circuit, 2-3, 2-9
internal busses, 2-3, 2-9
MAP status register, 2-3, 2-9
register file, 2-3, 2-9
section, 2-2, 2-3
section, block diagram, 2-6
skip flags, 1-7
specifications, viii
status register, 1-2, 1-21, 2-21
Chassis, 5-slot, 4-1 to 4-6
airflow in, 4-1, 4-3
compliant, 4-1 to 4-6
dimensions, 4-3
front view of, 98, 99
noncompliant 4-1 to 4-6
rear view of, 4-2, 4-6
pin assignments, backpanel, 4-4
slot assignments, 4-2
Chassis, 16-slot, 3-1 to 3-8
airflow in, 3-1, 3-5
compliant, 3-1 to 3-8
dimensions, 3-5
front view of, 3-1
noncompliant 3-1 to 3-8
rear view of, 3-2, 3-8
pin assignments, backpanel, 3-4
slot assignments, 3-3
+5V power distribution, 3-3
Configuration
APL jumpers (on SPU), 1-26, 1-27
minimum, vii
SIO jumpers (on SPU), 1-26, 1-27
SPU, 1-1, 2-1, 2-3
system, vii
system memory array, 2-33
Connector(s)
backpanel, 2-51, 3-4, 4-4
power supply, see backpanel
SPU board, A, B, 1-28, 1-29
front console, 3-2, 3-4, 4-201
Console bus, 2-1
Convert instructions, 1-13
CPU (see Central processing unit)
Current, 5-slot
available (power supply), viii, 6-14
draw, see power requirements
limiter, 6-13, 6-14
power supply output, viii, 6-14
Current, 16-slot
available (power supply), viii, 5-5, 5-8
draw, see power requirements
limiter, 5-3, 5-4
power supply output, viii, 5-5, 5-8

D

Data channel, 78-85 (see also I/O interface) 2-48 to 2-55
address latch, 2-50, 2-55
data latch, 2-50, 2-55
input timing, 2-53
loading via, 7-7
map, 1-8, 1-9
output timing, 2-54
rates, 1-2
state diagram, operation, 2-52
state machine, 2-29, 2-30
Data phase, 2-1, 2-3, 2-15, 2-31
Device code(s)
automatic program load (APL), 1-21, 1-22
asynchrounous line controller, input (TTI), 2-20, B-1
asynchrounous line controller, output (TTO), 2-20, B-1
central processing unit (CPU), B-1
error checking and correction (ERCC), B-1

memory allocation and protection unit (MAP), B-1
NOVA/ECLIPSE peripherals, B-1
programmable interval timer (PIT), 2-30, B-1
powerfail monitor, 1-2
real-time clock (RTC), 2-30, B-1
Diagnostic test plug, 3-2, 3-8
Dimensions
chassis, 5-slot, 4-3
chassis, 16-slot, 3-5
SPU board, 1-23
Divide instructions, 1-12

E

Emulator trap, 1-9
Error checking and correction,(see also Memory fault), 2-1, 2-34 to 2-39
bits, 1-9, 2-26
checking, 1-9
error detection correction chip, 2-25, 2-27
fault address register, 1-4, 2-25 to 2-27
fault code register, 1-4, 2-25 to 2-27
enable register, 1-4, 2-24, 2-26
fault, 1-10, 2-15, 2-26 to 2-29
instructions, 1-10, 1-18, 2-26, 2-27
interrupts, 2-27 to 2-29
sniffing, 1-3, 1-10, 2-26
Execution times, instruction, 1-1, 1-19 to 1-2
Expressions (virtual console), 7-2
Extended memory cycles, 2-15
Extended operations, 1-8, 2-1
External microcode controller, 1-1, 2-9
macroinstruction capacity, 2-9
location of microcode on, C-1
transfer protocol, 2-9

F

Fault(s)
floating point, 1-8
MAP, 1-8, 1-32
memory, (see Error Checking and Correction)
stack, 1-7
Features, system, viii
Floating point
accumulators, 1-1, 1-8
execution times, 1-20, 1-21
external microcode controller, 1-1, 2-9, C-1
faults, 1-8
operations, 1-7
range of decimal represented by, 1-8
status register, 1-4, 1-8
trap, 1-8
Flow diagrams
arithmetic/logic class (ALC) instruction sequences, 1-7
powerfail/autorestart sequence, 1-24, 1-25
power-up sequence, 1-22
system memory read/write, 2-16, 2-17
Format(s)
data, 1-5
floating point, 1-6
instruction, ii
integer, 1-6
signal name, ii
SPU program-accessible registers, 1-4
virtual console command, 7-2
Front console, 3-6, 4-4
Function commands (virtual console), 7-4
Fuses(s)
battery backup unit, 5-4, 6-16
line, 3-7, 4-6, 5-4, 6-16

H

Halt Dispatch
bit, 1-21, 1-26, 7-1
jumper, 1-26, 1-27
HALT instruction, 1-18, 1-21

