EK-70MP-TM-PRE

PRELIMINARY

PDP-11/70 MULTIPROCESSOR TECHNICAL MANUAL (PRELIMINARY)

PRELIMINARY

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PDP-11/70 MULTIPROCESSOR SYSTEM

The PDP-11/70 Multiprocessor System shown in Figure 1-1 consists of two modified KB11-C PDP-11/70 main processors (the PDP-11/74 Multiprocessor can have up to four KB11-E main processors), each one of which has independent access to a multi-ported common (tightly-coupled) MOS memory. Both processors are controlled by a single operating system, the RSX-11M+ that provides interaction between the processors, the programs they are executing and the I/O devices. The processors are all functionally equal with the ability to perform any task and function either as slave or master during program operations. Each processor has its own independent Unibus with peripheral devices connected to it.

Connected to each processor via its Unibus is an Interprocessor Interrupt Sanity Timer (IIST) Interface Module. The IIST modules are also connected to each other by the IIST independent Interrupt/Boot request Bus which passes Interrupt and Boot requests between them. The IIST sub-system is used by the RSX-11MP Executive to interrupt processors for the purpose of re-scheduling system activity. The IIST sub-system allows the operating program in one processor to send interrupts and boot requests via the IIST Bus and modules to any and all processors in the system. Execution of a task running on a processor can be pre-empted by the system software by means of the IIST sub-system whenever access to a device via that CPU is required. Only the interrupt or boot request is sent over the IIST Bus. The

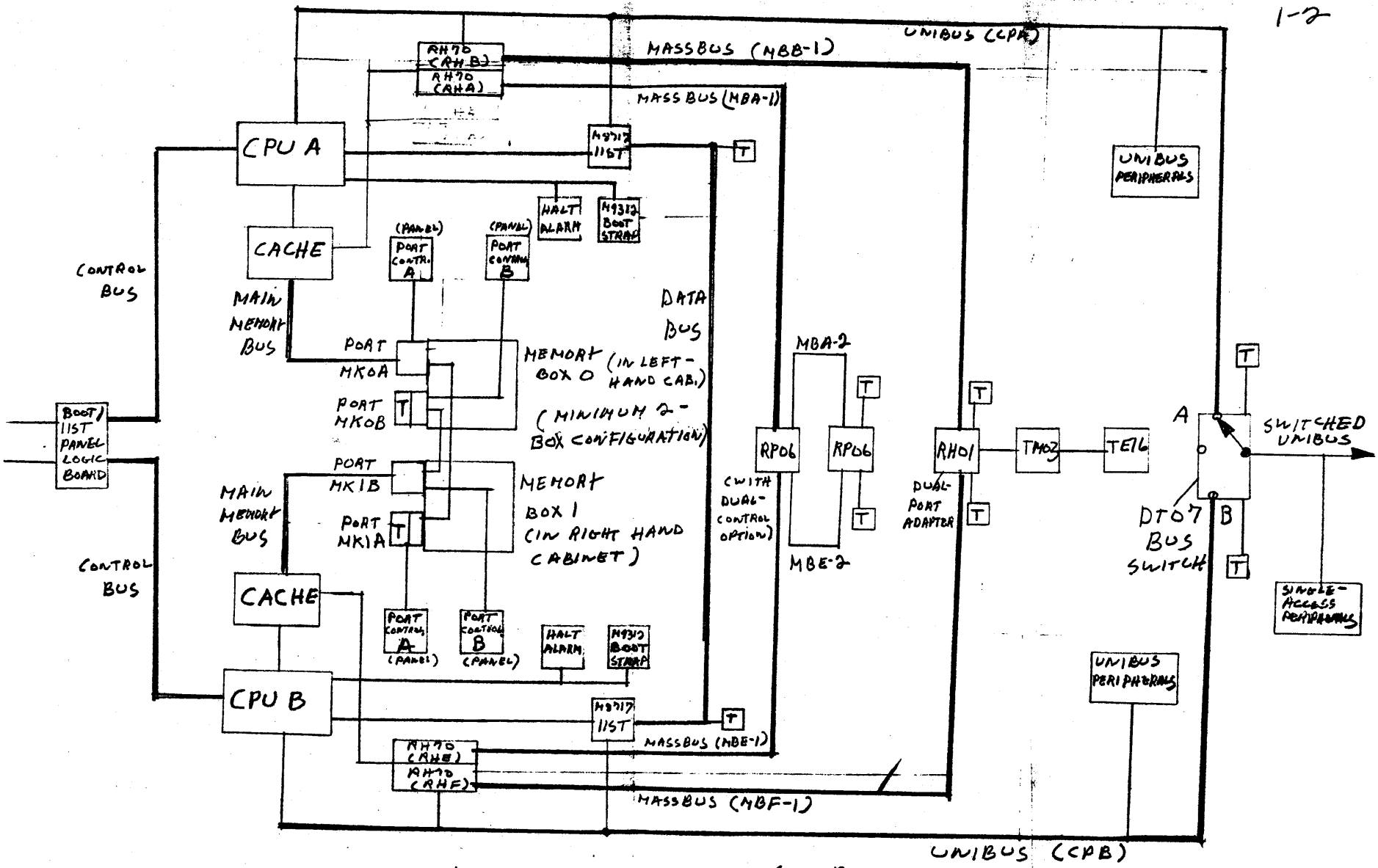


FIGURE 1-1 PDP-11/70 MULTIPROCESSOR (2-PROCESSOR) SYSTEM

actual interrupt handling service sequence involving the Interrupt signal and the interrupt vector address is accomplished via the Unibus as a normal interrupt service handling routine between the concerned processor and the IIST Interface Module receiving the Interrupt request.

Thus a task running on one CPU can request I/O service on any device regardless of where that device is located.* The IIST also provides a Sanity Timer that must be periodically refreshed by the program. If a timeout should occur before the timer has been refreshed, an interrupt and/or boot request can be transmitted to any or all of the processors (depending upon how the Sanity-Timer Transmission Enables Register is set) and the faulty processor is halted and its Unibus will be frozen or locked by the IIST asserting BBSY (Bus Busy), thereby, limiting the effect of a possibly faulty processor.

Both single ported and dual ported devices can be used. When required, single-ported devices can be replicated on different busses. Dual access massbus devices can be accessed by Massbus controllers on different massbuses. Dual port adapters and options are used to dual-port these devices.

In addition, single ported devices can be used on a shared bus and (optionally) switched for use with one processor or another on different Unibuses by means of the DT07 Bus Switch.

*Of course, requests for I/O and the servicing of interrupts must be performed by the processor to which the device controller is connected.

1.2 APPLICATIONS

The PDP-11/70 Multiprocessor supplies to the customer an increase in the capability of his system to perform its critical application (enhanced availability) by means of redundant functional units, and a system fault recovery capability that allows the operator to reconfigure around a failed element or system in order to recover the critical application part of his system.

It provides to customers significantly improved performance and functions in the areas of Commercial Data Processing. When coupled with properly designed applications, multiprocessors can provide a significant reduction in cost penalties resulting from critical downtime combined with substantial throughput improvements. It will improve customer's system performance providing the ability to execute tasks simultaneously and by efficient use of added resources (various configurations of multiprocessors and standalone processors can be used without interference of one with the other) during peak load handling.

Basically the multiprocessor system will provide the following capabilities:

1. Symmetrical multiprocessing

2. Use of Multiported memory

3. Use of dual-ported access massbus devices

4. Use of DT07 Bus Switch for use with shared Bus (optional)

5. Program-assisted reconfiguration of most units under operator control to allow powered-up fault isolation and maintenance

6. Improved Error Logging capability

7. Use of System Crash Analyzer

8. Ability to reconfigure system into independent systems

9. Ability to load stand-alone diagnostics into an off-line system from a Files-11 volume.

10. Ability to allow on-line processor diagnostics to be run by binding a task to a particular CPU (task-CPU affinity).

11. Ability to allow the operating system to be bootstrapped from any Files-11 device.

1.3 MULTI-PROCESSOR PACKAGING

The PDP-11/70 Multiprocessor System is packaged in Double-Width High-Boy Cabinets and in Single-Width High-Boy Expander Cabinets. Each individual processor and memory are contained in a Double-Width High-Boy (DWHB) cabinet which has two bays: the processor bay and the memory bay. Two of these DWHB cabinets (shown in Figure 1-2) are bolted together during installation. One is the left-hand processor, the other the right-hand processor. The left-hand processor has a left-mounted modified KB11-C and the right, a right-mounted modified KB11-C. One or two common memory boxes can be mounted on the memory side of

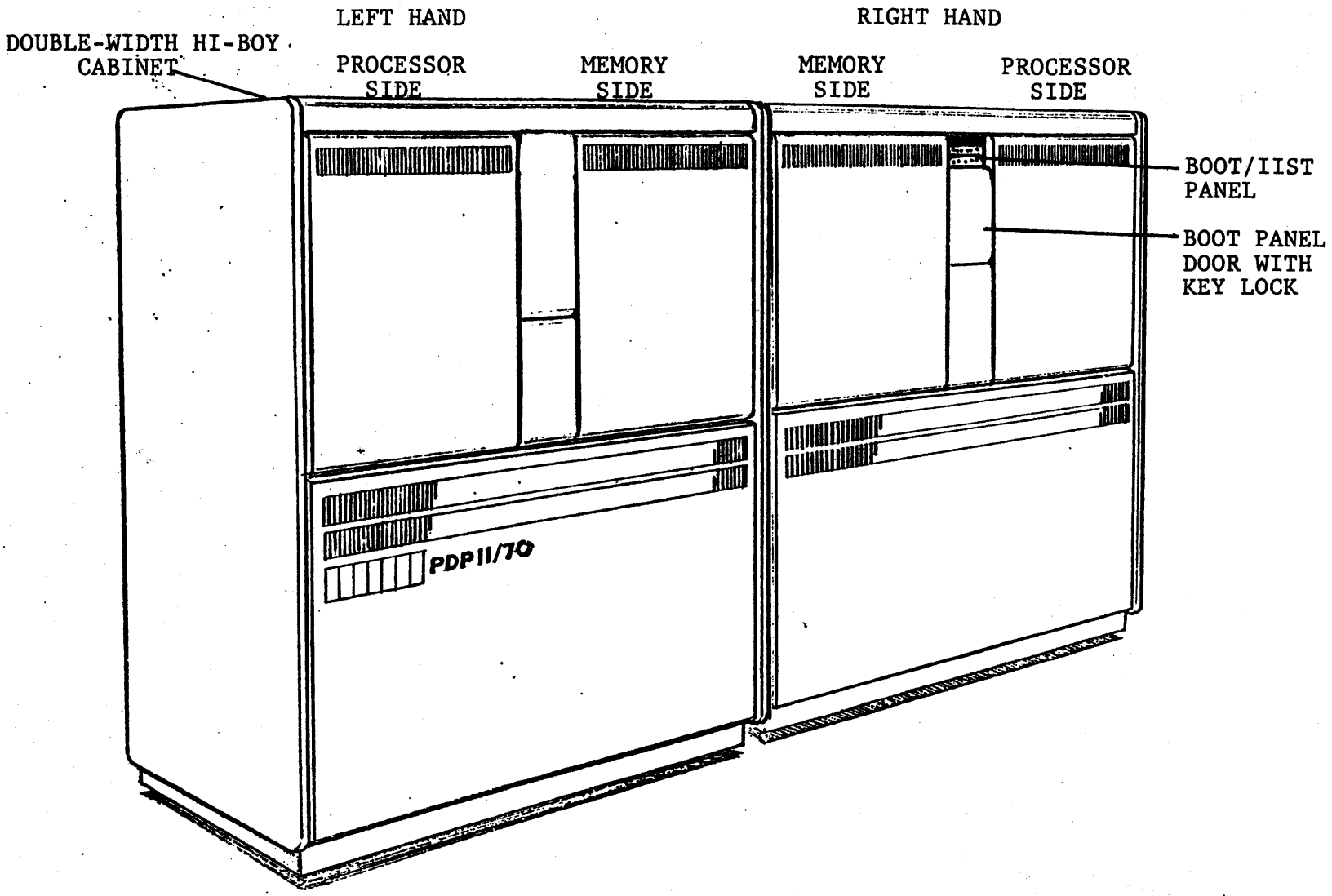


Figure 1-2 PDP-11/70 Multiprocessor (2-Processor) System

each processor. The right processor is a mirror image of the left one in that the location of components is reversed; left bay components to the right bay and right bay components to the left bay which puts the memory system bays of the different processors adjacent to each other when the DWHB cabinets are bolted together.

In addition, the PDP-11/70 multiprocessor system requires at least two single width-high boys Expander Cabinets, A and B, in which BA11-P boxes are mounted. Contained in each of the Expander Cabinets are the DIP11-A IIST interface module (M8717), the DD11-F system mounting unit (9-slots), and the various communication devices and switches (DZ11's, BSM-11, CSS-11, etc.). Expander Cabinet A is dedicated to CPU A, and Expander Cabinet B to CPU B.

Location of the major components contained in the DWHB cabinets is shown in Figure 1-3. Memory Port Controllers are located in the upper left portion on the processor side of the DWHB cabinet of the left-hand processor and in the upper right of the right-hand processor. There is one port controller for each port of the MKA11 memory. For the PDP-11/70 multiprocessor, up to four port controllers can be mounted on the processor side of each DWHB (assuming that two boxes of memory are contained on the memory side). Unused mounting locations on the panel are covered with blank panels. In the lower portion on the processor side of the DWHB cabinet are contained the CPU switch console or remote diagnostic console (optional), and in the BA11-F CPU mounting box which can be pulled out for easy board access are the modified KB11-C processor, the DL11-W Console Interface, the M9312 Bootstrap/Terminator Module, and the M9014 or M9302 Unibus terminators.

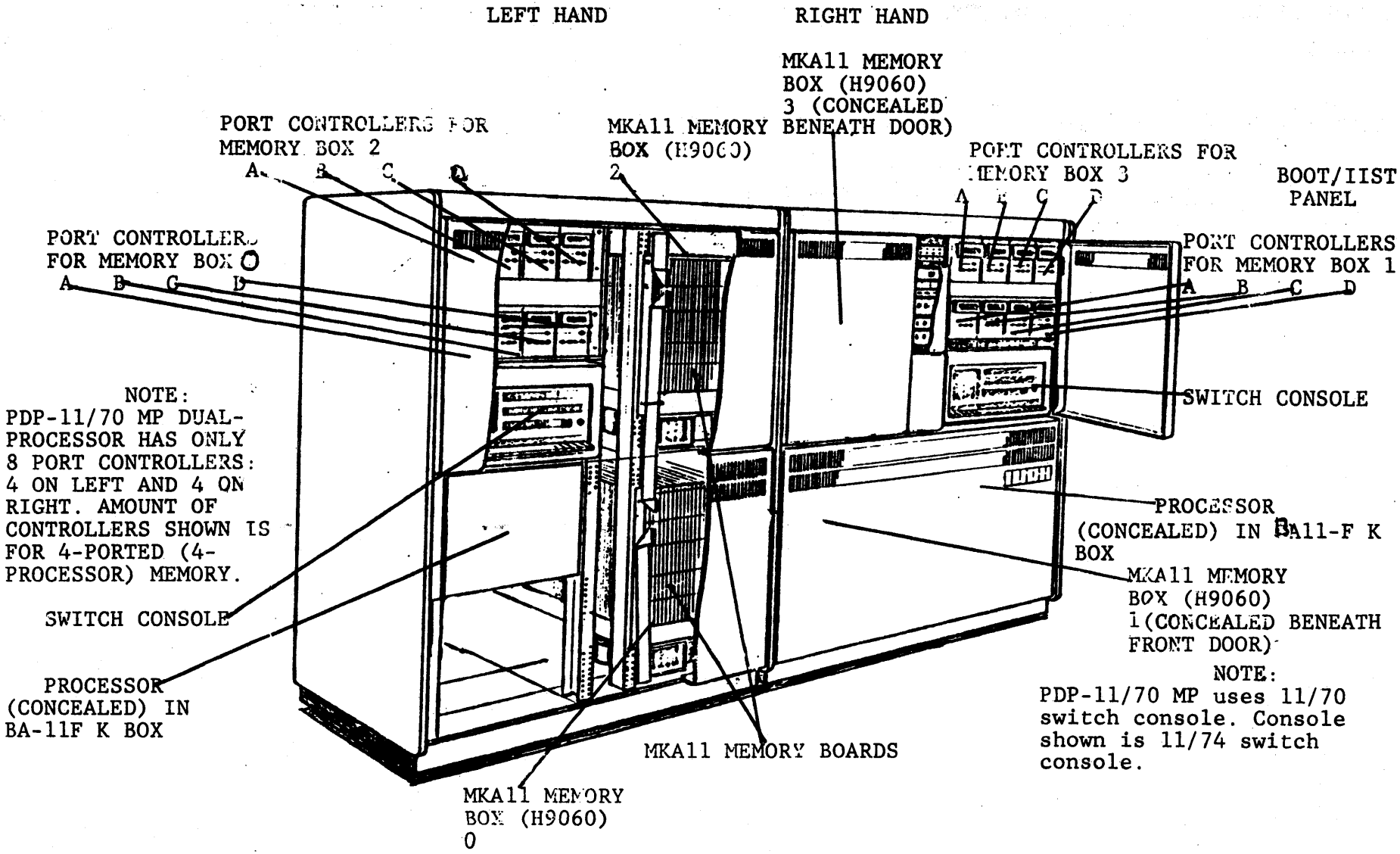


Figure 1-3 Major Components of Multiprocessor System (Front View--Sheet 1 of 2)

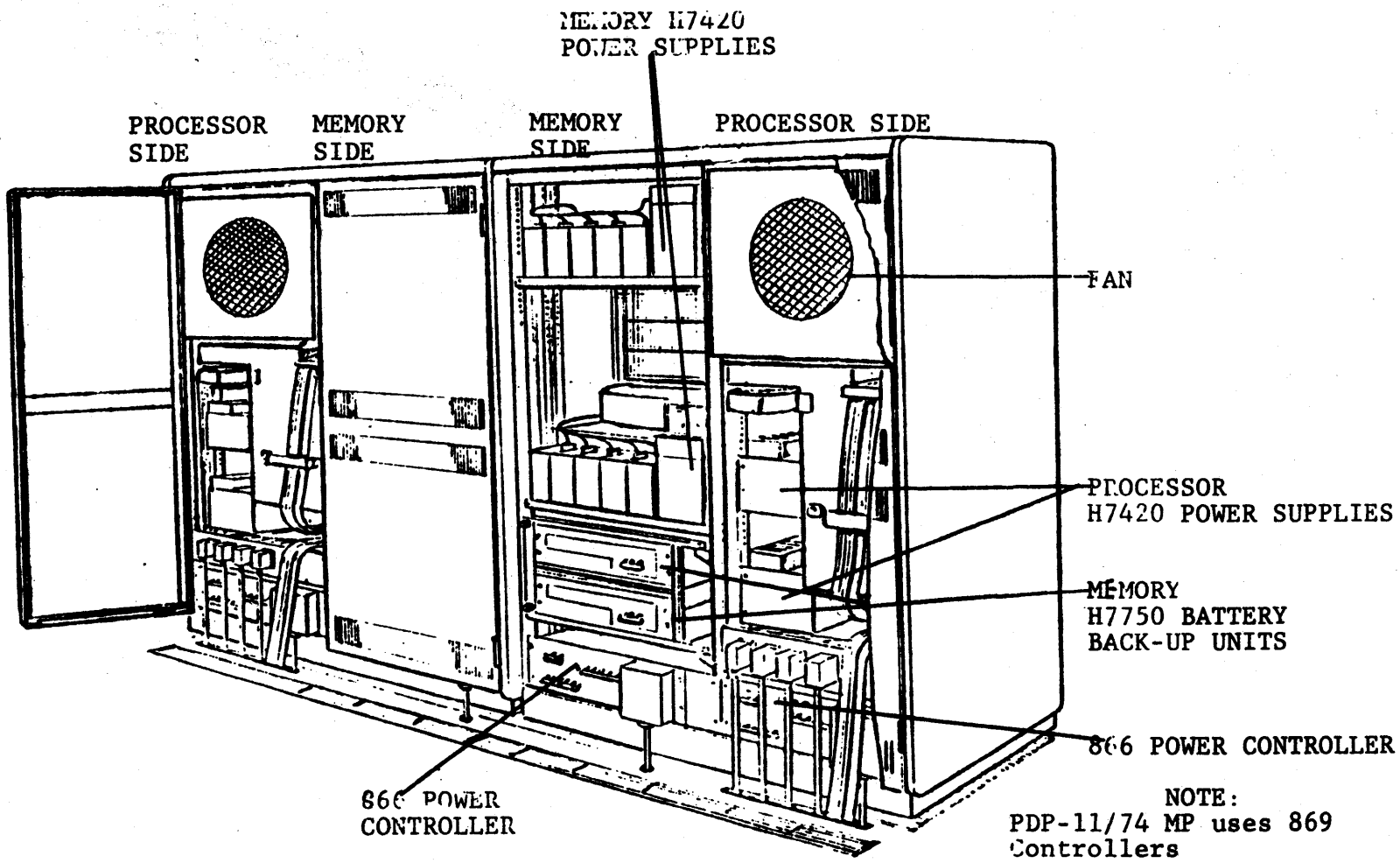


Figure 1-3 Major Components of Multiprocessor System (Rear View--Sheet 2 of 2)

NOTE: The M9302 will be installed in slot 44 during systems Manufacturing, and the DIP11-A will be in slot No. 42 during FA&T. At the customer site the M9014 will be installed in slot No. 44 and the DIP11-A (M8717) Interface Modules will be located in Expander Cabinets A and B.