I

Initialization, 1-3, 1-21
Input/Output (I/O) (see also data channel)
bus signals, summary, A-1 to A-5
clock cycle, 2-1
command flags, 1-16
decoder, 2-2, 2-22
interface, 2-2. 2-48 tp 2-55
interrupt instructions, 1-17
instructions, basic, 1-16
interrupt driven, 1-10
operations, 1-10, 2-1, 2-48 to 2-55
skip flags, 1-17
Installation, SPU board, 1-23
Instruction(s)
addressing, 1-5, 1-6
addressing modes, 1-5
arithmetic/logic class (ALC), 1-7, 1-8
computing, 1-10 to 1-14
device management, 1-17, 1-18
error checking and correction, 1-9, 1-10, 1-18
execution times, 1-1, 1-19 to 1-20
HALT, 1-21, 1-26, 2-21
input/output (I/O), 1-10, 1-16, 1-17
MAP manipulation, 1-8, 1-18
program flow management, 1-10, 1-15, 1-16
READS, 7-3, 7-7
SPU, 1-3 to 1-18
stack, 1-7, 1-10, 1-16
Interfacing (also interconnections)
power supply, 5-slot, 6-15
power supply, 16-slot, 5-8, 5-9
SPU board, 1-28, 1-29

Interrupt(s)
facilities, 1-2
frequencies of RTC, 1-2
Halt, 1-21, 1026, 2-21
identification of, 1-10
instructions, 1-15, 1-17
intervals of PIT, 1-2
latency, 1-1, 2-20
maskable, 1-2, 1-10, 2-21
nonmaskable, 1-2, 2-20, 2-21
powerfail, 2-20, 2-21
powerfail routine, 1-21, 1-24, 1-25
power-change, 1-21, 2-20
priorities, 1-2, A-1
program load, 2-21
request, 2-20
Interrupt-on bit, 51 (see also CPU status register) 1-4, 2-21
I/O Reset instruction, 1-18, 1-21, 1-25, 2-20, 2-26, 2-29, 7-7

J

Jump instructions, 1-15
Jumper(s)
automatic program load (APL), 1-26. 1-27
asynchronous interface operating characteristics, 1-23, 1-26
asynchronous interface transmission rate, 1-26
BREAK enable, 1-26, 1-27
diagnostic, 1-26. 1-27
halt dispatch, 1-26. 1-27
programmable interval timer (PIT), 1-26, 1-27
SPU board, 1-23, 1-26, 1-27

L

LED (light emitting diode)
power supply, 3-7
front console, 3-6. 4-5
Line cord, part numbers, 3-7 4-5
Load map operation, 2-29
Local space, 1-5, 1-8, 2-15, 2-16, 2-29
Locked operation, 1-21, 1-22, 3-6, 4-5
LOCK switch, 2-22, 3-6, 4-5
Logic instructions, 1-13

M

Maskable interrupts, 1-10, 2-27, 2-28
MAP (see memory allocation and protection)
Maps, number of, 1-8
Memory, system, 1-3, 2-2, 2-33 to 2-47
access time, 1-1, 2-33
addres register, 2-33, 2-34
address selection, 2-33, 2-36
address width, 1-1
banks, 2-33
block diagram, 2-34
bus, 2-2
bus control signals, 2-13, 2-14
bus buffers, 2-2
cell, 7-2
characteristics, viii, 2-33
crital voltage monitor, 2-44
configuration, vii
correction opration, 2-34
initiating, 2-33, 2-36
logical, 1-5, 1-8
mapping, see memory allocation and protection
maximum, system, 1-2
operations, pended, 2-37 to 2-47
physical, 1-5, 1-8, 2-2, 2-15, 2-29, 2-33
read operations, 2-36 to 2-39
refresh operations, 2-44 to 2-42
reserved locations in, 1-18
select logic, 2-33, 2-36
specifications, viii, 2-33
write operations, 2-40 to 2-43
Memory allocation and protection (MAP) unit, 1-3, 2-29 to 2-33
address register, 1-4
address translation, 1-8, 1-9, 2-29, 2-32
block diagram, 2-30
changing with virtual console, 7-6
current state of, 7-1
emulator trap feature, 1-9
fault, 1-8, 2-15, 2-32 to 2-33
initial (power-up) state, 1-8, 1-21
initiate page check register, 1-4
instructions, 1-8, 1-18
Load Effective Address instruction, 1-8
load map operations, 2-29
mapped mode operation, 2-32
memory protection logic, 2-33
operations, 1-8
page check register, 1-4, 2-29, 2-32
page 31 register, 1-4, 1-8, 1-21, 2-30, 2-32
power-up status, 1-8, 1-21
protection, 1-8
RAM memory, 2-29 to 2-32
status register, 1-4, 1-21, 2-3, 2-9, 2-29, 2-30, 2-32
unmapped mode operation, 2-32
Memory fault
address, see error checking and correction fault address
address register, 104, 2-25 to 2-7
code register, 1-4, 2-25 to 2-27
Memory, MAP (see Memory allocation and protection)
Memory, virtual console, (see Virtual console)
mE674 (see central processing unit)
mE676 (see System Input/Output chip)
Microcode
location of, 2-9, C-1
transfer protocol, 2-9

Microcycle, 2-1
Model numbers, system, vii
Move instructions, 1-12
Multimaster bus, 2-2, 2-14, 2-48
Multiply instructions, 1-11