Also located on the processor side of each DWHB are two processor H7420 power supplies and at the very bottom of the cabinet an 866D or E power controller.

The right bay or memory side of each DWHB can contain up to two H9060 memory boxes. One H9060 card cage contains the memory systems buffer, multiplexor, control, and storage array modules. These boards slide into the card cage on guides and plug into the memory backplane. Also contained on the memory side of each DWHB cabinet are the H7420 power supplies, the H7750, battery back-up units, and the 870 remote power control. Each memory box must have at least one H7420 power supply, one battery back-up unit H7750, and one 870 A or B power control. The 870 power control is connected to an 866 D or E Power Controller located on the memory side at the very bottom of each DWHB cabinet.

The BOOT/IIST Panel Switch Board (mounted behind a panel) and the BOOT/IIST Panel are located on the right-hand processor. (There is only one BOOT/IIST Panel and Panel Logic Board for each multiprocessor system.)

1.4 BRIEF FUNCTIONAL DESCRIPTION OF BASIC COMPONENTS

1.4.1 Modified KB11-C Processor

The basic KB11-C processor has been modified for use as the basic processor in a Multiprocessor System. Each modified processor's cache is connected via an independent main memory bus cable to an individual port on the MKA11 multiported shared memory. The operation of the modified KB11-C is identical to the KB11-C except for those functions required specifically for Multiprocessor operation:

1. Cache Flush (invalidates all the data in the cache when cache data no longer reflects the data in main memory.

2. Cache Bypass (Occurs when the cache is bypassed and all references are forced directly to main memory. Bypasses can occur under three different conditions and for different durations. The Bypass can occur continuously for short durations, or only at those locations associated with a particular Virtual page, or for locations mapped specifically by the Unibus Mapping Registers.

3. Ability to access Control Status Registers (CSR's) on main memory bus.

4. Memory timeouts are of longer duration in a modified KB11-C processor than in an unmodified one.

5. Use of ASRB instruction to implement a lock on shared

data locations in memory

1.4.2 MKA11-MEMORY

The MKA11 memory is a multiport MOS common main memory that can be shared by two, three or four CPU's.

NOTE: The PDP-11/70 and processor uses only 2 ports of the MKA11 memory. The PDP-11/74 multiprocessor can use up to four ports of the MKA11 memory.

Each processor's cache interfaces to one of the memory's four ports via independent main memory bus cables.

Each MKA11 memory has a storage capacity of 64K to 512 K 16-bit words. Larger main memory capacities are obtained by adding additional MKA11 memories to the system and daisy chaining the memory's ports on the main memory busses. Up to four memories can be daisy-chained in this fashion, expanding the shared main memory space to the maximum addressable 2M words.

Access to main memory is achieved through the memory ports with a multiplexor board that arbitrates within each memory box among contending CUP's and devices for access to memory.

All ports have equal priority. When one or more processors makes a request to a port, and the memory is ready to accept requests, an arbitration cycle is initiated. At the conclusion of the arbitration cycle, the requesting ports are serviced round-robin in sequence, starting with the port immediately after the last port to have accom-

plished a memory cycle.

The multiplexor permits instruction execution to be symmetrical and processor-independent. Concurrent access by different processors to the Executive shared data structure in memory is inhibited by a software locking mechanism.

Each memory box has its own independent power supply with power supply regulators. Every box also has a battery back-up unit that can provide up to five minutes of standby power for retaining memory integrity during loss of power or power down sequences.

1.4.3 INTERPROCESSOR INTERRUPT AND SANITY TIMER (IIST) SUBSYSTEM

The BOOT/IIST Panel Switch Board contains indicators that display the operating status of each processor, and switches that allow the operator to select and boot the various processors and the type of boot action that will or will not occur during power-up sequences. Switches on this panel also permit the operator to configure the processors to run as dual, tri, or quad multiprocessors in System A or B groupings or as standalone processors or any various combination of multiprocessor and standalone that the program requires. They also permit the operator to select any processor for ON or OFF LINE operation.

The BOOT/IIST Panel Switch Logic Board contains all the switch logic (controlled by the IIST CONFIGURATION Switches) for enabling or disabling the various drivers and receivers of the IIST Interface Modules of each processor so they can receive and transmit to other or to themselves when they are grouped properly in System A, System B, or

Standalone by the IIST Configuration switches.

1.4.4 IIST INTERFACE MODULE (M8717)

Each processor has an IIST Interface Module that is connected to all the other IIST Interface Modules by means of a Data Bus. Each module has its own data and clock line over which it transmits to the other modules (processors) any request for a boot or interrupt that it has received from its processor via the Unibus. It allows the operating program in one processor to send interrupt and/or Boot requests to any or all of the other processors in the system for the purpose of rescheduling activity on the various processors. For example, the RSX11MP Executive treats the different processors on different Unibuses as resource devices that it can use as it sees fit to perform all the various tasks required. There is also a Sanity Timer in the IIST Interface Module which must be refreshed periodically otherwise it will time out indicating a possible faulty processor.

1.4.5 MASSBUS CONTROLLERS AND DEVICES

Up to four RH70 Massbus controllers can be used with each CPU in the multiprocessor system. Only two controllers are shown with each CPU in Figure 1-1. Massbus devices used in a multiprocessor system are dual accessed by means of dual-access options or adapters. Dual access to the TE16 drive is accomplished by means of the RH01 dual port adapter which is used with the TM03 Formatter.

1.4.6 SINGLE ACCESSED DEVICES ON SHARED BUS

Single accessed devices can be connected to an optional switched Unibus by means of the DT07 Bus switch. Devices on the shared Bus

therefore can be accessed by two different processors at different times. In the event of a CPU failure, the Bus with its devices can be switched over to the other processor.

1.4.7 DB11-A BUS REPEATER

The PDP-11 Unibus is capable of handling 18 unit loads (including a DB11-A). The DB11-A Bus Repeater allows the Unibus to be extended beyond 20 Unit loads. And additional 18 unit loads can be attached to the Unibus using the bus repeater. The bus repeater does not affect the bus cycle time (it adds zero time) of any device before the DB11, and it adds a maximum of 375 ns to the cycle time of devices addressed beyond the bus repeater. It is a non-programmable device that interfaces two buses (left and right) in four sections:

1. Bus Repeater Control
2. Bus D (data) Lines
3. Bus A (address) Lines
4. Bus Unidirectional Lines.

1.5 FIELD TEST SYSTEM CONFIGURATION

Listed in Table 1-1 are the units making up the dual-processor multiprocessor (enhanced availability) Field Test System.

1.6 RECOMMENDED MINIMUM PDP-11/70 MULTIPROCESSOR CONFIGURATION (ENHANCED AVAILABILITY)

Table 1-2 lists the recommended components making up a minimum dual-processor system.

Table 1-1 Field Test System Dual-Processor Configuration

Quantity	Item	Description
1	PDP-11/70 Dual Processor System	consisting of:
2	KB11-C (Modified)	KB11-C Processor modified for multiprocessor operation
1	870 Power Control	Remote power controller used in memory power supply system.
2	MKA11-BA	Dualported multiprocessor MOS memory with Error Correction Coding and 256KB capacity per box (512KB total). Includes the following: 1) H9060 memory box 2) two sets of Port Buffer modules (M8158 and M8159) 3) two sets of Control Modules (M8161 and M8160) 4) two Port Controllers 5) one Multiport Multiplexor (M8162) 6) four 64KB MOS storage arrays 7) one H7420 power supply 8) one H7750 Battery Back-up Unit 9) Cable set

Table 1-1 Field Test System Dual-Processor Configuration

Quantity	Item	Description
2	H7420	Processor Power supplies
1	DIP11-C	BOOT/IIST Panel and Logic Board
2	M9312	Bootstrap Loader and Terminator
2	70-142211-24	Ruggedized Unibus Cables
2		Double-width Hi-boy Cabinets with 866A/E Controllers
2	LA36-HE	LA36CE and LAX-LG(EIA)
2	DL11-WC	DL11W with EIA Cable
1	RWP04-BA	2 RH70-A and RP04-BA (dual access-60 HZ)
1	RP04-BA	88MByte Disk pack and drive (dual access)
2	DIP11-A	IIST (M8717)
2	TWE16-AA	Magtape
2	BS11A-40	Remote Mag tape cable kit
2	BC037-25	Null Modern Cable
2		Single-width Hi-Boy expander cabinet
2	BA11-P	Expander Box (used in the SWHB Expander Cabinet)
16	DZ11	
1	M7258	Printer Control
1	DB11-A	Bus Repeater
2	DD11-DF	Peripheral Mounting Panel (9-slot)

Table 1-2 Minimum PDP-11/70 Multiprocessor Configuration (Enhanced Availability)

Quantity	Item	Description
1	PDP-11/70	Dual-processor consisting of: Double-Width Hi-boy Cabinets with 866A/D Controllers
2	H9602 BA	Expansion Cabinet (SWHB) with 866A/D controller
2	CPU Mounting Box	Installed in Double-width Hi-boy cabinets. Houses modified KB11-C processor, DL11-W, and M9312MP Bootstrap/Terminator for Module. It is also prewired for four RH70 controllers, and floating point processor.
2	Modified KB11-C	Processor modified for multiprocessor operation. Consists of the following:
	M8130 DAP Module	Data Paths (slot)
	M8131 GRA Module	General Registers and ALU control (slot)
	M8132 IRC Module	IR Decode and Condition Codes specially modified (slot 8)
	M8123 RAC Module	ROM and ROM Control (slot 9)
	M8134 PDR Module	Processor Data and Unibus Registers (slot 10)
	M8135 TMC Module	Trap and Miscellaneous Control (slot 11)
	M8136 UBC Module	Unibus and Console Control specially modified (slot 12)
	M8177 SAP Module	System Address Paths specially modified (slot 14)

Table 1-2 Minimum PDP-11/70 Multiprocessor Configuration (Enhanced Availability)

Quantity	Item	Description
	M8138 YA SSR Mo Module	System Status Registers (slot 15)
	M8139 TIG Module	Timing Generator (slot 13)
	M8140 SCC Module	System Descriptor/Console Cables (slot 16)
	M8181 MAP Module	Unibus Map (slot 22)
	M8182 CCB Module	Cache Control (slot 17)
	M8143 ADM Module	Address Memory (slot 18)
	M8144 DTM Module	Data Memory (slot 20)
	M8145 DCP Module	Cache Data Paths (slot 21)
	5411294 KNL Module	Console
2	MKA11-BA	Dual-ported MOS memory with Error Correction Coding. Consists of: H9060 Memory Box (with two pairs of Port Buffers, Multi-port Multiplexor, two pairs of Control Modules, and four 32K word MOS storage Arrays), and a H7420 Power Supply H7750 Battery Back-up Unit, and two Port Controllers. Port Controller contains control switches and indicators. One Port Controller controls one port on memory.
2	DL11W	Asynchronous Serial Line Interface with LA36
2	LA36 (or LA120)	DECWRITER 11 (300 baud printer terminal)

Table 1-2 Minimum PDP-11/70 Multiprocessor Configuration (Enhanced Availability)

Quantity	Item	Description
2	M9312	Bootstrap/Terminator Module with special MP ROM Code
4	RH70	Massbus High-Speed I/O Controller. Up to four RH70 Massbus controllers can be installed in prewired slots in individual CPU's backplane. Controller A in slots 24-27 Controller B in slots 28-31 Controller C in slots 32-35 Controller D in slots 36-39
	Each controller consists of the following:	
	M5904 MBS Module	Massbus transceivers (3 required) (slots 25-27, 29-31, 33-35 or 37-39, rows A-F).
	M8151 CST Module	Control and status (slots 25,29,33 or 37, rows C-F).
	M8152 AWR Module	Word Count and address (slots 26,30, 34 or 38, rows C-F).
	M8153 BCT Module	Unibus and register logic (slots 27, 31, 35 or 39, rows C-F). Each RH70 can control up to eight devices which must be of the same type.
2	RP05/6	Disk Drives (with dual-port option)

Table 1-2 Minimum PDP-11/70 Multiprocessor Configuration (Enhanced Availability)

Quantity	Item	Description
1	RH01*	Dual-port adapter. Used for dual-porting TE16.
1	TM03*	Tape controller (Used with TE16 and RH01)
2	TE16	Tape Transport
2	DIP11-A	Interprocessor Interrupt and Sanity Timer Interface Module (M8717)
1	DIP11-C	Consists of Boot/IIST Panel Switch Board (Only one per multiprocessor system) and Boot/IIST Panel Logic Board (Only one per multiprocessor system)
2	DD11 C or D	Peripheral Mounting Panel (System Unit)
2	H7420 Power Supplies	Refer to Figure 1-1. Two H7420s contain the dc power regulators for the CPU mounting box.
14	H7441+5 Regulators	Eight H7441+5 Vdc regulators supply dc power. An additional H7441 is required if floating point processor is installed.
2	5411086 Modules	Each H7420 contains one 5411086 module which monitors the ac input voltage in addition to its function as a 15 Vdc regulator.

*Optional

Table 1-2 Minimum PDP-11/70 Multiporcessor Configuration (Enhanced Availability)

Quantity	Item	Description
4	866D (115 V) or 866E (230V) Power Control	Controls ac power input.

Table 1-3 MKA11 Memory Options

Designator	Description
MKA11-AA (For 115 Vac)	Single Port Memory, 128K words Includes: H9060 Memory Box with one pair of Port Buffers, Single Port Jumper Card, two pair of Control Modules, and four 32K word MOS Storage Arrays, a Power Supply, Battery Back-up Unit, Port Controller, and Cable Set
MKA11-AB (For 230 Vac)	Single Port Memory, 128K words same as MKA11-AA, except for 230 Volt or 60 Hz operation
MKA11-AC (For 115 Vac/50 Hz)	Single Port Memory, 128K words same as MKA11-AA, except for 115 volt 50 Hz operation
MKA11-AD	Single Port Memory, 128K words same as MKA11-AA, except for 230 Volt 50 Hz operation
MKA11-AY	Single Port Memory Box, 64K words Includes: H9060 Memory Box with one pair of Port Buffers, Single Port. Jumper Card, two pair of Control Modules, and two 32K word MOS Storage Arrays
MKA11-BA	Dual Port Memory, 128K words Includes: H9060 Memory Box with two pair of Port Buffers, Multiport Multiplexor, two pair of Control Modules, and four 32K word MOS Storage Arrays; a power

Table 1-3 (cont) MKA11 Memory Options

Designator	Description
	Supply, Battery Back-up Unit, two Port Controllers, and Cable Set
MKA11-BB (For 230 VAC)	Dual Port Memory, 128K words. Same as MK11-BA, except for 230 volt 60HZ operation.
MKA11-BC (For 115 VAC/50HZ)	Dual port memory, 128K words, same as MK11-BA, except for 115 volt 50HZ operation
MKA11-BD (For 230 VAC/50HZ)	Dual port memory, 128K words, same as MK11-BA, except for 230 volt 50HZ operation.
MKA11-BY	Dual port memory box, 64K words, Includes: H9060 memory box with two pairs of port buffers, Multiport multiplexor, two pair of control modules, and two 32K word MOS storage arrays.
MKA11-UA	Single port upgrade Kit, upgrades single port memory (MKA11-AA through-AD) to dual port memory. Includes: one pair port buffers and one multiport multiplexor.
MKA11-UB	Multiport upgrade kit, adds another port to existing dual-port memory (MKA11-BA through BD) or triport memory. Includes: one pair of port buffers.

Table 1-3 (cont) MKA11 Memory Options

Designator	Description
MS11-KE	32 word Storage Array Module, implemented with 4K MOS RAM chips
M8158	Port Address Buffer Module
M8159	Port Data Buffer Module
M8160	Control A Module
M8161	Control B Module
M8162	Multiport Multiplexer Module
M8163	Single Port Jumper Card

CHAPTER 2

SYSTEM TECHNICAL DESCRIPTION

2.1 SCOPE AND PURPOSE

This section describes the system theory of operation of the PDP-11/70 Multiprocessor System. Its purpose is to give the reader an understanding of the operation of the Multiprocessor System at the system-level by means of a system functional overview and description in order to assist him in installation, maintenance, and troubleshooting. More detailed information concerning the operation of each one of the components making up the Multiprocessor System can be found in the appropriate field maintenance component manuals listed in Appendix C, a List of Related Documents.

2.2 PDP-11/70 MULTIPROCESSOR CONCEPTS

Multiprocessors have from two to four central processors (the PDP-11/70 MP has two processors, the PDP-11/74 can have up to four processors). The Multiprocessor system uses a common (tightly-coupled) multiported memory. Each processor has access to the memory by means of an individual port. The memory has four ports and, therefore, can be connected to four different processors via these ports. All the processors are controlled by a single operating system, the RSX-11 MP that provides interaction between the processors, the programs they are executing, and the I/O devices.

connected through the Boot LIST Panel logic board to the Unibus Map Board of each processor. The Unibus Map Board is connected to the Halt Alarm of each processor, the M9312 Bootstrap Terminator Module, and to the IIST Interface Module on the IIST BUS. The IIST Interface Modules of all processors are all connected to each other by means of the IIST Bus.

2.4 SIMPLIFIED OVERVIEW OF MULTIPROCESSOR OPERATION (NON-CACHED)

Assume that one task is currently active on each processor shown in Figure 2-1: Task A on Processor CPU A and Task B on Processor CPU B (to simplify this initial explanation the effect of the cache has been ommited). When Task A issues a non-I/O directive, the EMT (Emulator Trap) occurs on processor CPU A. CPU A enters kernel state and is vectored to the Executive code that handles synchronous traps. Since no I/O is performed, the Executive carries out the directive and returns control of the processor to Task A. Execution of Task B on Processor CPU B is uninterrupted.

When Task A issues a request for I/O, processor CPU A enters the kernel state and is vectored to the Executive code that handles synchronous traps. If the device can be accessed from CPU A, i.e., it is located on CPU A's Unibus, the driver is called immediately to perform I/O. Task A can suspend execution until I/O completes thus freeing CPU A for use by another task.