N

Nonmaskable interrupts, 1-10, 2-30
NOVA/ECLIPSE I/O bus, 2-2
  interface, 2-48 to 2-35
  number of controllers on, 1-10

O

Overvoltage, 5-6, 6-14

P

Page number
  logical, 2-29, 2-33
  physical, 2-29, 2-33
Physical memory, 1-5, 1-8, 2-2, 2-15, 2-29, 2-33
Pin assignments (see connectors)
Pipelines, IR and PC, 2-3, 2-9
Power change interrupt, 2-20
Power monitor, 1-2, 1-21, 2-20
Power requirement(s)
  power supply,
  SPU board, 1-23
  system, viii
Powerfail/autorestart,
Powerfail
  monitor,
  timing,
Power-up response, 1-21
Power supply, 5-slot, 6-1 to 6-17
  ac inputs, viii
  auxiliary voltage supply, 6-2, 6-4, 6-13, 6-14
  battery backup, 6-7, 6-8, 6-15
  block diagrams, 6-2, 6-3, 6-6, 6-13, 6-16
  compliant, 4-1, 6-10, 6-11
  current limiter, 6-7, 6-8
  dc outputs, viii, 6-1 to 6-14
  fault detection circuits, 6-8, 6-14
  filter, 6-16, 6-3, 6-12, 6-13
  interconnections, 6-11, 6-12, 6-15
  inverter, 6-12, 6-14
  jumpering, battery backup omitted, 4-4
  line cord connection, 4-2, 4-6
  LOCK switch, effect on, 4-5,
  noncompliant, 4-1, 6-12 to 6-17
  overview, 6-4, 6-12
  over-voltage shutdown levels, 6-14
  over-current detector, 6-8, 6-14, 6-15
  over-voltage detector, 6-8, 6-14
  power fail detector, 6-9, 6-14
  printed circuit board, 4-1
  pulse width modulation control, 6-12, 6-15, 6-18
  POWER switch, 4-5
  PWR indicator, 4-5
  real-time clock circuits, 6-13, 6-15, 6-18
  schematic number, 6-12
  short circuit detection, 6-8, 6-14 to 6-15
  soft start, 6-13, 6-14
  specifications, viii, 6-1, 6-14
  status signals, 6-9, 6-13, 6-17
  transformer, 6-6, 6-8, 6-9, 6-14
  under-voltage detecter, 6-8, 6-14
  VNR source, 6-4, 6-5, 6-12, 6-13, 6-14
Power supply, 16-slot, 5-1 to 5-9
  ac inputs, viii
  auxiliary suppy regulators, 5-5
  battery backup, 5-7, 5-8
  block diagrams, 5-1, 5-4, 5-6
  buck regulator, 5-1, 5-3
  clock logic, 5-5
  compliant, 3-1, 3-3
  dc outputs, viii, 5-5
  dc to dc converter, 5-3
  fault indicators, 3-7
  filter, 5-1
  interconnections, 5-8, 5-9
  line cord connection, 3-7
  LOCK switch, effect on, 3-6, 4-5
  memory disaster detector, 5-6
  noncompliant, 3-1, 3-3
  overview, 5-1, 5-3
  over- and undervoltage thresholds, 5-6, 5-7
  over-current detector, 5-5 to 5-6
  over-voltage detector, 5-6
  power on/off sequence, 5-4
  POWER switch, 3-6
  primary current limiter, 5-5, 5-6
  printed circuit board, 3-1, 5-1, 5-5
  PWR indicator, 3-6
  real-time clock circuits, 5-8
  rectifier, 5-1
  sequencing logic, 5-5
  short circuit detection, 3-23, 3-3
  specifications, viii, 5-5, 5-8
  status signals, 5-7 to 5-9
  supply regulators, 5-5 to 5-7
  switching regulator, 5-5
  transformer, 5-5
  under-voltage detecter, 5-7
  VNR unit, 3-1, 3-2, 5-1 to 5-5
PWR indicator, 3-6, 4-5
POWER switch, 3-6, 4-5
Priorities, interrupt, 1-2, A-1
Priority mask bit(s)
  Asynchronous line controller, input (TTI), 2-20
  Asynchronous line controller, output (TTO), 2-20
  Programmable interval timer (PIT), 2-20
  Real-time clock (RTC), 2-20

PR/LOAD switch, 1-22, 3-6, 4-5, 7-1, 7-7
Programmable interval timer, 1-2, 1-26, 1-27, 2-20
  configuration, 1-26,1-27
  count register, 1-4
  device code, 2-20
  initial count register, 1-4
  interrupt latency, found by, 2-20
  priority mask bit, 2-20
  rates, (see also configuration above), 1-2
Programmable read-only memories (PROMs)
  SPU, timing/control, 2-9
  SPU, data channel state machine, 2-49
  SPU, virtual console, 1-1, 1-3, 2-22, 2-23, 7-1
Programmed I/O
  devices, loading from, 7-7
Prompt(s), virtual console, 1-21, 1-22, 7-1, 7-2
Protection, (see also memory allocation and protection)
  bits, 1-8, 2-29
  indirection, 1-8, 2-29, 2-32
  Input/Output (I/O), 1-8, 2-29, 2-32
  write, 1-8, 2-29, 2-32
  validity, 1-8, 2-29, 2-32, 2-33