If the device resides on the Unibus of Processor CPU B then the Executive transfers control to CPU B by generating a program interrupt request via the Unibus to CPU A's Interprocessor Interrupt and Sanity Timer

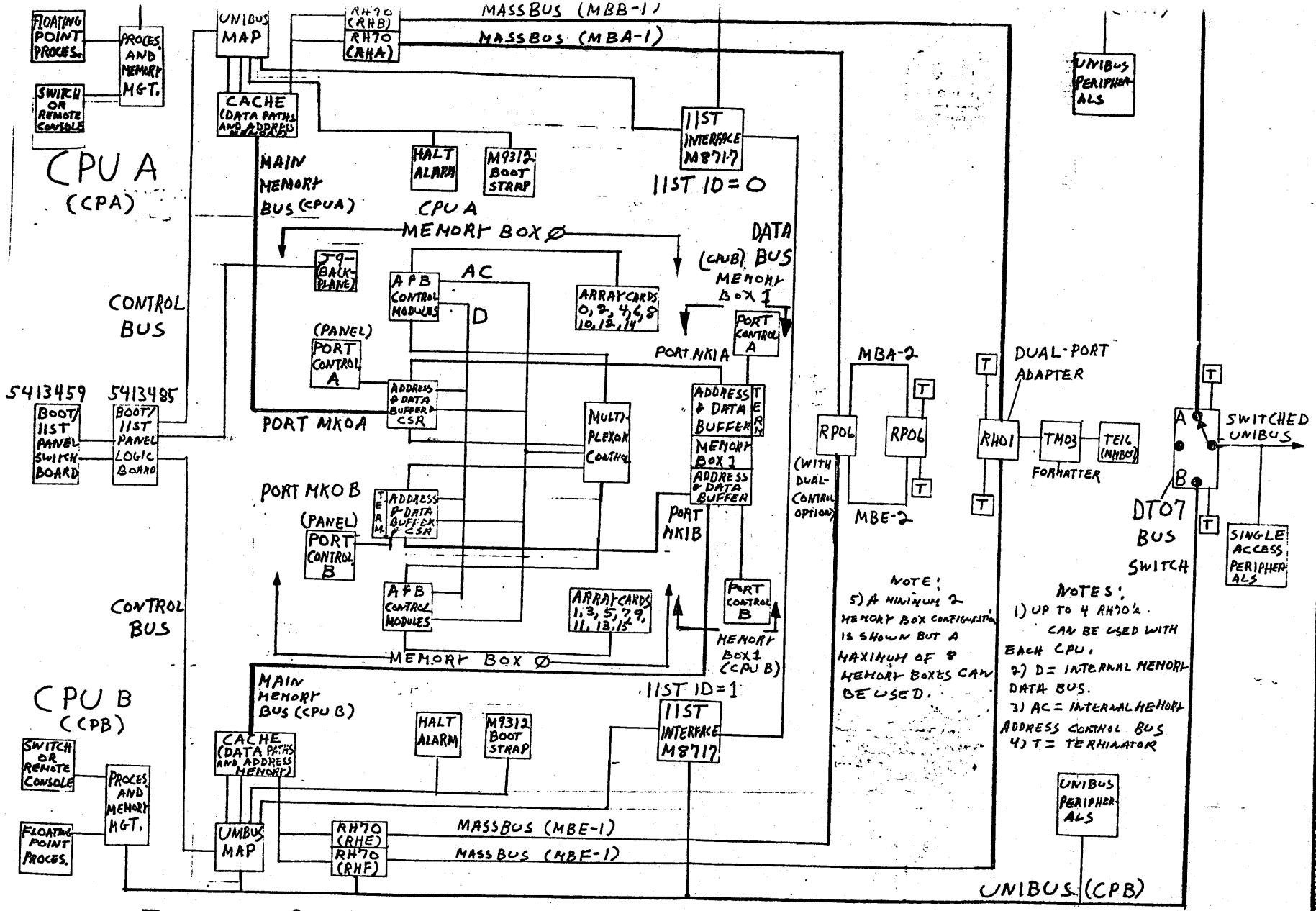


FIGURE 2-1 TYPICAL PDP-11/70 MULTIPROCESSOR SYSTEM (2-PROCESSOR CONFIGURATION)

(IIIST) module where it is loaded into a shift register and then clocked out over the serial data lines of the IIST Bus to be received by CPU B's IIST Interface Module. From there, it is sent via CPU B's Unibus to the CPU B Processor (the vector address that is sent to the CPU B Processor along with the Interrupt signal is determined by Vector Address Selection Switches on its IIST Interface Module.) The Executive has now transferred control to CPU B and released CPU A for use by another task. The Executive, now running on Processor CPU B calls the driver to initiate the I/O transfer with the device located on CPU B's Unibus. As in the single processor system, the allocation of CPU's to tasks is interrupt driven. When the I/O operation is completed, the Executive will scan the list of active tasks, searching for the highest priority tasks that can run (including the null task). On each scan of the list, the Executive will start unblocked tasks until all processors are busy.

When different processors are attempting to access the Executive at the same time, the contention is resolved by a single software lock that blocks access to the Executive by all but one CPU. Other processors will assert lock requests until the owning processor opens the lock and returns to user state. Contention involving the Executive is, therefore avoided by forcing all CPU's to accessthe Executive data base in serial fashion.

2.5 SIMPLIFIED OVERVIEW OF MULTIPROCESSOR OPERATION (WITH CACHE)

In the Multiprocessor System the cache of each processor is independent and unsynchronized and is connected by means of independent main memory bus cables to an individual port on the multiported MKA11 common memory. Multiple copies of the data in main memory are not always maintained

in all the caches. To avoid, therefore, the problem of using data in a processor's cache that does not reflect the current data in main memory, cache flush and cache bypass operations have been added to the modified KB11C Processor. These functions permit the contents of the cache to be invalidated or the cache to be bypassed for certain accesses when the cache at that moment no longer reflects accurately the data in main memory. In a cache flush the entire contents of a Processor's cache is invalidated; during a Cache Bypass the Cache is bypassed and all references are forced to main memory.

In addition to the functions listed above a software locking convention has been added to the processor by adding an uninterruptible read-modify-write (DATIP-DATO) memory cycle to the MKA11 memory and modifying the ASRBinstruction to bypass the cache, initiate the DATIP-DATO memory cycle, and implement a lock by the use of the Carry (C) bit which is used to determine and reflect the condition of the lock byte at the shared data location. During potential cached operations, Cache Bypass can be performed at different times in the Multiprocessor System. Bypass of the Cache can be uninterrupted, i.e. quickly turned on and then off for short sections of code. Uninterrupted cache bypass is required when short segments of code, such as interrupt service routines must have guaranteed access to main memory.

Cache Bypass is also, of course, required when two or more parallel processes are accessing a common data base. In this case the effect on system performance caused by bypassing the cache is reduced by confining the virtual address space over which cache bypass occurs to the region containing shared data.

When a process is allowed to run serially on all CPU's, total cache invalidation is necessary by means of a cache flush. Because all con-

text switching is performed by the Executive, the cache flush occurs without task intervention. A system-initiated cache invalidation procedure occurs as follows:

1. Read location X on CPU A.

2. Flush the CPU A Cache.

3. Write location X on CPU B.

4. Flush the CPU B Cache.

5. Read location X on CPU A.

The flush of CPU A's cache performed during the initial transition from CPU A to CPU B ensures that the next reference to location X from processor CPU A will correctly access main memory.

Invalidation of the cache's contents by means of a cache flush will also occur if a match should occur between an I/O memory address and a cache tag address when I/O traffic is being sent to main memory.

2.6 INTERPROCESSOR INTERRUPT AND SANITY (IIST) SYSTEM

The IIST Multiprocessor System consists of the following:

1. IIST Interface Module (Each processor has its own M8717).

2. BOOT/IIST Panel Switch Board (Each Multiprocessor System has only one.)

3. BOOT/LIST Panel Logic Board (Each Multiprocessor has only one.)

4. Unibus Map Board (Part of each CPU System, but interfaces with the IIST Interface Module).

5. Cable Assemblies

   5A. IIST Bus Cable (Connects all IIST Interface modules. Each multiprocessor has one.)

   5B. BOOT/LOCK/CONTROL 20-conductor ribbon Cable. One for each processor. Connects between M8717 and M8181.)

   5C. BOOT/HALT Alarm 8-conductor ribbon Cable (Connects between M8181, the Halt Alarm and the M9312. One for each processor.)

   5D. BOOT/IIST Panel Switch Cables (Connect between the BOOT/IIST Panel Switch Board and the BOOT/IIST Panel Logic Board. Each Multiprocessor has two.)

   5E. BOOT/IIST Panel Logic Board Cable (Connects between the IIST Panel Logic Board and the Unibus Map Board of each processor. (Each processor has one.)

   5F. Memory Boot Cable (Connects between the BOOT/IIST Panel Logic Board and the J9 connector on the backplane of the first box of memory on the system or memory 0).

2.6.1 Major Purpose

The major purpose of the IIST subsystem (IIST Interface Module M8717 and the IIST Bus) is to allow the operating program in one processor to send interrupt and boot requests to any or all of the other processors, in the system by means of the IIST Interface Module and the IIST Bus for the purpose of rescheduling activity on the various processors. (The actual Interrupt signal and vector address is sent via the Unibus to the selected Processor by the IIST Interface Module that received the request over its IIST Bus.) In addition a Sanity Timer is included in the IIST Interface Module which, if not refreshed periodically by the operating program within a programmable timout period will cause a Sanity Timer generated Interrupt or Boot request to be transmitted to any or all of the processors in the system, thereby notifying them of a possible faulty processor.

2.6.2 BOOT/IIST Panel Switch and Logic Boards

The BOOT/IIST Panel Switch (together with the BOOT/IIST Panel Logic Board) is used to perform the following functions:

1. Displays Power and operating Status of each of the processors (POWER, RUN, OR HALT)

2. Allows operator to manually boot any of the processors.

3. Allows operator to select type of BOOT action occurring on CPU POWER-UP action sequence (Boot, Run or Boot if battery depleted, or Run and halt if battery depleted)

4. Permits operator to partition the processors for operation as

dual, tri or quad multiprocessors in A or B groupings or as Standalones.

5. Permits the operator to place any Processor in ON or OFF line mode by disabling or enabling its IIST Interface Module.

2.6.3 IIST Interface Subsystem

The IIST Interface subsystem consists of from two to four IIST Interface modules (M8717) depending upon the number of main processors, connected to each other by means of the IIST Bus cable. (Refer to Figure 2-2, which shows a simplified block diagram of the IIST Interface Subsystem using a four-processor configuration).

Each IIST Interface module is a hex-height module that plugs into a small peripheral Controller (SPC) slot in a DD11-C or equivalent backplane. Each module is connected to the Unibus through the backplane. The configuration of the IIST interconnecting Bus Cable depends upon the number of IIST Interface Modules (number of processors) present in the system and the distance between them. One IIST Interface Module is required for each Processor.

2.6.4 Summary of Major Functions

The IIST Interface Subsystem performs or provides the following functions: functions:

a. Allows the operating program in one processor to send interrupt and Boot requests to any or all processors.

b. Provides a Sanity Timer that can time out and generate

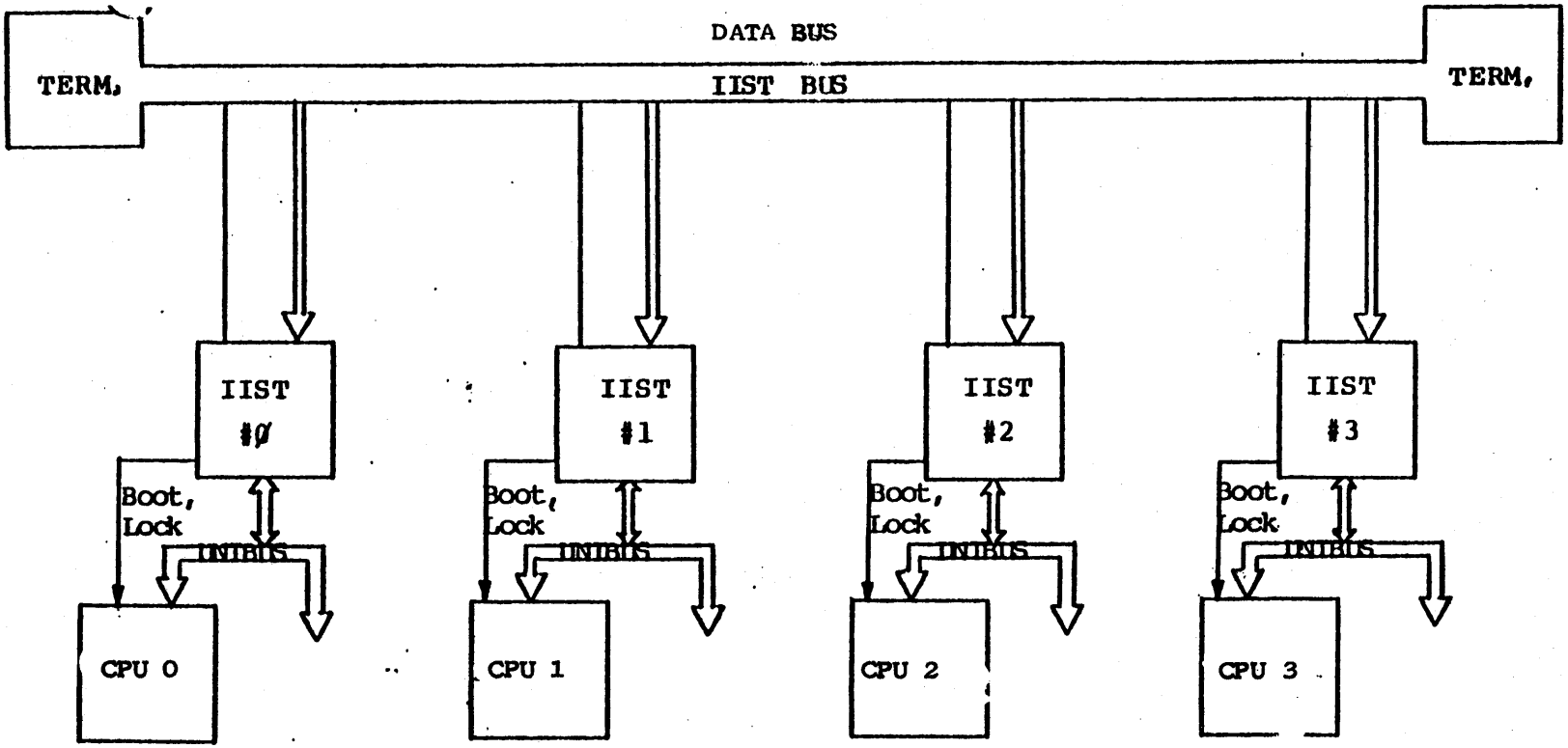


FIGURE 2-2.

IIST INTERFACE SUBSYSTEM BLOCK DIAGRAM

interrupt or boot requests to any or all processors

c. Provides for locking up or preserving the state of a faulty processor when its Sanity Timer times out

d. Contains Processor (IIST Interface Module) identification switches

e. Allows a processor to accept or ignore interrupt and/or boot requests

f. Contains a DCLO/Disconnect detection circuit

g. Provides for Error detection by means of parity bits

h. Includes On and Off line maintenance capabilities

Further details of these major functions are contained in the following paragraphs.

Shown in Figure 2-3 is a Multiprocessor IIST subsystem signal flow block diagram using a two processor configuration.

2.6.5 Program-Generated Interrupts and/or Boots

An operating Program in one processor causes interrupt and/or boot requests to be sent to other processors in the system by first setting transmit enable bits in an IIST Interface Module internal device register specifying which processors are to receive an interrupt and/or boot request and then setting a GO bit which starts transmission of a serial character on the IIST Bus to the selected IIST Interface

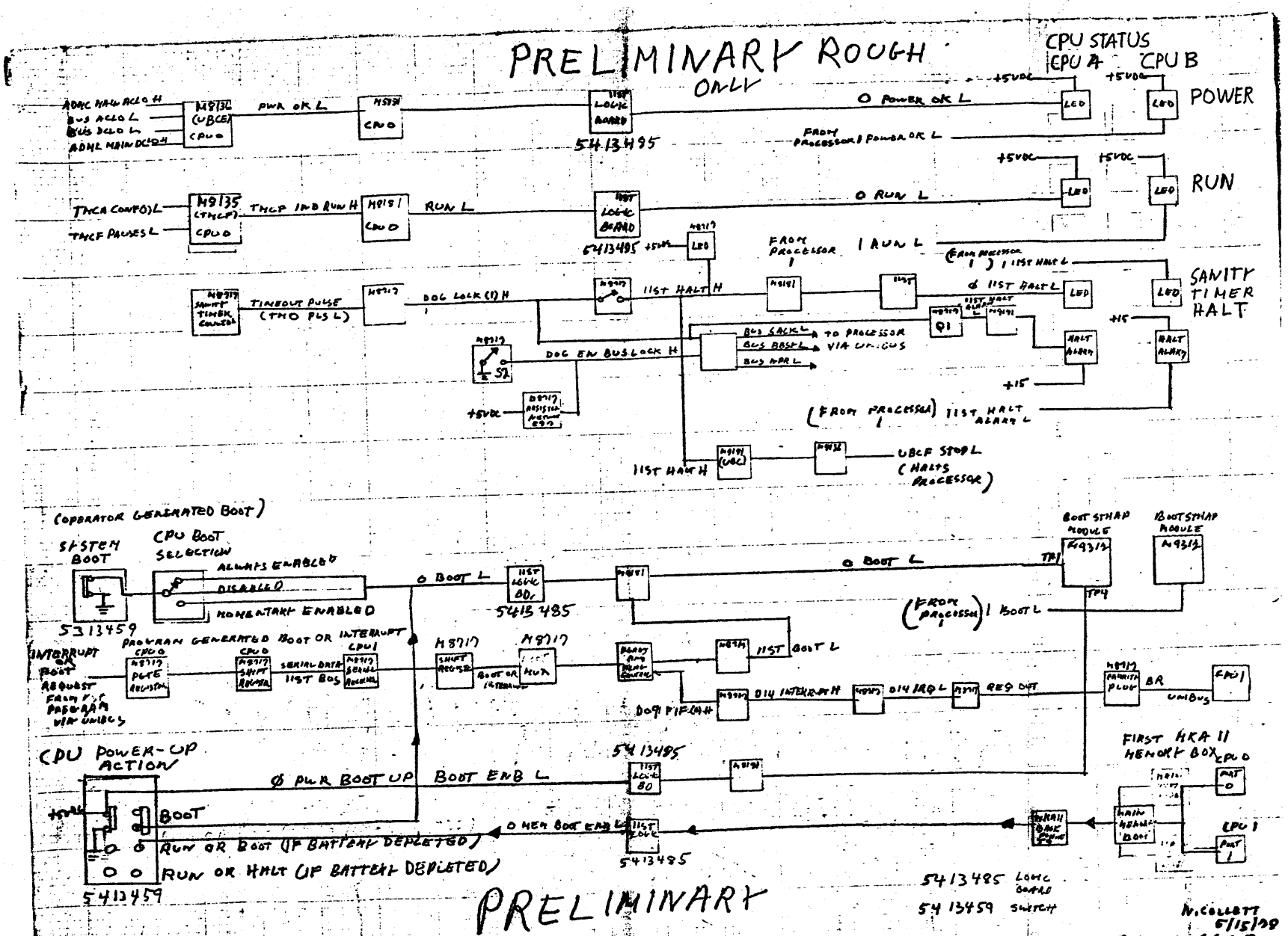
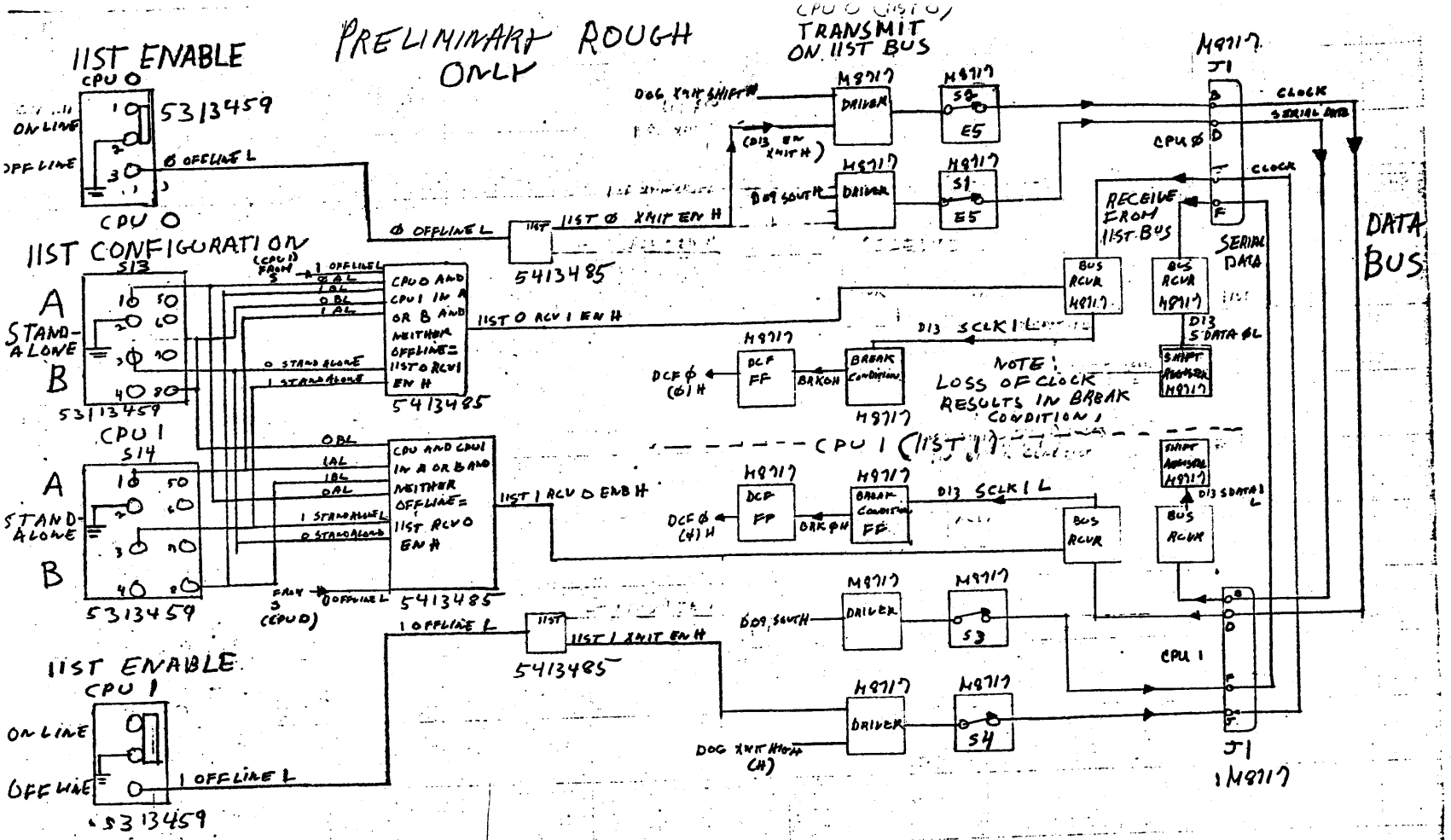