R

RAM array, 2-33, 2-35, 2-36
Random access memory
  characteristics, 2-33
  specifications, viii, 2-33
Range
  decimal, of floating point number, 1-7
  of integers, signed and unsigned, 1-6
  of memory addresses, 2-33
  of power supply voltages, 6-1, 6-14
Rate(s)
  baud (TTI/TTO), 1-2, 1-26
  data channel transfer, 1-2
  programmable interval timer (PIT), 1-2, 1-26
  real-time clock (RTC), 1-2
Real-time clock, 1-2, 2-30
  circuits (power supply, 5-slot), 6-13, 6-14
  circuits (power supply, 16-slot), 5-7 to 5-9
  device code, 2-20
  initial (power-up) state, 2-20
  priority mask bit, 2-20
  rate, 1-2
  register, 1-4
Refresh
  address, 2-44
  interval, 2-44
  operation, 2-44
  pended, 2-15, 2-44
Register(s)
  automatic program load (APL), 1-4, 2-5, 2-22, B-1
  CPU status, 1-2, 1-4, 1-21, 2-21
  device status, 1-4
  floating point status, 1-4
  MAP address, 1-4
  MAP address translation, 1-9
  MAP initiate page check, 1-4
  MAP page check, 1-4,2-29,2-32
  MAP page 1-4, 2-1
  MAP status, 1-4
  memory fault address, 1-4
  memory fault code, 1-4,
  error checking and correction (ERCC) enable,1-4, 2-25
  programmable interval timer (PIT) count, 1-4
  programmable interval timer (PIT) initial count, 1-4
  RAM memory, internal,
  real-time clock (RTC), 1-4
  system input/output (SIO) jumpers, 1-26, 1-27, 2-21
  system processing unit (SPU) program-accessible, 1-4
  TTI character buffer, 1-4
  TTO character buffer, 1-4
  virtual console, 1-4,
Related references, ii, 1-5, 1-8, 2-2, 2-20, 2-26, 5-8, 6-12
Reserved memory locations, 1-18, 1-19
Reset
  I/O, (see I/O Reset instruction)
  switch, 2-20
  system, (see also power-up response). 2-22
Rubout key (with virtual console), 7-2
Run light, 1-22, 2-26

S

Schematic number(s)
  system processing unit (SPU), 2-2
  power supply, 6-12
Self-test, 1-21, 1-22, 7-1, 7-7
Significant decimal digits, 1-7, 1-8
Signals (see also signal index)
  SPU external, D-1, D-2
SIO chip, see system input/output chip
Skip instructions
  computational, 1-14
  noncomputational, 1-14
Skip flags
  CPU, 1-18
  I/O, 1-17
Slot assignments
  5-slot chassis, 3-3
  16-slot chassis, 4-2
Specifications
  battery backup feature, 5-7
  central processing unit (CPU), viii
  memory, viii
  power supply, input, viii
  power supply, output, viii, 5-5, 5-8, 6-1, 6-14
  system processing unit (SPU), power consumption, 1-23
  system, viii
  timing, viii (see also Timing)

SPU, see System processing unit
Stack and data management instructions, 1-16
Stack
  operations,1-7
  faults, 1-8
State diagram, data channel, 2-52
State machine(s)
  NOVA/ECLIPSE I/O interface, 2-49, 2-50
  SPU timing, 2-9
Status instructions, 1-14
Subtract instructions, 1-11
Subroutine instructions, 1-15
Switch(s)
  LOCK, 1-21, 1-22, 1-25, 3-6, 4-5
  PL/LOAD - RESET, 1-22, 3-6, 4-5, 7-1, 7-7
  POWER, 3-6, 4-5
System
  architecture, vii
  bus, 2-2
  definition, vii
  features, viii
  model numbers, vii
  minimum configuration, vii
  specifications, viii
  timing, 2-1, 2-9 to 2-13
System input/output
  chip, 1-1. 1-2
  register, (See also jumpers below), 2-20, 2-21
  jumpers, 1-26, 1-27
System management instructions, 1-10, 1-11
System processing unit (SPU),
  architecture, 1-1, 2-1
  bus arbitration logic, 2-2, 2-14, 2-78
  bus control signals, 2-13, 2-14
  components, 1-1
  central processing unit (CPU) section, 2-2, 2-3
  CPU status register, 1-2, 1-21, 2-21
  CPU support elements, 2-18 to 2-33
  definition, i
  interfacing, 1-1, 1-28, 1-29
  jumpers, 1-23, 1-26, 1-27
  major elements, 1-1 to 1-3
  memory, 2-33 to 2-47
  NOVA/ECLIPSE I/O interface, 2-48 to 2-55
  on-board busses, 2-2
  power requirements, 1-23
  power-up response, 1-21
  tailoring (see jumpers above)
  timing/control circuits, 2-1, 2-9 to 2-13
  20-MHz reference signal, 2-9