FIGURE 2-3 PDP-11/70 MULTIPROCESSOR 1IST SUBSYSTEM (2-PROCESSOR (SHEET 1 OF 2)



PRELIMINARY

FIGURE 2-3 PDP-11/70 MULTIPROCESSOR 11ST SUBSYSTEM SIGNAL FLOW (2-PROCESSOR, SHEET 2 OF 2)

N. COLLETT 5/16/78

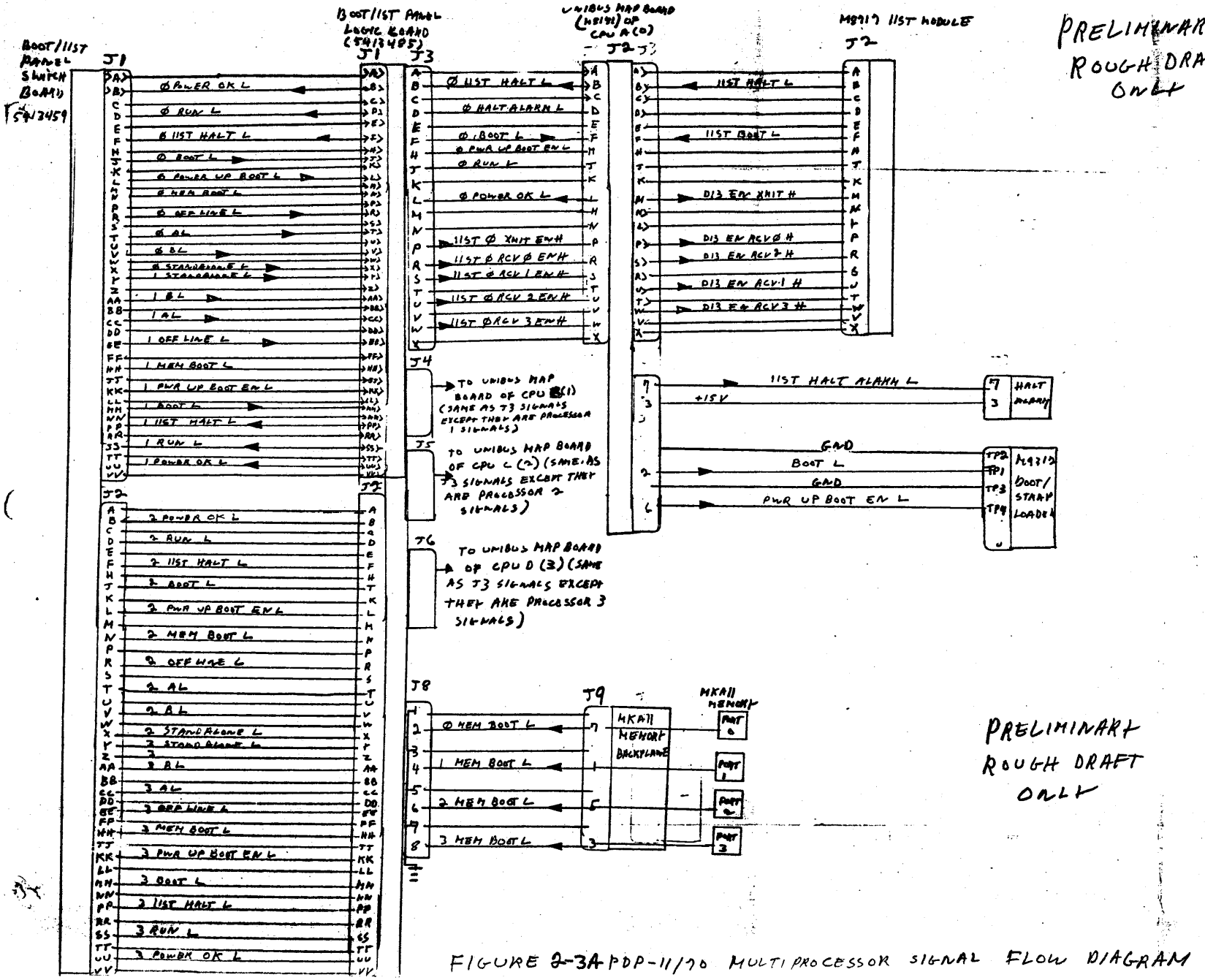


FIGURE 2-3A PDP-11/70 MULTIPROCESSOR SIGNAL FLOW DIAGRAM

Module. This is the primary function of the IIST sybsystem and is used by the operating system to re-schedule system activity on the various processors. Pre-empting task execution on a processor for I/O service is an example of rescheduling system activity.

A boot request properly received and enabled boots the receiving CPU by triggering its M9312 bootstrap module. (The M9312 is electrically connected through the Unibus Map Board to the IIST Interface Module by means of the Boot/Lock/Control Cable.) This boot function is used by the multiprocessor system boot code to permit the first CPU that has been booted to bring up the other CPU's in the system.

There are eight enable bits; four to specify processors selected to receive interrupt requests, and four to specify processors selected to receive boot requests. Therefore, in a single operation any or all processors may be interrupted and/or booted. One processor can selectively ignore interrupt or boot requests from another processor by setting mask bits in the IIST interval device masking registers.

2.6.6 S Sanity-Timer Generated Interruptsand/or Boots and Processor Lock-Up

Under normal operation the Sanity Timer contained in the IIST Interface Module will be periodically refreshed by the Executive software if, however, the refresh commands to not occur within the programmed period, the IIST will issue an interprocessor interrupt to inform the other processors in the On-Line system of the faulty processor operations.

In addition a system lockup will be generated so that the state of

the processor can be frozen upon expiration of the sanity Timer thereby preserving the processor state and limiting the effect of a faulty processor. The Lockup condition is held until an interrupt or boot request is received by the IIST Interface Module. The method of lockup can vary since it is switch selectable and will depend upon the settings of the two switches on the IIST Interface Module. The Lockup can be accomplished by holding the Unibus of the faculty Processor or by directly halting the processor itself through a lock line.

2.6.7 Processor Identification Switches

The processor identification switches (IIST Interface Module) contained on the IIST Interface Module can be read by the operating program to facilitate CPU-specific operations and to facilitate fault isolation during troubleshooting and maintenance. They also provide unique identification of the source and destination of all interrupt and boot requests. They, therefore, can be thought of as the CPU number, even though they are switches on the IIST Interface Module and determine the identification of the IIST Interface Module.

2.6.8 Masking of Processors Against Interrupts and/or Boots

Mask Registers in the IIST Interface Module can be programmed so that a processor can selectively accept or ignore interrupt and/or boot requests from other processors in the system. This capability is used by the reconfiguration software to isolate one (or more) of the processors that have been established as an independent or standalone system.

For example, a tri-processor multiprocessor system could be masked by use of the mask registers into an off-line thread for troubleshooting purposes with one processor while leaving the other two processors grouped together on-line and running as a dual-processor multiprocessor. (Of course the same division could be manually accomplished by the operator at the BOOT/IIST Control Switch Panel by use of the IIST CONFIGURATION switches.)

2.6.9 DCLO/Disconnect Detection

A DCLO/Disconnect detection circuit in the IIST Interface Module is used to notify a processor when other IIST Interface Modules in the system are either powered down or disconnected from the IIST interconnecting Bus. The interconnecting IIST Bus is T-connected to permit an IIST Interface Module to be disconnected or powered down without disturbing the remaining IIST Interface Modules on the IIST (Data) Bus.

2.6.10 Error Detection

The IIST Interface subsystem provides error detection by means of parity bits transmitted along with interrupt and boot requests on the IIST interconnecting Bus. If an error is detected, an error bit identifying the transmitting IIST Interface Module is set but no normal interrupt or boot action is taken.

2.6.11 ON/OFF Line Maintenance Capability

ON or OFF Line maintenance and checkout is accomplished by first running the single-interface portion of the IIST diagnostic program, YM-Z095A with the other processors halted and initialized, and then the Multi-interface Exerciser portion of it on all processors in the

system simultaneously for at least fifteen minutes. The results of the diagnostics must be completely error-free to indicate that the IIST Interface Subsystem is operating properly.

2.6.12 IIST Interfaces

2.5.12.1 IIST to Unibus

The Unibus provides the interface between a PDP-11/70 Processor and its IIST Interface Module. All data transfers to or from the IIST to the processor are accomplished under program control through IIST device registers residing in the Unibus peripheral address space. The Unibus also provides the path by which the IIST may interrupt and send the Vector Address to the processor.

The IIST Interface Module functions as any standard device on the Unibus. Addresses, data, and control information are sent along the standard lines of the Unibus. Communication between the processor and the IIST Interface Module is like any two standard devices on the Bus and takes the form of a master-slave relationship with the IIST capable of being either masteror slave. Priority of a particular IIST Interface Module is established by a priority plug in the IIST Interface Module. The IIST can have a priority level of anywhere from BR4 - BR7 or when generating a Sanity Timer Halt it can generate an NPR to hold the Bus, if certain option switches have been enabled. Horizontal priority is established, like any standard Unibus device, by its location on the Bus.

Two 16-bit Unibus registers are employed to interface the IIST to the PDP-11/70 Processor. These registers act as an access path to the ten

internal device registers of the IIST Interface Module which may be read and loaded under program control to monitor IIST status and to control operation

2.6.12.2 Program-Generated Interrupt Transmission (IIST to Processor via Unibus)

Shown in Figure 2-4 is the Interrupt Handling sequence that occurs between the IIST and the processor.

When the IIST Interface Module receives an interrupt request and the Interrupt Enable (INTR ENB) signal is set, then an interrupt request is asserted. As a result of the presence of the interrupt request and the INTR ENB signal, a Bus Request (BR) is asserted on the Unibus to the processor by the IIST Interface Module. The priority level of the BR is determined by a priority plug in the IIST Interface Module.

The processor responds with a Bus Grant (BG). Since an interrupt is being requested by the IIST Interface Module, the BG is blocked and Bus Sack is asserted by the IIST. The processor then clears BG, BBSY, and MSYN (froma previous operation). When the IIST sees that BG, Bus BBSY, ane MSYN are all cleared, it becomes bus master, aserting the BBSY signal. This signal causes the INTR signal to be asserted (SACK is now cleared after INTR but prior to start of last bus cycle) and the Interrupt Vector Address (selected by switches on the IIST Interface Module) to be driven onto the Unibus Data lines. When the Processor sees the assertion of Bus INTR, it waits a short time and then strobes the data lines, and issues Bus SSYN. When the IIST sees the assertion of Bus SSYN,

The processors are all functionally equal with the ability to perform any task and function either as slave or master during program operation. The common memory contains a single copy of the Executive and its data base. Each processor is connected to each memory port and all physical memory is addressable from each CPU.

The system is fully symmetrical, i.e., any task can run on any processor and any processor can service a non-I/O Executive request. Of course, requests for I/O and the servicing of interrupts must be performed by the processor to which the device controller is attached. Each processor is connected by an Interprocessor Interrupt and Sanity Timer Bus (IIST) in order to permit the shared Executive to interrupt processors for the purpose of re-scheduling system activity. The Executive, in effect, treats the processors as if they were additional resource devices available for use whenever program activity demands it, and can schedule tasks to run on the various processors whenever it is required.

2.3 TYPICAL PDP-11/70 MULTIPROCESSOR SYSTEM (2-PROCESSOR CONFIGURATION)

Shown in Figure 2-1 is a simplified block diagram of a typical PDP-11/70 Multiprocessor System using two main processors. The cache of each processor is connected by means of independent main memory Bus cables to a separate port on a box of the MKall multiported common memory; CPUA to port MK0A and CPUB to port MK1B. The memory boxes are connected to each other with a minimum of two boxes required and a possible maximum of up to four memory boxes (eight on a PDP-11/74 MP). Every port on every memory box is connected to a memory Controller panel which controls the operation of the memory port to which it is connected. There is one Boot processor Interrupt and Sanity Timer Panel (IIST) switchboard that is

TO BE SUPPLIED

it drops control of the Unibus by clearing the INTR signal, the vector address and the Bus BBSY signal. The Processor then clears SSYN and asserts BBSY making the CPU the bus master and completing the cycle.

NOTE

The logic circuits in the IIST Interface Module that generate SACK AND BBSY signals when the Sanity Timer times out in order to freeze or lock the Bus are not the same circuits used to generate these signals during a normal communication exchange between the IIST Interface Module and the Processor.

2.6.12.3 IIIST to IIST Interface

All IIST Interface Modules in the system are interconnected by means of the IIST Bus, composed of four signal pairs. This bus carries interrupt and boot requests between the interfaces. (Refer to Figure 2-5.)

Interrupt and boot commands are transmitted serially on the bus. Each IIST drives one pair of signal lines on the bus: one line of the pair is for data, the other for a clocking signal. The pair of lines to be used for transmission is selected by the Self-Indentification switches within the IIST. Each IIST must have a unique Self-ID. Therefore, each signal pair on the bus has at most one driver, eliminating any requirement for a bus arbitration scheme.

TO BE SUPPLIED

Each list interface receives all four serial signal pairs. The line from which an interrupt or boot request is received therefore identifies the source of that request.

Upon transmission, the IIST interface sending interrupt and/or boot requests specifies within a single serial character one or more destination interfaces. The character format is shown on Figure 2-5. There are four pairs of Interrupt/Boot destination fields, one Type bit (specifying whether the Interrupt/Boot requests are program- or sanity-timer generated), one parity bit (provided by the program), and a Framebit (which must be 0). Each receiving interface connected to the particular transmission line receives and assembles the entire character, checks for parity and framing errors, and, assuming no errors are detected, responds to a particular 2-bit field, selected by its Self-ID switches.

Absence of the clocking signal on a transmission line is detected by the receiving interfaces as a powerdown or disconnect condition.

The bus is structured such that there are no serially-threaded paths through any interface. Also, the driving circuits are chosen such that only passive terminations to ground are required at the ends of the cable; these terminators are part of the cable assembly. Furthermore, the receiving circuits always present high impedance to the cable, whether the interface is powered on or off. Therefore, any IIST interface may be powered down or disconnected without affecting the other interfaces.

2.6.12.3.1 Program-Generated Interrupt/Boot Operation

The program causes interrupt and/or boot requests to be sent to other processors in the system by first setting "transmit enable" bits, specifying which processors are to receive an interrupt and/or boot request, and then setting a GO bit, which starts transmission of a serial character on the IIST bus. There are eight enable bits: four to specify processors destined to receive interrupt requests, and four to specify processors destined to receive boot requests. Therefore, in a single operation, any or all processors may be interrupted and/or booted. One processor can selectively ignore requests from another processor by setting a mask bit.

2.6.12.3.2 Sanity Timer Operation

The Sanity Timer portion of the IIST consists of a clock and an 8-bit counter. To start the timer, the program loads the counter with a number specifying the desired timer duration. The clock then decrements the counter once every 256 uS. If the counter underflows from 0 to -1, a timeout is said to have occurred. In a properly running processor, the program will reload the counter before it times out, thereby continually preventing a timeout condition.

A timeout usually indicates presence of a malfunction. The program can specify that other processors be notified, by means of interrupt or boot requests, should a timeout occur. This is done by setting transmit-enable bits, similar to those used for sending program-generated interrupt and boot requests, before the timer is started. When a timeout occurs, the selected request bits are broadcast to receive sanity timer interrupt and boot requests.

In addition, a sanity-timer can be enabled to freeze the state of the processor and its Unibus should a timeout occur. Normally, this is done by requesting NPR mastership and then holding it until an interrupt request is received from another processor. The alternate capability also exists to assert a special lock line connected directly to the processor in order to halt it.

2.6.12.4 IIST to Unibus Map Board and to IIST Boot Control Panel
The IIST Interface Module is connected by means of a 20-conductor ribbon cable designated the Boot/Lock/Control cable (not supplied with the basic IIST subsystems) to the Unibus Map Board (M8181) which is connected to the BOOT/IIST Panel Logic Board. The Panel Logic Board connects to the IIST Panel Switch Board. The Unibus Map Board is also connected by means of an 8 conductor ribbon cable to the Halt Alarm Board and to the M9312 Bootstrap Terminator Module.

2-1

Shown in Table 2-1 are the Boot/Lock/Control Cable signals. Over these lines the IIST Interface Module when it receives a Boot request can send an IIST BOOT signal (IIST BOOT L) to the M9312 Bootstrap Module for booting a processor, or a sanity-timer generated IIST Halt (when the sanity-timer timeouts) which goes to the Unibus Map Board and then to the M81 Board in order to generate a UBCF STOP signal and place the processor in the HALT state. At the same time a HALT ALARM signal is sent through the Unibus Map Board to the HALT Module to sound an audible alarm (optional).