T

Tailoring, see Configuration
Threshold voltage levels (power supply), 5-5, 6-1, 6-14
Time(s)
  data channel state machine, 2-49
  instruction execution, 1-1, 1-19 to 1-21
  interrupt latency, 1-1
  I/O clock cycle, 2-1
  memory cycle, viii, 1-1
  microcode fetch, 2-9
  microcycle, 2-3
  operating, after power failure, 1-3
  self-test execution, 1-21
Timing
  data channel, viii, 1-2
  decoder, SPU,
  finite state machine, SPU, 2-9
  I/O interface, 2-1
  memory, read, 2-37, 2-38
  memory, refresh, 2-45, 2-46
  memory, write, 2-41, 2-42
  extended memory operation, 2-15
  powerfail, 1-21
  power-up, 1-21
  system, 2-1
Timing diagrams
  control, timed, 2-12
  data channel input, 2-53
  data channel output, 2-54
  extended memory cycle, 2-18
  memory bus control, 2-18
  memory byte write, 2-42
  memory read, 2-37, 2-38
  memory refresh, 2-45, 2-46
  memory word write, 2-41
  system, state machine, 2-10, 2-11
Transformer, power, 5-1, 5-3, 6-6, 6-8, 6-9, 6-13, 6-14
TTO/TTI (see also asynchronous line controller)
  character buffer registers, 1-4

V

Virtual console,1-1, 1-3, 2-22, 7-1 to 7-8
- breakpoints, 7-4, 7-5
- cells, 7-2, 7-3
- cell commands, 7-2 to 7-4
- changing the MAP from, 7-6
- correcting entry errors, 7-2
- delete key with, 7-2
- entering conditions, 1-21, 7-1
- function commands, 7-4
- interrupts, (see also nonmaskable interrupts), 2-21
- program load commands, 2-21
- program load device code for READs, 1-18, 7-7
- prompt, 1-21, 1-22, 7-1, 7-2
- read/write operations, 2-22
- register, 1-4, 7-3
- resuming program execution from, 7-5, 7-6
- rubout key with, 7-2
- search, 7-7
- self-test, 1-21, 1-22, 7-1, 7-7
- single stepping, 7-5
- test commands, 7-7, 7-8
- user memory test, 7-8

X

XMC, see external microcode controller

Index of Signals

-5MEM,2-34, 2-44, 3-7, 6-1, 6-4 ,6-6, 6-8 to 6-11, 6-14 to 6-17
-5MOK, 6-3, 6-6, 6-7, 6-8, 6-9
-5V, 6-1 to 6-6, 6-8, 6-10 to 6-14, 6-17
-5VMEM, See -5MEM
-11V, 6-12, 6-13, 6-14, 6-17
-12CS,
-12V, 6-1, 6-3, 6-4, 6-5, 6-10, 6-11, 6-12
-300VNR, 6-13
.7REF, 6-2, 6-4
2XCLCK, 2-9, 3-49, 3-50
2.7REF, 6-2, 6-4
3A, 6-16
5CLK, 6-13
10CLK, 6-13
20CLK, 6-13
40CLK, 6-13
5MHZCLK, 6-3, 6-11
300VNR, 5-4, 5-6, 6-12, 6-14
500HZ, 6-3, 6-11, 6-13
+3V, 6-1, 6-2, 6-3, 6-4, 6-5, 6-10, 6-11, 6-17
+5AUX, 6-2, 6-4, 6-13, 6-14
+5AUXBBU, 6-2, 6-4
+5MEM, 4-4, 6-1 to 6-4, 6-6 to 6-11, 6-14, 6-15, 6-16, 6-17
+5SEC, 6-2, 6-3, 6-5
+5V, 6-1, 6-4, 6-5, 6-10, 6-12 to 6-15
+5VA, 4-4, 6-2, 6-3, 6-5, 6-11, 6-13, 6-17
+5VAUX, See +5AUX
+5VB, 6-3, 6-11, 6-13, 6-17
+5VC, 6-3, 6-5, 6-11
+5VMEM, See +5MEM
+11SW, 5-5
+12AUX, 6-1, 6-4
+12AUXBBU, 6-2, 6-4
+12MEM, 2-34, 2-44, 6-1, 6-2, 6-4 to 6-11, 6-13 to 6-17
+12MEMSEC, 6-6, 6-7
+12V, 6-1 to 6-8, 6-10 to 6-14, 6-17
+12VMEM, See +12MEM
+12VNR, 6-3, 6-5, 6-6
+15V, 6-12, 6-13, 6-14, 6-17
+15VAUX, 6-13, 6-14

A

AUX, 6-2
Ax, 2-2, 2-5, 2-14, 2-22, 2-23, 2-25, 2-30, 2-33, 2-35, 2-48, 2-50, 2-51, 2-55
AxH01, 2-35, 2-36
AxH23, 2-35, 2-36
AxL01, 2-35, 2-36
AxL23, 2-35, 2-36
ACHOT, 5-3
ACK, 2-6
ACRELAY, 5-3
ACS, 5-3, 5-4
ADREN, 2-9, 2-13, 2-15, 2-19, 2-22, 2-30, 2-33, 2-34, 2-36, 2-40, 2-44 2-49, 2-50
APLSEL, 2-32
AUXIN, 5-3
AUXSW, 5-2
AUXVNR, 5-3, 5-4