Along this cable the operator by means of switcheson the control panel can send EN XMIT and EN RECVO through EN RCV3 signals that are

Table 2-1

Boot/Lock/Control Cable Signals

From	To	J2 Connector Pin	Signal Name	Description
		A, E, K	GND A	Ground return for output signals IIST BOOT, IIST HALT, and IIST HALT ALARM (pins B, D, and F, respectively).
		B	IIST BOOT L	25 microsecond pulse, asserted LOW when Boot command is accepted by interface. Drives Boot input of M9312 Bootstrap/Terminator module. In unasserted state, this line is pulled to +5 V through 1K-ohm resistor.
		C, V	PANEL +5 V	Supplies +5 Vdc through a 0.5 A fuse. Powers small amount of logic on Boot IIST Panel.
		D	IIST HALT L	Asserted LOW as the result of Sanity-Timer timeout. In unasserted state, this line is open; it should be pulled up by the external receiving logic.
		F	IIST HALT ALARM L	Asserted LOW as result of Sanity-Timer time-out. No pullup resistor

F (cont)		is connected. This signal is also connected to pin 7 or Mate-N-Lock connector JB.
H	(PWR UP BOOT EN)	Connected to pin 6 of Mate-N-Lock connector JB (not driven or received by IIST interface).
J, L, M	None	Not used.
N, T	GND B	Ground return for input signals to M8717.
X	J2 INSERTION Error	When LOW LED indicator on M8717 lights, indicates cable is improperly inserted.
P	EN XMIT H (INH XMIT L)	When high (open), M8717 is enabled to transmit on IIST Bus. When asserted low/ M8717 is inhibited from transmitting on the IIST Bus. Receiving interfaces detect this condition as BREAK.
R	EN RCVO H (INH RCVO L)	When high (open), M8717 is enabled to receive from IIST Bus Line 0 commands transmitted by Interface 0 (Processor 0). When low, M8717 is inhibited from receiving from IIST

R (cont)		Bus Line 0 (Processor 0) causing the Break (BRK) 0 bit in DCLO/Disconnect Flags (DCF) register to set.
S	EN RCV1 H (INH RCV1 L)	Identical to above except used for Interface 1 (Processor 1).
U	EN RCV2 H (INH RCV2 L)	Identical to above except used for Interface 2 (Processor 2).
W	EN RCV3 H (INH RCV3 L)	Identical to above except used for Interface 3 (Processor 3).

used to enable or inhibit the receivers in the IIST Interface Modules of the various processors. These signals are controlled by the IIST CONFIGURATION switches and can be used to group or partition the multiprocessor system in various ways. Figure 2-6 is an interface diagram showing the signal flow between the IIST Interface Module, IIST Control Panel, IIST Logic Board, and the HALT ALARM and the M9312 Bootstrap Terminator module.

2.6.13 BOOT/IIST Switch Panel and Associated Logic

The BOOT/IIST Switch Panel contains the indicators for monitoring the status of all the processors, and the switches for processor BOOT or CPU POWER-UP boot control. Also included are switches for establishing the system grouping or configuration of the multiprocessor system, and for enabling or disabling each processor's IIST Interface Module(On/Off line switch).The IIST Configuration switches determine what combination of multiprocessors and standalone processors will make up the system. Various configurations are possible for example, a dual processor could run with both processors as a multiprocessor in Group A or Group B or each processor could be run as a standalone depending upon the program requirements. The control panel is, therefore, divided into three major sections:

1. CPU STATUS; POWER, RUN or SANITY TIMER HALT

2. SYSTEM BOOT control

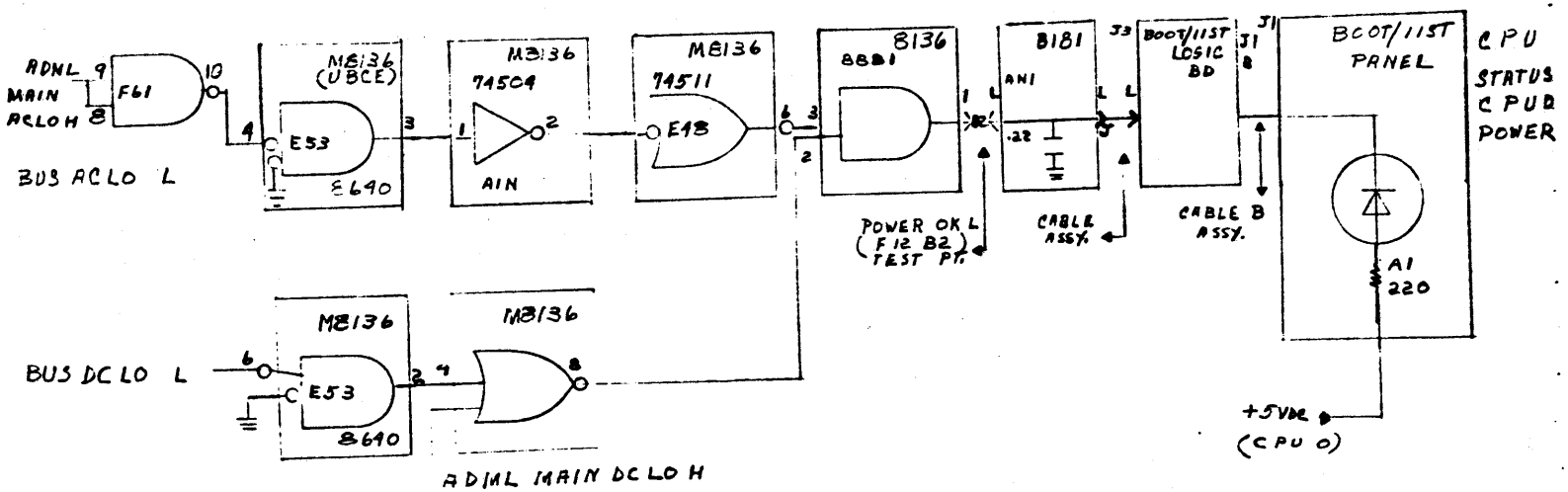
3. IIST CONFIGURATION CONTROL

2.6.13.1 CPU Status Indications

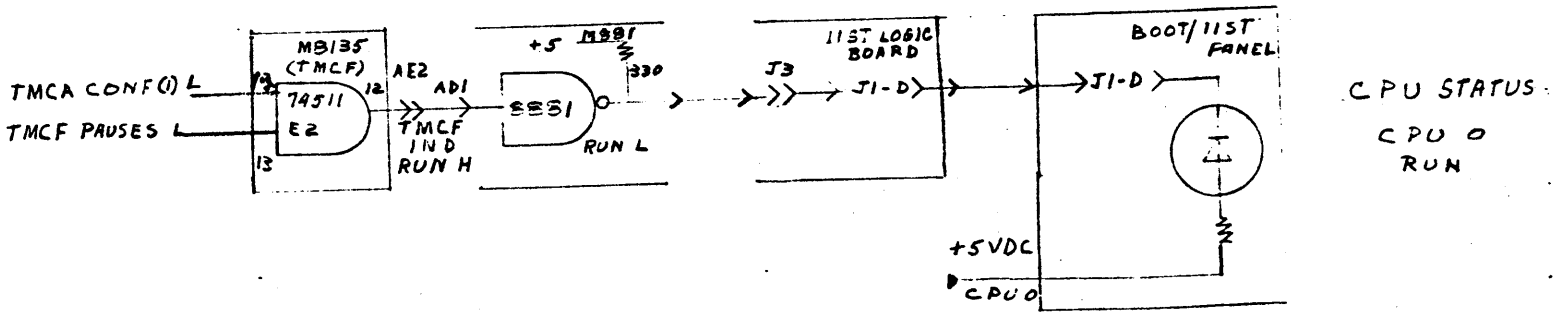
2.6.13.2 Power Led Indicators - These four green LED's labeled CPU 0 through CPU 3, when lit, indicate that their respective processors are powered up and that the power supply status signals ACL0 or DCL0 are not asserted. The simplified signal flow schematic showing the functional signal path from the M8181 Unibus Map Board to the CPU 0 POWER indicator on the IIST Control panel is shown in Figure 2-7. As long as BUS ACL0, or ADMC MAIN ACL0 H or BUS DCL0 L or ADML MAIN DCL0 H is not asserted, the PWR OK L coming from the M8181 (See sheet 6 of schematic) remains asserted and goes to the POWER LED on the IIST control panel causing it to light. If BUS ACL0 L or ADMC MAIN ACL0 H or BUS DCL0 L or ADML MAIN DCL0 H becomes asserted, then the POWER indicator will go out. The +5 Vdc for the LED comes from the Processor power supply. Functional signal flow for the other three CPU status POWER indicators is identical except that they indicate power supply status from other processors. (CPU 1-3). The PWR OK L signal is not used by the IIST Interface Module.

2.6.13.1.2 Run Indicators

The CUP STATUS RUN indicator on the control panel indicates when lighted that the processor is in the RUN state. A simplified schematic showing the functional signal flow for the RUN indicator is shown in Figure 2-7. As long as TMCA CONF L and TMCE PAUSES L signals on the M8135 board (See sheet 6 of the schematic) are not asserted the TMC IND RUN H signal is asserted causing RUN L to be asserted on the M8181 Unibus Map Board and the RUN indicator on the BOOT/IIST Panel to light. If either TMCA CONF L or TMCE PAUSES L goes high indicating that the processor is no longer in the RUN state then the RUN L signal



(A) SIMPLIFED SIGNAL FLOW FOR POWER INDICATOR (CPU 0)



(B) SIMPLIFIED SIGNAL FLOW FOR RUN INDICATOR (CPU 0)

SIGNAL FLOW DIAGRAM FOR CPU STATUS POWER AND RUN INDICATORS (CPU 0)

FIGURE 2-7

is no longer asserted and the RUN indicator on the control panel will go out. The functional signal flows for the other RUN indicators are identical except that they come from different processors.

2.6.13.1.3 Sanity Timer Halt Logic and Indications

The IIST HALT L signal which indicates a Sanity Timer Time-out has occurred is generated on the IIST Interface Module by the Sanity Timer. The Sanity Timer section of the IIST Interface Module (Refer to Figure 2-8, a simplified block diagram of the Sanity Timer) consists of a timing generator, a programmable 8-bit counter, various control and status flags, and the associated system lockup logic. The term "Sanity Timer arises from the application of this section of logic towards determining if the processor and its software to which the IIST Interface is connected is 'Sane', that is, operating properly. The operating program starts the timer by loading the programmable counter, with a number specifying the desired timer duration and will then periodically issue refresh commands to prevent it from timing out. As long as the counter is reloaded periodically to prevent it from timing out, the program and system will then be considered to be operating normally or 'sane'.

The program can specify that any or all other processors be notified by means of interrupt or boot requests when a timeout occurs (usually indicates the presence of a malfunction). This is done by setting transmit-enable bits in the Sanity Timer Transmission Enables (STTE) register before the timer is started and transmitting the contents of the register via the IIST Interface Modules to notify other processors in the system of an insane condition. In addition, the sanity timer can be enabled to halt the processor and trigger its Unibus should

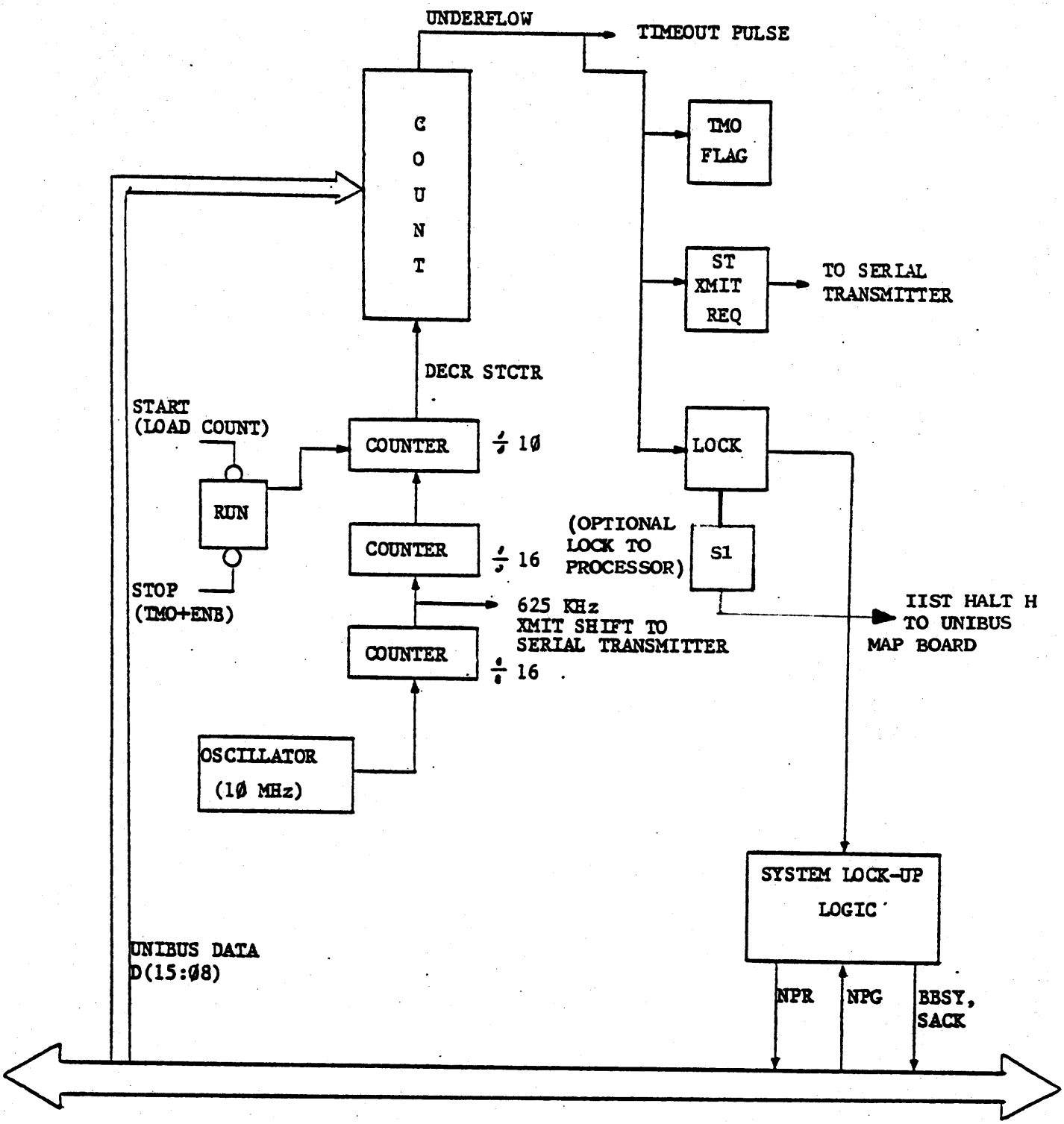


FIGURE 2-8

SANITY TIMER - BLOCK DIAGRAM

a timeout occur. The Bus is frozen by requesting NPR mastership and then holding the Bus by keeping BBSY and SACK asserted until an interrupt request is received from another processor. The processor is halted by assertion of the UBCF STOP L signal on a special lock line that is connected directly to the processor.

The HALT instruction stops program execution by the processor; and control is transferred to the console service sequence which waits for manual intervention to determine further operations. This is performed by setting the console flag which generates a (BRQ)Break Request. The console flag is set only if the processor is in Kernel mode when the HALT occurs. If the processor is not in Kernel mode then control is transferred to the trap service sequence, and a processor halted in Super or User modes traps through location 4.

2.6.13.1.3.1 Detailed Halt and Lock Signal Flow

Shown in Figure 2-9 are the functional signal paths for the Sanity Timer Halt and Lock operation.

When the counter underflows from 0 to 1 the Timeout Pulse (TMO PLS) is asserted. The TMO PLS sets the Lock Flag (D06 LOCK (1) H is asserted). When set the lock flag will attempt to lock up the processor and its Unibus, preventing an insane processor from interfering with other processors in the system. Two switch selectable modes of lockup are available. (SWS1 and SWS2 on switch pack at E86.) In one, a special line connected directly to the processor is asserted. In the other, the IIST will attempt to lock the processor by hanging its Unibus. It does this by first asserting an NPR (NON-PROCESSOR REQUEST) on the Unibus. Then, when the processor returns an asserted

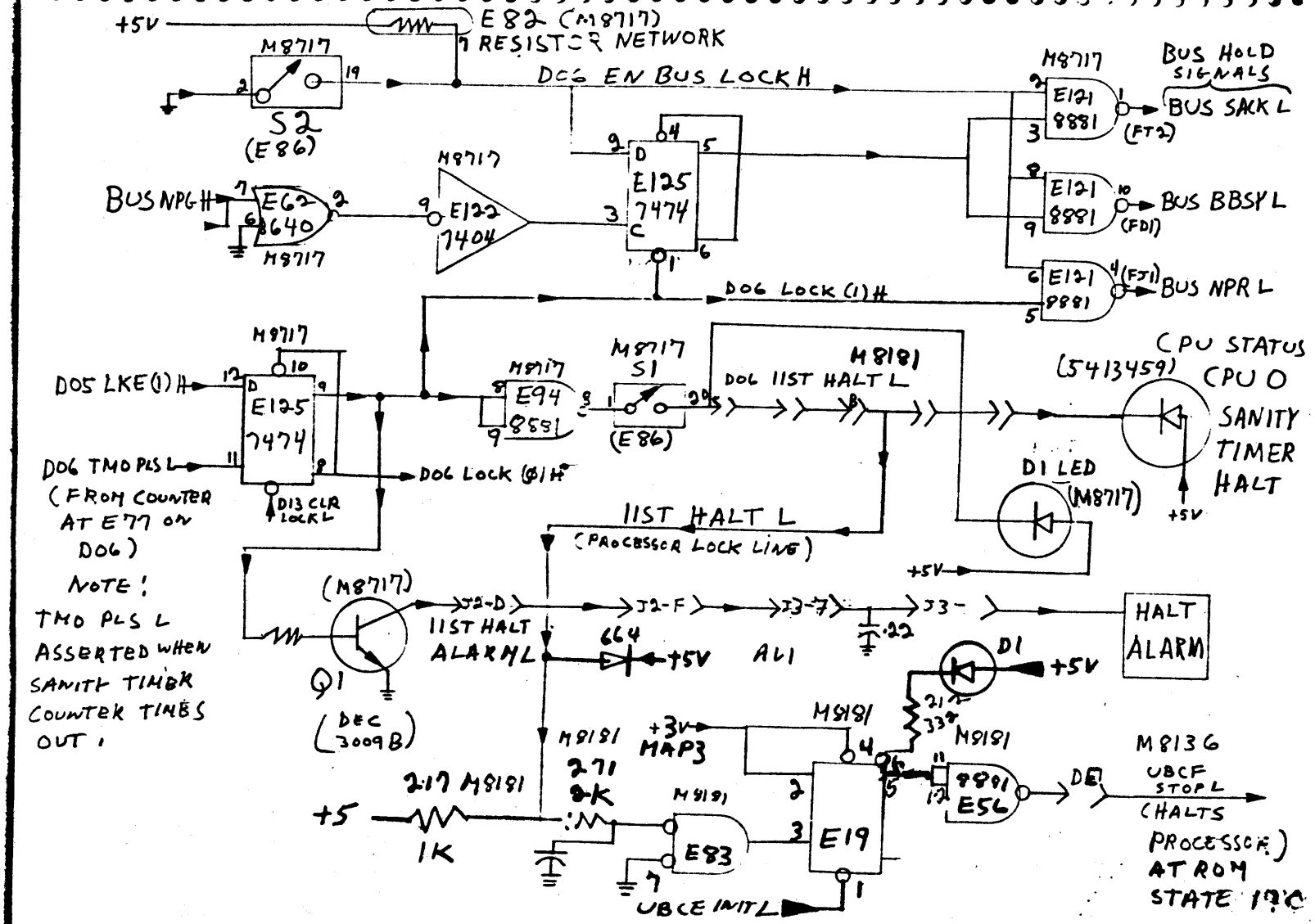


FIGURE 9-9 SIMPLIFIED SCHEMATIC OF SANITY TIMER HALT AND LOCK LOGIC

NPR (Non-Processor Grant), the IIST immediately asserts SACK and BBSY and will hold these signals asserted.

2.6.13.1.3.2 NPR Bus Hold

When the counter underflows from 0 to -1 (Sanity Timer Timeout), the Timeout Pulse (TMO PLS L) is asserted. This pulse clears ST RUN and sets the Time-out flag in the IIST Sanity Timer Control Status (STCS) register. It also sets ST XMIT REQ which requests that the contents of the STTE register be sent to other IIST interfaces in the system.

If the Lockup Enable, LKE (1) H, in the Sanity Timer Control Status register is set causing the D input of the LOCK flip-flop at E125 to go high then the TMO PLS signal will set the LOCK flip-flop and LOCK (1) will be asserted. (Switch S1 at E86 should be open-off). If switch S2 at E86 is off when LOCK (1) H is asserted than a NPR is asserted by the IIST Interface Module on the Unibus. When the processor returns an asserted NPG (Non-Processor Grant), the IIST immediately asserts SACK and BBSY and will hold these signals asserted. Bus Control, therefore, will have been seized either immediately if BBSY was not already asserted on the Unibus or when the current bus master drops its BBSY. The Lockup condition will be cleared upon a valid interrupt command request received over the IIST Bus. It will also be cleared by a Boot command; the boot need not be mask-enabled. Lockup is also cleared by a Unibus INIT or ACLO signal.

TO BE SUPPLIED

2.6.13.1.3.3 Direct Sanity Timer Halt Lock to Processor

If switch S1 on the IIST Interface Module is closed when the timeout occurs, generating the Timeout Pulse (TMO PLS L) which sets the LOCK flip-flop causing LOCK (1) H to be asserted, then the IIST HALT L signal is asserted and sent to pin B of connector J2 of the IIST Interface Module and
to the Unibus Map Board, M8181. IIST HALT L sets a flip-flop on the Unibus Map Board, whose output is sent through a NAND gate to the Unibus Console Board, M8186, causing pin DE1 to go low and generating the UBCF STOP L signal which halts the processor and preserves its state.

The lock condition is held, thereby locking the processor, until the LOCK flip-flop is cleared by occurence of the CLR LOCK L signal which is a pulse asserted by BUS ACLO, INIT, or upon reception of a valid BOOT request received along the IIST BUS.

Assertion of the LOCK (1) signal also causes the HALT ALARM to sound as is described in the HALT ALARM paragraph (2.6.13.1.3.4.) In addition, the IIST HALT (1) L signal is sent along the 20-conductor ribbon cable through the Unibus Map Board, M8181, through the IIST Logic Board and to the CPU STATUS SANITY TIMER HALT indicator on the BOOT/LIST

Switch Panel Board causing it to light, thereby, indicating a processor HALT caused by a Sanity Timer timeout.

NOTE

The CPU STATUS SANITY TIMER HALT indicator does not light when the system lockup is achieved by freezing the Bus through an NPR request and then by maintaining SACK and BBSY on the Bus.

The IIST HALT L signal is also tied to an LED on the IIST Interface Module, and causes it to light when it is asserted.

2.6.13.1.3.4 Halt Alarm

The assertion of the LOCK (1) H signal that occurs when a Sanity Timer times out also causes an Alarm to sound if the Halt Alarm option is installed. The LOCK (1) H is tied to the base of an NPN transistor on the IIST Interface Module, and when asserted causes the transistor Q1 to turn on and assert an IIST HALT ALARM L signal. This signal goes from the IIST Interface Module through the Unibus Map Board to the HALT Alarm module which sounds an alarm indicating that the processor or its Bus is in a hold state. A +15 Vdc is supplied to the HALT ALARM board from the IIST Interface Module. The HALT ALARM will sound in both types of system lockup. The IIST HALT ALARM L signal path is shown in Figure 2-9.

2.6.13.2 System Boot From Boot/IIST Switch Panel Board

The System Boot section on the Boot/IIST Switch Panel, which is used to boot any or all of the processors by means of a direct BOOT

signal to the M9312 module of each processor, consists of the following switches:

1. SYSTEM BOOT - A push-button switch with momentary contact that actuates the external boot inputs of all boot modules enabled by the CPU BOOT SELECTION switches.

2. CPU BOOT SELECTION - These four switches select the processors to be booted and are labeled CPU 0 through CPU 3. They have 3 positions:

   2a. MOMENTARY ENABLED - Momentary contact switch with spring return to DISABLE. Bootstrap actuation is enabled only while the switch is momentarily held in this position and the SYSTEM BOOT switch is pressed.

   2b. DISABLE - No BOOT is generated when the SYSTEM BOOT switch is pressed.

   2c. ALWAYS ENABLED - When in this position the BOOT signal will always be sent to the M9312 BOOTSTRAP Module when the SYSTEM BOOT pushbutton switch is pressed.

Shown in Figure 2-10 is the Functional Boot Signal Path from the BOOT/IIST Panel Switch Board.

When the SYSTEM BOOT switch is pressed and the CPU BOOT SELECTION switch of any one or all of the processors is either MOMENTARILY

TO BE SUPPLIED

ENABLED or in the ALWAYS ENABLED position, a BOOT L (One of 0 through 3 signals) signal is asserted and sent from the BOOT/IIST Switch Panel through the BOOT/IIST Logic Board through the M8181 Unibus Map Board and to the TP1 input of the M9312 Bootstrap Module of the particular processor that has been selected.

When a low is applied to TP1, BUS ACLO L is asserted on the Unibus. Upon detecting this Unibus signal the processor begins a power-down routine anticipating a real power failure. After completing this routine, the processor then waits for the unassertion of BUS ACLO L at which time it will perform a power-up sequence through location 24 and 26. (normal address locations for a power-up sequence without a boot). Upon release of the external SYSTEM BOOT switch on the BOOT/LIST Switch Panel, TP1 returns to a high and a timeout is generated. At the end of the timeout, BUS AC LO L is cleared. The processor is then forced to read its new PC and PSW from locations 773024 and 773026 respectively. This location is an address in the M9312 ROM space and the address of the offset switch bank, which contains the starting address of a specific routine. Having obtained a new PC from location 773024, the processor then obtains a new Processor Status Word from address 773026 in the ROM address space. This new PC and PSW direct the processor to a Bootstrap Program in the M9312 ROM memory (Unibus memory locations 773000 through 773776 and 76500 through 765776 depending upon the routine selected as determined by the setting of S1-1 and S1-3 through S1-10 switches on the M9312). The actual boot routine is selected by these nine switches.

2.6.13.3 CPU POWER-UP ACTION

There is a CUP POWER-UP ACTION switch for each CPU (labeled CPU 0

through CPU 3) which determines whether a boot will occur and, if so, under what conditions during power-up sequences. These switches have three positions:

1. BOOT - Any power-up sequence will cause a BOOT to be executed.

2. RUN OR BOOT (IF BATTERY DEPLETED) - During a power-up if the back-up battery for the MKA11 memory is depleted a boot will automatically occur. Otherwise, the CPU goes through a normal power-up sequence.

3. RUN OR HALT (IF BATTERY DEPLETED) - The power-up BOOT is disabled. During a power-up, if the back-up battery for the memory is depleted a boot will not occur, instead the CPU will trap to 0 and halt. Otherwise, the CPU goes through a normal Power-up sequence and proceeds to RUN. (execute instruction cycles).

2.6.13.3.1 Automatic Boot on Power-Up

Shown in Figure 2-11 is the Functional Signal Path for Automatic Boot on Power-Up.

Automatic booting on power-up can be enabled or disabled using the CPU POWER-UP ACTION BOOT switch. When this switch is in the Boot position during a power-up, PWR UP BOOT EN L (one of 0-3 signals) is asserted from the IIST Control Panel through the IIST logic Board, the M8181 Unibus Map Board and to TP 4 on the M9312 Bootstrap Module (or processors) of the processor whose CPU POWER-UP ACTION switch is in the BOOT position.

TO BE SUPPLIED

As long as this switch is in the BOOT position, the processor during a power-up will obtain its new PC and PSW from locations 773024 and 773026 respectively. The address of the offset Switch Bank (S1-1 and S1-3 through S1-10) is 773024 (773224 if the processor traps to 224 on power-up). The new PC and PSW will then direct the processor to a program bootstrap in the M9312 ROM memory (Unibus memory locations 7730000 through 773776, and 765000 through 765776.

2.6.13.3.2 Run or Boot Only if Battery Depleted

Shown in Figure 2-12 is the Functional Signal Path for a BOOT on Power-Up when the battery is depleted.

When the CPU POWER-UP ACTION switch is in the RUN OR BOOT (IF BATTERY DEPLETED) position, Mem BOOT L lines are connected from the J9 connector of the first memory box of the MKA11 multiported memory to the BOOT/IIST Logic Board through the Unibus Map Board (M8181) and to the M9312 Module of each individual processor. The MEM BOOT L lines are connected through the J9 connector to the battery status signal lines of the memory power supply. If this status signal goes low, during a power-up it indicates that the contents of the memory are no longer valid, and a MEM BOOT L (one of four possible signals 0-3) that is sent to TP1 of the M9312 Module causing a boot of the processor identical to the sequence (once the BOOT L signal is generated) described in paragraph 2.6.13.2. (PC and PSW are obtained from locations 773024 and 773026 since a boot is to occur.) If the battery status input remains high indicating that the contents of the memory are still valid, then a MEM BOOT L signal will not be asserted and the M9312 will not automatically boot on power-up. In this case the processor obtains its

TO BE SUPPLIED

new PC from location 24 and its new PSW from location 26.

2.6.13.3.3 Run or Halt (If Battery Depleted)

Shown in Figure 2-13 is the functional signal path for a HALT on Power-Up without a boot, when the battery is depleted.

When the CPU POWER-UP ACTION switch is in the RUN OR HALT (IF BATTERY DEPLETED) position the MEM BOOT L lines from the J9 connector on the backplane of the memory are open and the MEM BOOT L signal from the memory cannot be asserted to the M9312. Therefore, even if the battery is depleted, a boot cannot occur. Instead, if the memory battery back-up is depleted, the CPU will trap to 0 and HALT.

2.6.13.4 IIST BOOT PULSE (Program Generated)

Software running in a CPU can cause any of the CPU¦s to begin executing the code of the M9312 Bootstrap Module that is local to each processor. When any serial receiver in the IIST Interface Module of a particular processor receives a boot request from a processor and its BOOT MASK bit is clear, a timer in the IIST Interface Module is triggered which produces a 25 us pulse which is applied to an 8881 open-collector driver. The other input to the driver is the D13 DSBL BOOT (0) H output of the Maintenance Control Register (MTC). If the DSBL BOOT bit in the MTC register is clear, D14 IIST BOOT L pulse signal is asserted on pin F of the J2 connector. This signal is tied via the 20-conductor cable to the Unibus Map board from which it is sent along an 8-conductor cable to the TP1 input of the M9312 in order to boot the processor.

TO BE SUPPLIED

In multiprocessor operation the IIST BOOT pulse is used by the multiprocessor boot code to permit the first CPU that has been booted to bring up the other CPU's in the system.

2.6.13.5 IIST ENABLE ON LINE/OFF LINE Switches

The four IIST ENABLE switches labeled CPU 0 through CPU 3 control the ON LINE and OFF LINE status of the IIST Interface Modules for each processor. When set to the ON LINE position this switch enables its associated CPU's IIST Interface Module to transmit and receive from itself and from any other CPU's IIST that is on its system as determined by the IIST CONFIGURATION switches.

When this switch is set to the OFF LINE position, it disables the drivers and BUS receivers of its associated CPU's IIST preventing the IIST from transmitting to or receiving from any of the other processor's IIST's. This switch position overrides the IIST configuration switches.

NOTE

This switch does not affect the internal maintenance send/receive features of the IIST.

Shown in Figure 2-14 is a functional ON/OFF LINE signal path to the IIST Interface Module from the BOOT/IIST Panel Switch Board. When the IIST ENABLE switch is in the OFF LINE position an OFF LINE L is asserted and sent to the IIST BOOT/IIST Panel Logic Board where it is used to disable or prevent the assertion of the XMIT and receive enables: IIST 0 XMIT EN H (assuming that it is CPU 0 in the OFF LINE

TO BE SUPPLIED

position) and IIST O RCV O through 3 EN H. The IIST Interface Module for CPU O is, therefore, disabled and is unable to receive or transmit over the IIST Bus.

If Processor O is in the ON LINE position then O OFF LINE L is not asserted and the transmit and receive enables are asserted enabling IIST O to transmit and receive over the IIST Bus. The transmit and receive enabling signals are normally high since they are tied through R9 and R10, 4.9K resistors, to the +5 Vdc.

2.6.13.6 IIST Configuration Switches

These three switches labeled CPU O through CPU 3 control the various multiprocessor and standalone system configurations. The processors in a multiprocessor system can operate as multiprocessors when they are grouped by means of these switches together in System A or System B, or they can operate individually as standalone processors when the switch associated with that CPU is placed in Standalone. These four switches have three positions:

A - Enables its respective CPU's IIST to transmit to and receive from any other CPU's IIST that is also in the A position.

STANDALONE - Disables respective CPU's IIST from receiving from any other CPU's IIST. Also disables all other IISTS from receiving it.

NOTE

The processor in the STANDALONE position can transmit and receive from itself.

B - Enables its respective CPU's IIST to transmit to and receive from any other CPU's IIST that is also in position B.

Shown in Figure 2-15 is a simplified schematic of the switch configuration logic contained on the BOOT/IIST Panel Logic Board, and connected to the drivers and receivers in the IIST Interface Modules.

To assert the IIST RCV ENABLES at least two processors must be grouped in position A or B together and neither can be offline. In the event that at least two processors are not grouped together in either A or B, the IIST RCV ENABLES will not be asserted. Any combination or configuration of multiprocessor and standalones are possible with the two to four multiprocessor configurations. For example in a four processor configuration, two could be grouped in A and two in B, or three or four could be grouped in System A or B, or two could be in A or B and the other two operating as standalones. Any one processor can be taken off line from the multiprocessor operating system without interfering with the other processor's operation. Taking a processor OFFLINE in this case simply means setting its IIST ENABLE switch to OFFLINE thereby disabling its IIST Interface module. It still would be available for completely independent operation or for OFF LINE maintenance purposes.

With the IIST ENABLE switch of CPU 0 and CPU 1 in the ON LINE position, 0 OFF LINE L is not asserted, therefore IIST 0 RCV 0 EN H and IIST 1 RCV 1 EN H and IIST 0 XMIT EN H and IIST 1 XMIT EN H are asserted (high) enabling the drivers in IIST 0 and IIST 1 and bus receiver 0 in IIST 0 and bus receiver 1 in IIST 1.

TO BE SUPPLIED

Assuming a dual processor configuration with both processors in System A, (CPU 0 and CPU 1) then (input signals to L502 at E3) 0 A L signal would be asserted and the 1 B L signal would not be which would cause the IIST 1 RCV 0 EN H and IIST 0 RCV 1 EN H signals to be asserted, enabling receivers 0 in IIST 1 and receivers 1 in IIST 0. Thus Processor 0 and 1 (via their IIST Interface Modules 0 and 1 and the IIST BUS) would be able to transmit and receive INTERRUPT AND BOOT requests from each other. If, for example, CPU 0 was placed in the STANDALONE position, then a 0 STANDALONE L signal would be asserted to the inverted input of NAND L509 and the IIST 0 RCV 1 EN H and the IIST 1 RCV 1 EN H would go low disabling receivers 1 in IIST 0 and receivers 0 in IIST 1. Therefore, CPU 0 and CPU 1 would no longer be able to transmit and receive from each other along their IIST Bus.

While the example above describes a two-processor configuration, tri or quad processor configurations would operate essentially in the same way. For example, if four processors were grouped in System A, the drivers and all the receivers (bus receivers 0 through 1) in each Processor's IIST Interface Module would be able to receive and transmit to each other. If a processor were set to a STANDALONE position, the receivers associated with that processor in each IIST Interface Module would be disabled and that processor would not be able to receive from the other processors.

2.6.14 IIST Interface Module Registers

Shown in Figure 2-16 is a simplified block diagram of the IIST Interface Module.

Two 16-bit Unibus registers are employed to interface the IIST to the PDP-11-70 processor. These registers act as an access path to ten internal device registers, which may be read and loaded under program control (by first specifying an Access Code) to monitor IIST status and to control operation.

Table 2-2 lists the two access-path registers and their relative locations (e.g., Unibus addresses). The base address, b, of the block of two registers is switch-selectable in the range 760000-777770. Table 2-3 lists the internal device registers and their respective access codes.

Table 2-2

IIST Unibus Registers

Mnemonic	Register Name	Address
ACR	Access Control Register	B+0
ADR	Access Data Register	B+2

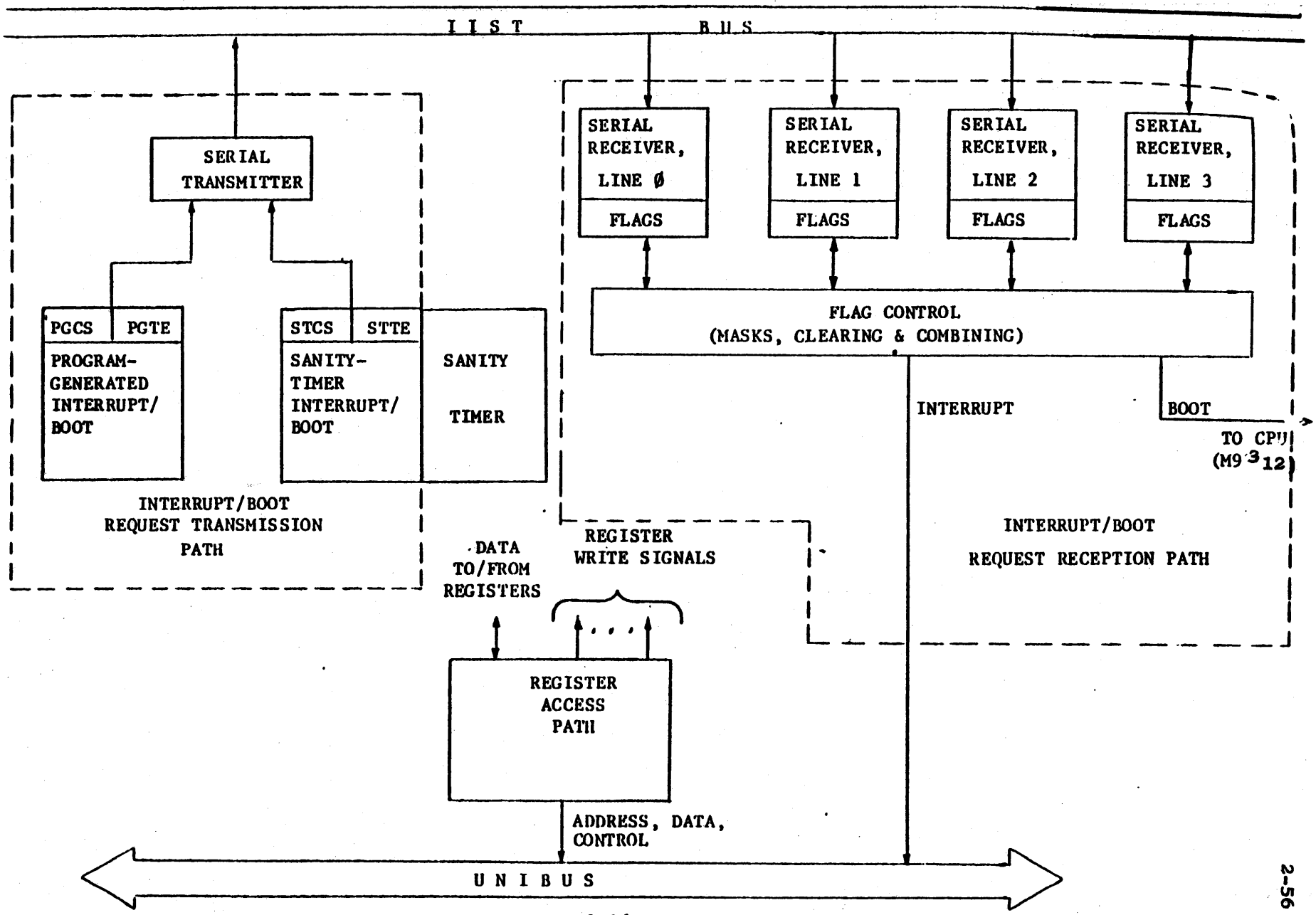


FIGURE 2-16
SIMPLIFIED IIST BLOCK DIAGRAM

Table 2-3

IIST Device Registers

Mnemonic	Register Name	Access Code
PGTE	Program-Generated Transmission Enables	0
PGCS	Program-Generated Control-Status	1
STTE	Sanity-Timer Transmission Enables	2
STCS	Sanity-Timer Control-Status	3
IMSK	Input Masks	4
PFG	Program-Generated Flags	5
STF	Sanity-Timer Flags	6
DCF	DCLO/Disconnect Flags	7
EXC	Exceptions	10
MTC	Maintenance Control	15
None	Not used	16-17

2.6.14.1 Access-Path Register Descriptions

Figure 2-17 shows the bit configurations of the Access Control Register (ACR) and the Access Data Register (ADR). The following paragraphs describe each register in detail.

In all register descriptions, the term RESET means the occurrence of the Unibus INIT signal or a programmed clear via the CLR bit in the ACR register.