B

B10CLK, 6-3, 6-11
B20CLK, 6-3, 6-14
B40CLK, 6-3, 6-11
BANKx, 2-35
BATON, 5-8
BATT, 6-15
BBU, 6-6, 6-7
BBUSHUTDOWN, 6-6, 6-8
BLOCK, 2-9, 2-14, 2-48
BMBx, 2-2, 2-5, 2-7, 2-14, 2-24, 2-30, 2-32
BREQ, 2-6, 2-9, 2-48 to 2-50
BUSDRV, 2-49, 2-50, 2-55
BUSOK, 2-14, 2-48 to 2-50

C

CAS, 2-11. 2-34, 2-36, 2-44
CASxH, 2-35, 2-36
CL, 6-6
CASxL, 2-35, 2-36
CHGRELAY, 5-4
CLR, 2-22, 2-31, A-2, A-4, D-1
CMBx, 2-2, 2-4, 2-7, 2-14

CNTRST, 2-9, 2-48, 2-51
CO, 2-5, 2-14, 2-18, 2-19, 2-50, 2-55
COLSEL, 2-11, 2-34, 2-36
CONLOCK, D-2
CONPL, D-2
CONRSTH, D-2
CONRSTL, D-2
CONLED, D-2
CORCLR, 2-9
CORCYC, 2-11, 2-34
CRx, 2-4, 2-6, 2-9
CTS, 2-4, 2-19, 2-20, D-2
CYCACT, 2-11, 2-24, 2-25, 2-50, 2-55
CYCCLR, 2-11, 2-34
CYCTST, 2-9

D

DATAx, 2-1, 2-5, 2-48 to 2-51, 2-55, A-1, A-4, D-2
DATEN, 2-9, 2-11, 2-14, 2-15. 2-19, 2-22, 2-23, 2-25, 2-27, 2-30, 2-37
DCA, 2-49, 2-50
DCHA, 2-51, A-4, D-1
DCHDRV, 2-15, 2-49, 2-50, 2-55
DCHI, 2-49 to 2-51, A-4, D-1
DCHINP, 2-49 to 2-51
DCHM0, 2-49, 2-51, A-5, D-1
DCHO, 2-49 to 2-51, A-2, A-3, D-1
DCHP, 3-6, 4-4, A-5
DCHPIN, 3-6, 4-4, A-5
DCHPOUT, 3-6, 4-4, A-5
DCHR, 2-48, 2-49, 2-51, A-5, D-1
DCREQ, 2-48 to 2-51
DENABLE, 2-49, 2-50
DIA (also **DATIA**, 2-32, 2-35, 2-37, 2-51
DIB (also **DATIB**, 2-32, 2-35, 2-37, 2-51
DIC (also **DATIC**, 2-32, 2-35, 2-37, 2-51
DIS, 2-21, 2-25, 2-29, 2-50
DISCORERR, 2-24
DLATCH, 2-49, 2-50
DMAPx, 2-30, 2-32, 2-51
DOA (also **DATIA**, 2-22, 2-24, 2-26, 2-29, 2-51, A-2, A-4, D-1
DOB (also **DATIB**, 2-22, 2-31, 2-32, 2-51, A-2, A-4, D-1
DOC (also **DATIC**, 2-30, 2-32, 2-32, 2-39, 2-51, A-2, A-4, D-1
DSx, 2-48, 2-51, A-2, A-4, D-1
DV, 6-6
DV77SEL, 2-21, 2-22

E

ECCCOR, 2-9, 2-25
ECCDETECT, 2-24
ECCDR0, 2-24
ECCDR1, 2-24
ECCGEN, 2-9, 2-25
ECCINTEN, 2-25
ECCIREQ, 2-25, 2-51
ECCLATCH, 2-9, 2-25
ECCSEL, 2-22, 2-24 to 2-27, 2-29
ECCWRT, 2-9, 2-14, 2-14, 2-24, 2-27, 2-28
ENB10CLK, 6-3, 6-11, 6-13
ERCCERR, 2-25, 2-26, 2-29, 2-37
ERR, 2-24, 2-25, 2-27, 2-28
EXDCH, 2-49, 2-51, A-2, A-3, D-1

F

F1, 6-2
F2, 6-6
FETCH, 2-9
FRCNMI, 2-24

G

GND, 6-11, 6-17
GRANT, 2-30, 2-32, 2-49, 2-50

H

HBEN, 2-14, 2-22, 2-23, 2-30
HROMSL, 2-23, 2-30
HVS, 5-3, 5-4, 6-9
HWRTPLS, 2-9, 2-14, 2-22, 2-23, 2-29, 2-30

I

INT, 2-9
INTA, 2-22, 2-51, A-1 to A-3, A-5, D-1
INTACYC, 2-22, 2-49, 2-50
INTMOK, 6-2, 6-6, 6-7, 6-8
INTP, 3-6, 4-4, A-3
INTPF, 6-2, 6-5, 6-6, 6-13, 6-14, 6-15, 6-16
INTPIN, 2-25, 2-51, 3-6, 4-4
INTPOK, 6-2, 6-6, 6-8, 6-13, 6-14, 6-15, 6-16
INTPOUT, 3-6, 4-4, A-4
INTR, 2-19, 2-51, A-3, A-4, D-1
INTRQ, 2-20, 2-21, 2-51
IOADREN, 2-49, 2-50
IODATEN, 2-6, 2-49, 2-50, 2-55
IOOUTx, 2-2, 2-5, 2-7, 2-14, 2-25, 2-25, 2-31, 2-32, 2-48 to 2-50, 2-55
IOPLS, 2-22, 2-51, A-3, A-4, D-1
IORST, 2-22, 2-24 to 2-26, 2-51, A-3, A-5, D-1
ITTO, 1-28, D-2