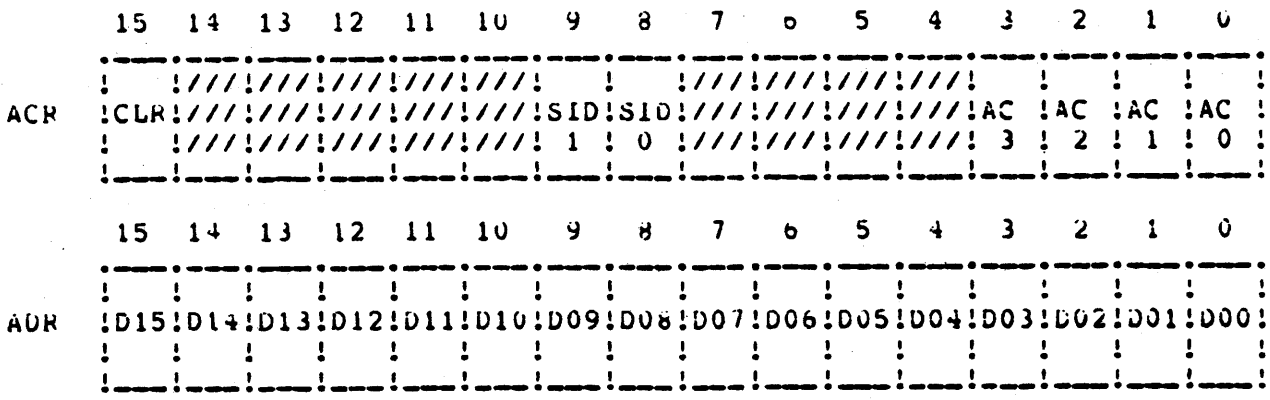


Figure 2-17
Access Path Register BITS

2.6.14.2 ACR (Access Control Register)

The ACR contains a 4-bit code which defines which of the internal IIST device registers is to be accessed via the Access Data Register (ADR). The codes, defined by AC(3:0), are assigned as follows:

Code	Internal Register
0	PGTE
1	PGCS
2	STEE
3	STCS
4	IMSK
5	PGF
6	STF
7	DCF
10	EXC
11-14	Unused
15	MTC
16-17	Unused

The AC (3:0) field is read/write and is cleared by RESET. This field is auto-incremented after each DATI, DATO, or DATOB access to the ADR. (all accesses except DATIP) when one of the codes 0-10 is present in AC (3:0). If AC(3:0) contains any code in the range 11-17, incrementation will not occur, thereby aiding in the prevention of erroneous accesses to the MTC register.

Bits 8 and 9 of ACR display the state of the hardware Self-ID Select switches, thereby allowing the program fast access to this information.

Bit 15 of ACR, CLR, is a write-only bit. When a 1 is written into this bit, the IIST interface is initialized.

Bits 4-7 and 10-14 of ACR are unused. They are always read as 0/ writing into them has no effect.

The ACR is byte-addressable.

2.6.14.3 ADR (Access Data Register)

The ADR provides the access path for reading and writing the internal device register selected by the AC(3:0) code in ACR. If an undefined code is specified, a Slave-Sync response occurs on the Unibus, but no device register is read or written and the AC(3:0) field is not incremented.

All registers selected to appear in ADR are byte-addressable except PGTE and SITE; these two registers should be written by word-operand instructions only.

2.6.14.4 Device Register Descriptions

Table 2-4 describes the bit configuration of each device register. Figure 2-18 lists the bit configurations of the IIST device registers.

2.6.14.5 Register Access Path

The register access path shown in Figure 2-19 is part of the Unibus Interface portion of the IIST. It provides the interface that enables the program to read from or write into any register in the IIST. Specific bits in these registers are designated as follows: 'read only' bits indicate that the program can read the status of these bits but cannot load them; 'write only' bits indicate that the program can load them but will read back a 0; 'read/write' bits indicate that the program can load them and read back the status; 'read-clear' bits indicate that the program can read the status of the bits and can clear them by writing a 1 (writing a 0 has no effect); unused bits are always read as 0 and writing into them has no effect.

The IIST contains two types of registers, described in terms of their method of access. One type is termed direct access, meaning that the register is directly accessible as an addressable location on the Unibus. The Access Control Register (ACR) is a register of this type. The other type of register is termed 'indirect access'. This type of register is accessed by first writing a 'register selection code' into the ACR and then reading or writing the selected register via the direct-access Access Data Register (ADR). The ADR is merely a path allowing access to the various internal (indirect-access) registers of the IIST.

REGISTER				BITS					BIT FUNCTION
MNEMONIC	BYTE ADDRESS-ABLE?	NAME AND FUNCTION	ACCESS CODE	BITS	SIGNAL	NAME	RD?	WRT? CLEARED BY?	
PGTE	NO. Loaded by word operand instructions only.	Program-Generated Transmission Enables – Enables transmission of program-generated Interrupt and Boot signals to processors.	0	0-3	PIE 0-3	Program Interrupt Transmit Enables 0-3	Yes	Yes Reset	If 1 allows corresponding processor (PIE 0-4 corresponds to CPU A-CPU D) to receive Interrupt commands when GO bit in PGCS is 1 (assuming PTP bit carries correct parity for PGTE and RDY is 1.)
				4-7	None	None	Always 0.	No	Not used.
				8-11	PB 0-3	Program Boot Transmit Enables 0-3	Yes.	Yes. Reset	If 1, allows corresponding processor to receive Boot commands when GO bit in PGCS is 1 (assuming PTP bit carries correct parity for PGTE and RDY is 1).
				12-15	None	None	Always 0.	No.	Not used.
PGCS	Yes.	Program-Generated Control Status - contains control bits that cause program-generated interrupt and boot signals to be transmitted to IIST interface modules of each processor. Also contains status and control bits that effect operation of IIST Interface modules in other ways	1	0	GO	Program Generated Interrupt/ Boot GO			a. If RDY is 1, when GO bit is set to 1 RDY is cleared and a 10 bit serial character (processor destinating ON interrupt and/or Boot command, program or sanity timer generated PTP parity bit and Frame bit. is clocked over IIST Bus to all IIST interface modules. Self-identification (ID) switches manually preset on IIST interface module determine what pair of transmission lines will be used UPON completion of transmission cycle, RDY is set to 1. b. If RDY is 0, when GO is set, GRT (GO REJECT) error bit in PGCS is set.
				1	PTP	PGTE	Yes	Yes Reset	PTP bit must reflect ODD parity (PGTE register plus PTP bit itself equals odd number of ones) when GO is set. If PTP does not reflect ODD parity, IIST interface modules will set RTE (parity or framing errors) bits in EXC (Exceptions) register causing program interrupts.
				2	IE	Interrupt	Yes.	Yes. Reset	a. when 0, IIST cannot interrupt processor. b. when 1, ERR (Error) bit in PGCS register and interrupt flags in PGF, STF, DCF and EXC registers may cause program interrupts.

Table 2-4 DIP11-A (IIST Interface Module) Register Bit Descriptions
(Sheet 1 of 8)

MNEMONIC	BITS	SIGNAL	NAME	RD?	WRT?	CBEARED BY?	BIT FUNCTION
PGCS (Cont)	3	IP	Interrupt	Yes	No.	Reset	If set to 1 when IE is 1, a program interrupt BR (Bus Request) is asserted.
	4-7	None	None	Always 0	No.		Not Used
	8-9	SID (1:0) V 2-bit Field	Self I-D	Yes	No.		Indicates logical state of self-identification (SID) signals applied to four serial receivers in IIST interface module. SID signals can mean: a. If EN MXD (Enable Maintenance Data Mux) is 0, SID (1:0) indicates configuration of actual SID switches on IIST interface module. b. If EN MXD is 1, hardware switches are overridden and SID (1:0) indicates state of MNT ID (1:0) field in MTC register. NOTE **For transmission, switches cannot be overriden.**
	10	None	None	Always 0	No.		Not Used
	11	RDY	Ready	Yes.	No.	Setting GO	When 1, program-generated interrupt/Boot logic is ready to accept GO command to initiate program-generated interprocessor Interrupt/Boot cycle set upon completion of cycle.
	12	STRMR	STTE Register Modification refused	Yes.	No.	Reset or by 1 into ERR bit of PGCS register.	Set to 1 if attempt to write to STTE register is made while Sanity-Timer Transmission Cycle is inprogress. STTE register is not modified and ERR bit is set.
	13	PG RMR	PGTE Register Modification refused.	Yes.	No.	Reset or by 1 into ERR bit of PGCS Register.	Set to 1 if attempt to write to PGTE register is made while RDY is 0. PGTE register is not modified and ERR bit is set.
	14	GRJ	GO Reject	Yes.	No.	Reset or by 1 into ERR bit.	Set to 1 if GO is not while EDY is 0, indicating a PI/PB cycle could not be initiated because one was already in progress.

Table 2-4 DIP11A (IIST Interface Module) Register Bit Descriptions (Sheet 2 of 8)

PGCS (Cont)									is not effected and PGTE register is not modified.
				15	ERR	Error	Yes.	No.	Reset or by 1 into ERR bit	Set to 1 if write to PGTE register is attempted while RDY is set or to STTE register while Sanity-Timer transmission Cycle is in progress also, if GO is set while RDY is 0, indicating PI/PB cycle could not be initiated because one was already in progress. Of set to 1 when IE is 1, an interrupt will be generated.
STTE	No. Loaded by word-operand	Sanity-Timer Transmission Enables - Enables Interrupt and Boot signals so that transmission can occur upon timeout of the Sanity Timer	2	0-3	SI0-SI3	Sanity-Timer Interrupt Enable 0-3	Yes.	Yes.	Reset (to 0)	If this bit is 1, when timeout of Sanity-Timer occurs (assuming that the STP bit in the STCS register has odd parity), Sanity-Timer Interrupt command is sent to the IIST Interface Modules to Interrupt corresponding processor (SI0-SI3 corresponds to CPUA - CPUD respectively).
				4-7	None	None	Always 0	No.		Not Used.
				8-11	SBO-SB3	Sanity-Timer Boot	Yes.	Yes.	Reset (to 0)	If this bit is 1, when timeout of Sanity Timer occurs (assuming that the STP bit in the STCS register has odd parity), Sanity-Timer Boot Command is sent to the IIST interface modules to boot corresponding processor (SB0 - SB3 corresponds to CPUA - CPUD).
				12-15	None	None	Always 0	No.		Not Used.
STCS	Yes	Sanity-Timer Control	3	0	ENB	Sanity	Yes.	Yes.	Reset	When 0, inhibits Sanity-Timer counting by holding clear Sanity-Timer RUN flip-flop. When 1, enables Sanity-Timer so that counting proceeds when count field is loaded setting RUN. (COUNT and ENB may be loaded simultaneously to start counting.)
				1	STP	STEE	Yes.	Yes.	Reset	When STP reflects odd parity (STTE Register plus STP bit itself equals odd number of ones) processors respond to interrupt and/or boot commands when timeout occurs. If STP does not reflect Odd parity RTE bit (parity or Framing error) will be set in EXC register cuasing RTE errors when timeout occurs and preventing processors from responding to Interrupt or Boot Commands.

Table 2-4 DIP11-A (IIST Interface Module) Register Bit Descriptions (Sheet 3 of 8)

STCS (Cont.)				2	LKE	Lockup	Yes.	Yes.	Reset	If Sanity-Timer timeout occurs when LKE is 1, IIST freezes the state of the processor either by taking Unibus mastership by means of an NPR request or by a direct lock line to the processor depending upon configuration of Lockup Option Select switches. The freeze is released by INIT or by valid received interrupt command. If LKE is 0, no lock occurs.
				3	TMO	Timeout	Yes	No	Reset or by writing 1	Set when Sanity-Timer underflows causing interprocessor interrupts and/or boots (if enabled by bits in STTE) and/or lockup of processor's state (if enabled by LKE).
				4-7	None	None				Not Used.
				8-15	Count (7:0) field	Sanity-Timer Duration Count	Yes	Yes	Reset	Count FIELD is a binary counter specifying timer duration. Resolution is 256US, giving a maximum duration of 68.536 milliseconds (256 counts).Writing count starts timer if ENB is 1. If ENB is 0, counter is loaded, but counting does not occur. Loading Count sets the RUN flip-flop (assuming ENB is 1), allowing Count to be decremented once every 256 US. When counter underflows from 0 to -1, timeout occurs causing TMO to be asserted and clearing RUN. If LKE is asserted, system lockup will occur. (Loading Count with 0 gives minimum count of 1.)
IMSK	Yes	Input Masks - Contains bits disabling reception of inter-processor interrupt and boot commands.	4	0-3	IMO-IM3	Interrupt Masko -3	Yes	Yes	Reset	When set to 1, interrupt commands will not be accepted from corresponding processors. (IMO-IM3 corresponds to CPU A - CPU D respectively). However, corresponding UIF-UI3 bit will be set in the EXC register. If an interrupt command is sent. When set to 0, any program-generated, Sanity-Timer or DCLO/ Disconnect interrupt signal will be accepted from corresponding processor, causing appropriate flag bit in PGF STF, or DCF registers to be set. In addition, corresponding RTE 0 - RTE 3 bit in EXC register will set upon reception of an information packet containing parity or framing error.
				8-11	BM0-BM3	Boot Mask 5 Boot Mask 3	Yes	Yes	Reset	Identical to 1MO-1M3 bit operation, except boot commands are inhibited from corresponding processor when they are set.

Table 2-4 DIP11-A (IIST Interface Module) Register Bit Descriptions (Sheet 4 of 8)

IMKK Continued				4-7 and 12-15	none	none	Always no read effect 0		Not Used.
PGF	Yes	Program-generated I/B Flags - indicates valid, mask-enabled program-generated interrupt and boot commands (information packets) have been received.	5	0-3	PIF0-PIF3	Program-generated Interrupt Flags	Yes	Writing a 1 or Reset.	Set to 1 when program-generated interrupt requests are received from corresponding processors (P1FO-P1F3 corresponds to CPU A - CPU B)
				4-7	none	none	Always 0		Not Used.
				8-11	PBF0-PBF3	Program Boot Flag	Yes	by writing 1 (not affected by Reset)	Set to 1 when program generated boot requests are received from corresponding processors. NOTE: Actual boot operation is Triggered by signal that sets flag not by flag itself. Since bits are not affected by Reset, Processor can identify source of BOOT after BOOT operation has been completed.
				12-15	none	none	Always 0		Not Used.
STF		Sanity-Timer Flags - Indicates Sanity-Timer requested interrupt and boot commands have been received.	6	0-3	S1FO-S1F3	Sanity-Timer Interrupt Flag			Set to 1 when interrupt requests caused by the faulty-timer are received. (caused when Sanity-Timer times out and TIMEOUT pulse is generated.
				4-7	none	none	Always 0		Not Used.

Table 2-4 DIP11-A (IIST Interface Module) Register Bit Descriptions (Sheet 5 of 8)

				8-11	SBF	Sanity-Timer Boot Flag		Writing 1 (not affected by Reset)	Set to 1 when boot request caused by the Sanity-Timer received. **NOTE** **Any signal that sets a PIF, PBF, SIF or SBF bit clears lock-up condition. Since bits are not affected by RESET, Processor can identify source of BOOT after BOOT operation has been completed.**
				13-15	none	none	Always 0		Not Used.
DCF	Yes	DC LO Disconnect Flags - Indicates IIST Interface Module is disconnected or has lost power.	7	0-3	DCF0-DCF3	DC LO Disconnect Flags	Yes	Reset or by writing 1.	When 1 indicates interrupt is being requested because the respective processor (i.e., the IIST interface module) is either disconnected from the IIST BUS or else has lost power. Break condition is first detected when respective IM (Interrupt Module) bit is 0.
				4-7	none	none			Not Used.
				8-11	BRK0-BRK3	BREAK	Yes No		When 1 indicates actual "roll" disconnect or power loss condition of corresponding processor (IIST Interface Module)
				12-15	none	none	Always No 0		Not Used.
EXC	Yes	Exceptions - Indicates that error-free information packet was received from respective processor but that appropriate mask bit in IMSK register was not clear to enable action or that an information packet was received that had a parity or framing error.	10	0-3	RTE0-RTE 3	Register Transmission error.		Writing 1 or by	When 1, indicates that information packet was received from corresponding processor (RTE 0 - RTE 3 corresponds to CPUA - CPU B) but that it had a parity or framing error (incorrect number of stop bits). Setting of RTE bit causes program interrupt but does not clear lockup condition.
				4-7	none	none			Not Used.

Table 2-4 DIP11-A (IIST Interface Module) Register Bit Descriptions (Sheet 6 of 8)

EXC (Continued)			8-11	UI0-UI3	Unexpected Interrupt			Writing 1 or Reset	When 1, indicates that valid, error-free information packet was received from corresponding processor (UI0-UI3 corresponds to CPUA - CPUB) but that appropriate mask bit in the IMSK register was not clear to enable action, UI bits do not cause interrupts.
			12-15	none	none				Not Used.
MTC	Maintenance Control - contains bits controlling test and maintenance features of IIST Interface module. Bits can be used to: 1) Disable drives to IIST Interconnecting BUS 2) Enables maintenance data loop 3) Prevent BOOT commands from actually booting processors. 4) Select arbitrary Self-ID to allow full testing of 4 serial data receivers and their flags. 5) Force various transmission errors.	15	0	DSB DRV	Disable Driver	Yes	Yes	Reset	When 0, IIST transmitter drives Bus line normally. When 1, driver is disconnected causing receiving interface to detect "Break" condition.
			1	EN MLP	Enable Maintenance Loop	Yes	Yes	Reset	When 0, drivers operate normally and receivers receive data from IIST BUS lines. When 1, transmitting driver is idle (unless disabled) and receivers receive data directely from internal serial transmitter and internal path. For idle state, transmitting driver remains active but operates as if no data is being transmitted. During maintenance, therefore, tests can be performed without interfacing with other interfaces in the system.
			2	EN MXD	Enable Maintenance Data Mux	Yes	Yes	Reset	When 0, IIST interface operates normally and bits 8-11 of MTC register have no effect. When 1 bits 8-11 effect operation.
			3	DSB BT	Disable Boot	Yes	Yes	Reset	When 0, processor boots upon reception of boot command. When 1, Boot command only sets its flag in PGF or STF registers and processor is not booted.

Table 2-4 DIP11-A (IIST Interface Module) Register Bit Descriptions (Sheet 7 of 8)

MTC (Continued	4-7	none	none	Always 0	No	Not Used.
	8-9 V Field	MNTIDI and MNTID0	Maintenance ID			Two-bit field supplies self-identification to receivers when EN MXD is 1, overriding hardware switches. **NOTE** **Switches cannot be overriden for Transmission.**
	10	MNT FRM	Maintenance Framing Error	Yes	Yes Reset	Supplies framing bit to transmitted character. When MNT FRM is 1 and EN MXD is 1, a framing error is generated.
	11	MNT TTP	Maintenance Type	Yes	Yes Reset	Supplies type bit to transmitted character when EN MXD is 1, 0 for program transmission and 1 for sanity-timer transmission.