J

JUMPER1, 6-11
JUMPER2, 6-11
JUMPER3, 6-11

L

LBEN, 2-14, 2-22, 2-13, 2-30
LCLK, 6-3, 6-13, 6-15, D-2
LCLSEL, 2-22, 2-50, 2-55
LFREQ, 2-19
LMAPEN, 2-15, 2-30 to 2-32
LMAx, 1-9, 2-5, 2-25, 2-26, 2-33, 2-34, 2-36
LMB0, 2-23, 2-30, 2-32
LMECCDETECT, 2-27
LMEMCYC, 2-14, 2-15, 2-22, 2-23, 2-25, 2-30, 2-32, 2-35, 2-50, 2-55
LMPSLx, 2-29, 2-30, 2-32
LRAMDx, 2-31
LVALERR, 2-14, 2-15, 2-24, 2-25, 2-27, 2-31, 2-33, 2-37
LWH, 2-13, 2-14, 2-22 to 2-24, 2-28, 2-30
LWL, 2-13, 2-14, 2-22 to 2-24, 2-28, 2-30
LWRTINH, 2-15, 2-31, 2-33
LXADRx, 1-9, 2-5, 2-25, 2-26, 2-33, 2-34, 2-36

M

MAPEN, 2-9, 2-13, 2-22, 2-33
MAPSEL, 2-14, 2-22, 2-29 to 2-32
MBIT1, 2-29, 2-30
MBx, 2-2, 2-4 to 2-6, 2-14, 2-15, 2-19, 2-21 to 2-30, 2-32 to 2-34, 2-37 2-40, 2-48 to 2-51
MECCDETECT, 2-25, 2-27, 2-28
MEMAx, 2-35, 2-44
MEMCYC, 1-9, 2-9, 2-13, 2-14, 2-19, 2-22, 2-33
MEMD (see also **MEMDISASTER** below), 5-6, 6-17
MEMDISASTER (see also **MEMD** above), 2-34, 2-44, D-2
MEMIOx, 2-2, 2-5, 2-7, 2-14, 2-25, 2-27, 2-28
MEMOK, 5-7, 5-8, 6-3, 6-9, 6-10, 6-13 to 6-17, D-2
MEMSEL, 2-6, 2-24, 2-25, 2-35, 2-36
MEMSTRT, 2-9
MINx, 2-5, 2-25, 2-27, 2-28, 2-35, 2-40
MOUTx, 2-5, 2-24, 2-25, 2-27, 2-28, 2-35, 2-36, 2-40
MSKO, 2-22, 2-51, A-3, A-4, D-2
MWRITE, 2-30

N

NEUTRAL, 6-2, 6-11
NMI, 2-9, 2-19, 2-20
NOECC, 2-25

O

ONLED, 5-6, 6-3, 6-9
ONLED-P, 6-17
ONSW, 5-5, 5-6, 6-5, 6-6, 6-9
OV, 6-7
OVERLOAD, 6-8

P

PF, 2-19, 2-20
PFAIL, 1-21, 2-20
PHASE, 6-11
PHx, 2-6, 2-9, 2-15, 2-19
PIPE, 2-9
PON, 5-6, 5-9
PVPROT (see also **LVALERR**), 2-9
PWPROT (see also **LWRTINH**), 2-9
PWROK, 1-3, 1-21, 2-20, 5-6, 5-7, 6-2, 6-3, 6-8, 6-9, 6-10, 6-13, 6-14, 6-17, D-2
PWRFAIL, 1-3, 1-21, 2-20, 2-21, 5-6, 6-2, 6-3, 6-5, 6-9, 6-10, 6-13, 6-14, 6-17, D-2

R

RAMAx, 2-23, 2-30
RAMDIR, 2-30
RAMDx, 2-2, 2-5, 2-7, 2-13, 2-14, 2-22, 2-23, 2-29, 2-31 to 2-34
RAMP, 5-5
RAS, 2-14, 2-24, 2-26, 2-34
RASxH, 2-35, 2-36
RASxL, 2-35, 2-36
READ, 2-14, 2-25, 2-31
READY, 2-6, 2-9, 2-11, 2-14, 1-15, 2-19, 2-14, 2-44, 2-48
REFCYC, 2-6, 2-13, 2-24, 2-25, 2-28, 2-31, 2-34, 2-44
REQ, 2-9
RESET, 1-21, 2-4, 2-6, 2-19 to 2-21, 2-35
RETURN, 6-11
RQEN, 2-49 to 2-51
RQENB, 2-48 to 2-51
RVS, 6-2, 6-3

S

SC DET, 5-5
SCx, 2-25 to 2-28
SDI, 2-4
SDO, 2-4
SELB, 2-48, 2-51, A-3, A-4, D-1
SELD, 2-48, 2-51, A-3, A-4, D-1
SHUTDOWN, 6-5, 6-6, 6-7, 6-8
SINx, 2-5, 2-25, 2-27, 2-28, 2-35, 2-40
SOUTx, 2-5, 2-25, 2-27, 2-28, 2-35, 2-36
SSCAP, 6-2, 6-5
STATUS, 2-9
STRT, 2-22, 2-25, 2-29, 2-31, A-2, A-4, D-1
SV31, 2-23, 2-30, 2-32

T

T2, 6-13
T3, 6-14
TP, 2-5, 2-14, 2-18, 2-19, 2-50, 2-55
TTI, 2-19

TTIN, 1-28, 2-4, D-2
TTL01, 2-6, 2-9, 2-11, 2-13, 2-15, 2-22, 2-33, 2-34, 2-49, 2-50
TTL02, 2-6, 2-9, 2-14, 2-25, 2-48
TTO, 2-19
TTOUT, 1-28, 2-4, D-2

U

UMAPx, 2-30, 2-32
UNMAP, 2-23, 2-30
USRMEM, 2-6, 2-30, 2-32

V

VAC, 6-2, 6-3, 6-4, 6-13
VAUX, 6-13, 6-14
VNR, 6-2, 6-4, 6-5, 6-11

W

WE01H, 2-35, 2-40
WE01L, 2-35, 2-40
WE23H, 2-35, 2-40
WE23L, 2-35, 2-40
WEPLS, 2-11, 2-13, 2-33, 2-34, 2-36, 2-40, 2-44
WH, 2-9, 2-13, 2-15, 2-19, 2-22, 2-33, 2-36, 2-40, 2-49, 2-50
WL, 2-9, 2-13, 2-15, 2-19, 2-22, 2-33, 2-36, 2-40, 2-49, 2-50
WPEN, 2-30, 2-32
WRTCYC, 2-13, 2-14, 2-24, 2-25, 2-7, 2-28, 2-50, 2-55
WRTEN, 2-9

DataGeneral

Design Services Publications Comment Form

Please help us improve our future publications by answering the questions below. Use the space provided for your comments.

Title: _____

Document No. _____

Yes	No			
☐	☐	Is this manual easy to read?	O You (can, cannot) find things easily. O Language (is, is not) appropriate. O Technical terms (are, are not) defined as needed.	O Other:
		In what ways do you find this manual useful?	O Learning to use the equipment O As a reference O As an introduction to the product	O To instruct a class. O Other:
☐	☐	Do the illustrations help you?	O Visuals (are,are not) well designed. O Labels and captions (are,are not) clear. O Other:	
☐	☐	Does the manual tell you all you need to know? What additional information would you like?		
☐	☐	Is the information accurate? (If not please specify with page number and paragraph.)		

Name: _____ Title: _____

Company: _____ Division: _____

Address: _____ City: _____

State: _____ Zip: _____ Telephone: _____ Date: _____

DG-06895

DataGeneral

Data General Corporation, Westboro, Massachusetts 01580

CUT ALONG DOTTED LINE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 26 SOUTHBORO, MA 01772

Postage will be paid by addressee

DataGeneral

Design Services
M.S. E-219
4400 Computer Drive
Westboro, MA 01581

DataGeneral

Design Services Publications Comment Form

Please help us improve our future publications by answering the questions below. Use the space provided for your comments.

Title: _____

Document No. _____

Yes	No		
☐	☐	**Is this manual easy to read?**	○ You (can, cannot) find things easily. ○ Other: ○ Language (is, is not) appropriate. ○ Technical terms (are, are not) defined as needed.
		In what ways do you find this manual useful?	○ Learning to use the equipment ○ To instruct a class. ○ As a reference ○ Other: ○ As an introduction to the product
☐	☐	**Do the illustrations help you?**	○ Visuals (are,are not) well designed. ○ Labels and captions (are,are not) clear. ○ Other:
☐	☐	**Does the manual tell you all you need to know?** **What additional information would you like?**	
☐	☐	**Is the information accurate?** **(If not please specify with page number and paragraph.)**	

Name: _____ Title: _____

Company: _____ Division: _____

Address: _____ City: _____

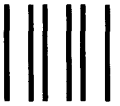
State: _____ Zip: _____ Telephone: _____ Date: _____

DG-06895

DataGeneral

Data General Corporation, Westboro, Massachusetts 01580

CUT ALONG DOTTED LINE



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

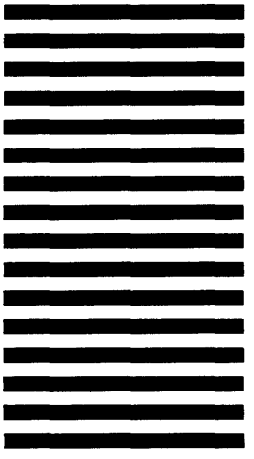
BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 26 SOUTHBORO, MA 01772

Postage will be paid by addressee

DataGeneral

Design Services
M.S. E-219
4400 Computer Drive
Westboro, MA 01581




014-000690-03