Table 2-4 DIP11-A (IIST Interface Module) Register Bit Descriptions (Sheet 8 of 8)

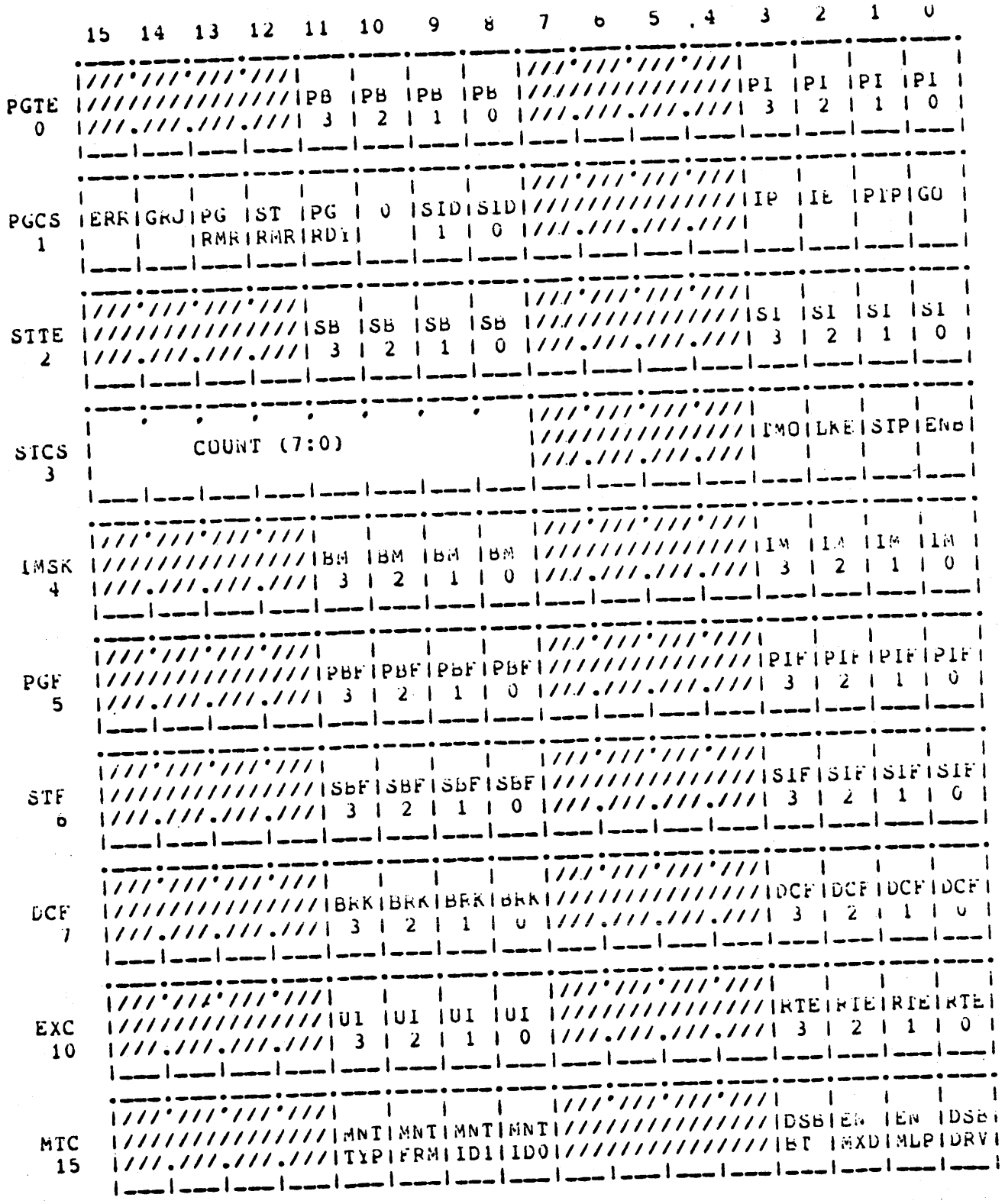
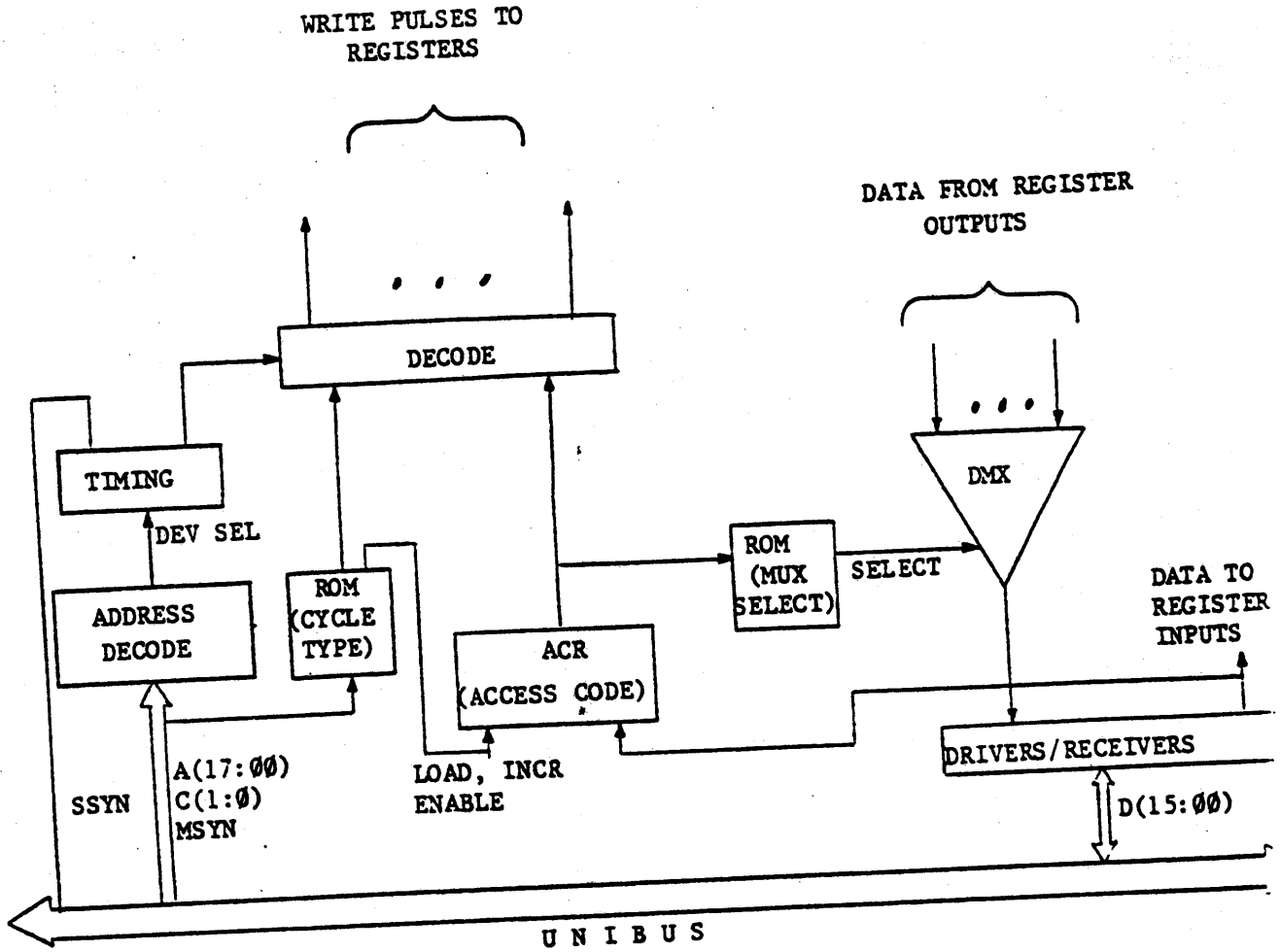


Figure 2-18
IIST Device Register Bit Configurations

FIGURE 2-19

REGISTER ACCESS PATH BLOCK DIAGRAM



The IIST examines Unibus address bits 02 through 17 to determine if the register being addressed on the Unibus is an IIST register.

The address field can be defined by a set of switches within the IIST. The Unibus address is compared with the set of switches and if the two match, the addressed register is a valid direct-access IIST register, enabling the circuitry for a register access function. If the Unibus address does not compare with the switches, the IIST will not accept the address and will not initiate a data transfer with the Unibus. (In actual fact, only address bits 02 through 12 are switch-selectable; bits 13 through 17 must all be 1 for an address to be recognized as a peripheral device address.)

2.6.14.6 IInterrupt/Boot Transmission Path

Figure 2-20 is a block diagram of the Interrupt/Boot Transmission Path. The diagram shows the flow of Interrupt/Boot request data from the Unibus, through the PGTE and STTE transmit enables registers, into the parallel-to-serial converter shift register, and through a driver out onto the IIST interconnecting bus. The following paragraphs describe the transmission of command data in detail.

2.6.14.7 Development of Transmitted Character

The way in which the transmitted character is developed from the STTE, PGTE, and STCS registers is shown in Figure 2-21.

2.6.14.8 Program-Generated Interrupt/Boot Transmission Cycle

Figure 2-22 is an interface diagram showing the action of the IIST during a program-generated transmission cycle. Initially, the program supplies data specifying which processors in the system are to receive interrupt and/or boot commands by writing into the PGTE register. This

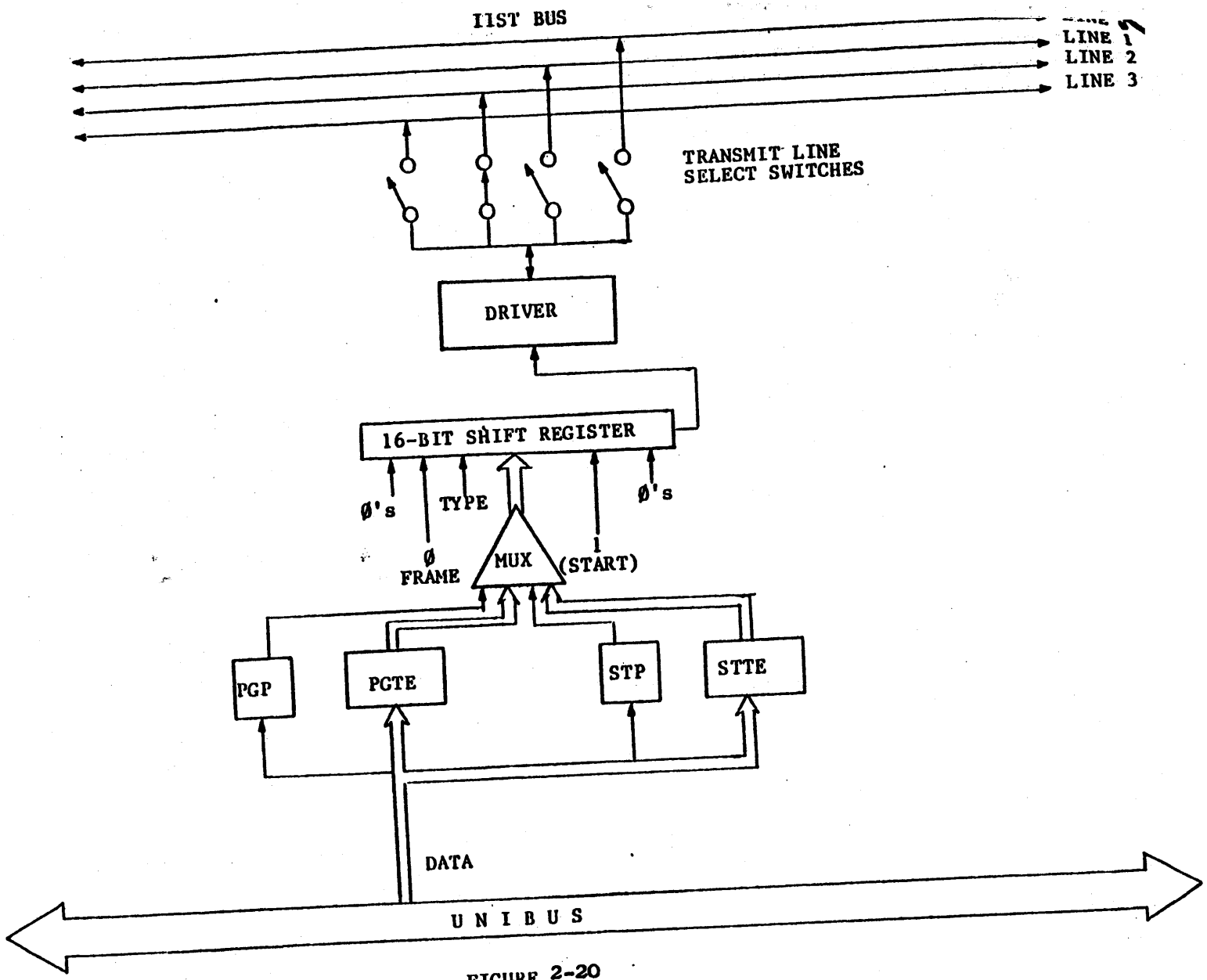


FIGURE 2-20

INTERRUPT/BOOT REQUEST TRANSMISSION PATH BLOCK DIAGRAM

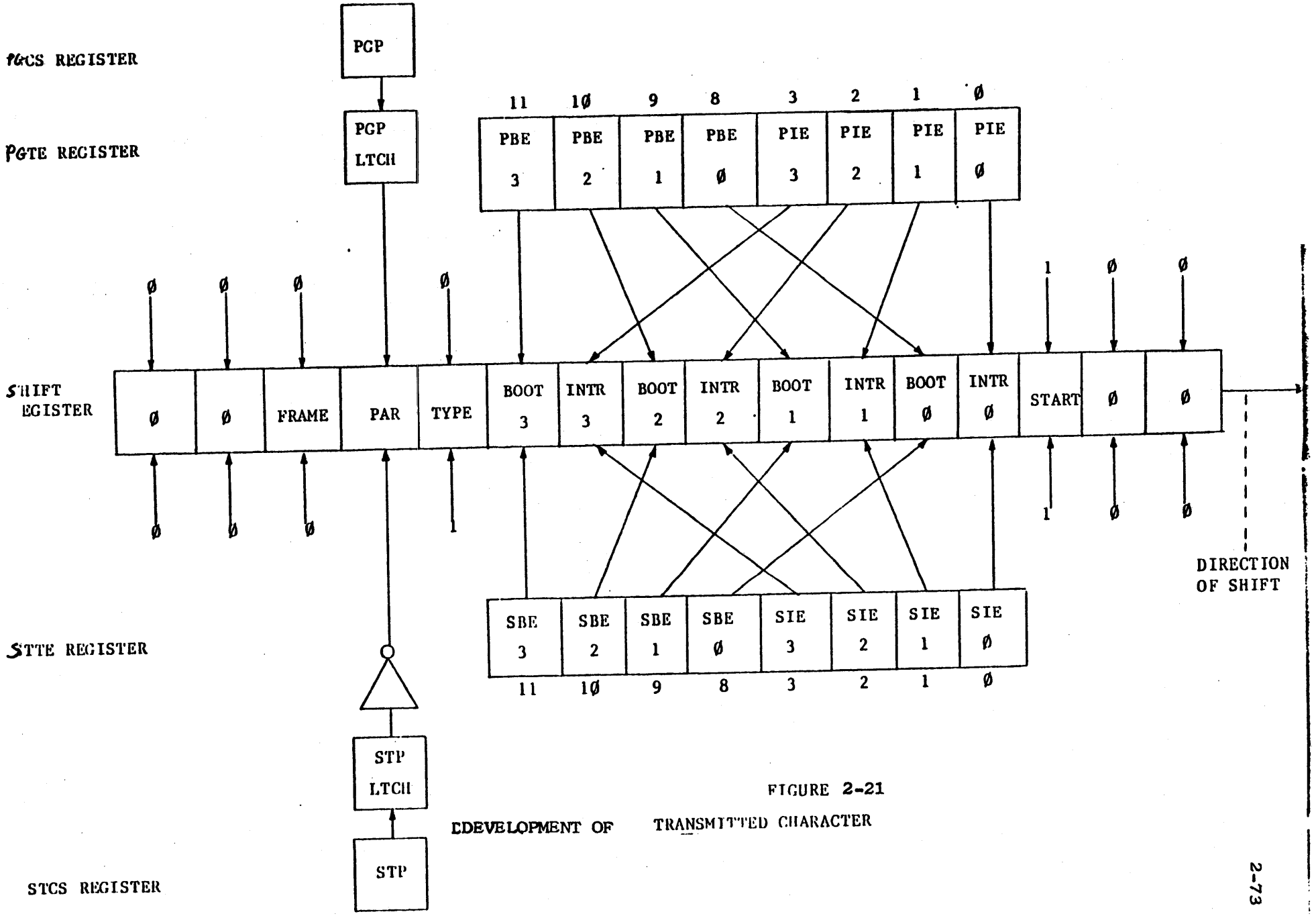


FIGURE 2-21

EDEVELOPMENT OF TRANSMITTED CHARACTER

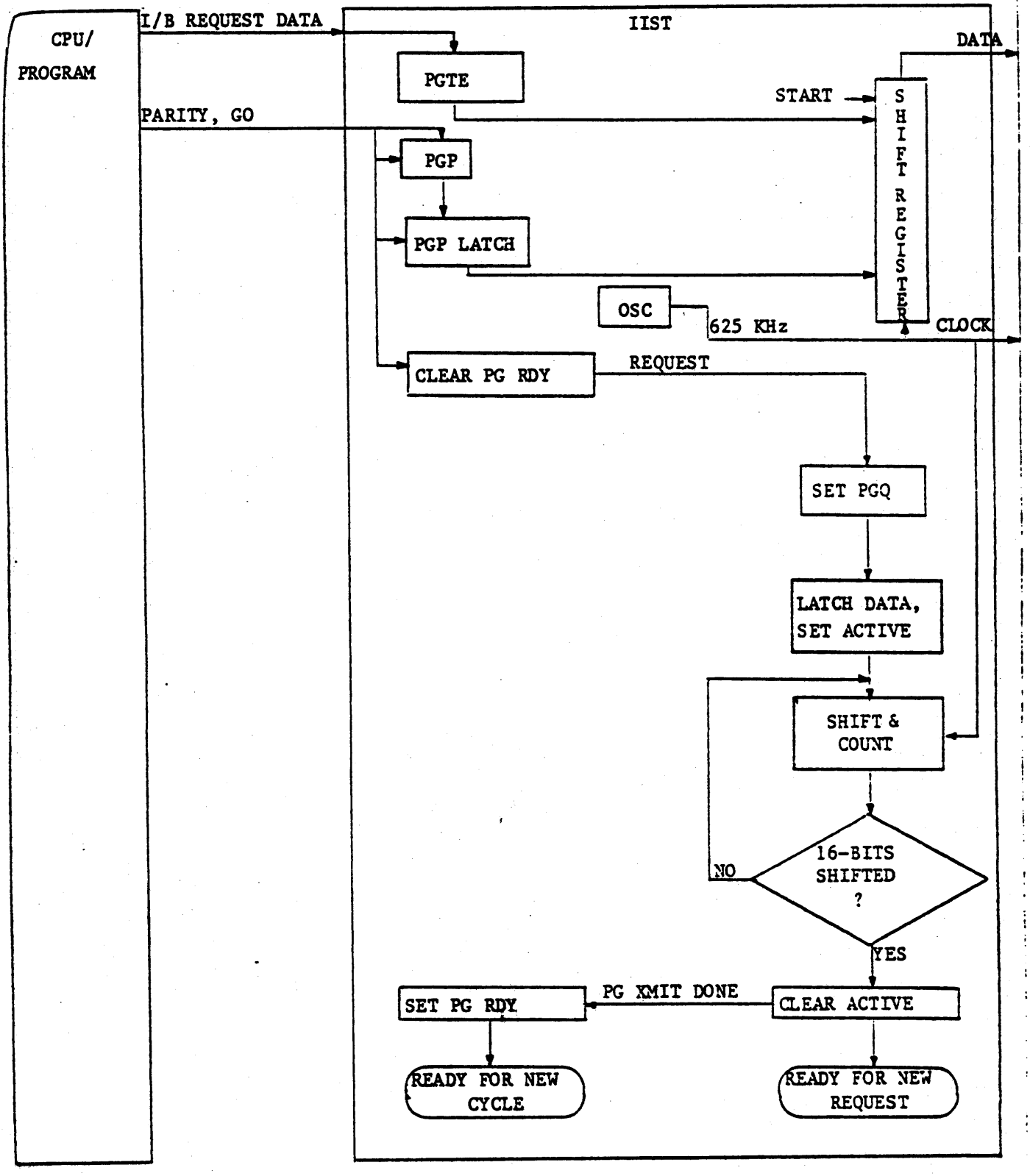


FIGURE 2-22

PROGRAM-GENERATED TRANSMISSION INTERFACE DIAGRAM

data will eventually be sent to all IIST interfaces in the system. The program computes the (odd) parity bit to be sent with this data and writes this result, along with the GO bit, into the PGCS register, starting the transmission cycle. The parity information is written into the PGP bit in PGCS. When the GO bit is set, the program-generated transmission logic becomes busy and cannot accept another data transfer command until the current data has been transmitted.

Setting GO clears the PG RDY bit, signifying that a transmission cycle is in progress. The serial transmission logic monitors the state of PG RDY. If the transmission logic is not busy (as from a transmission request from the Sanity Timer), the dropping of PG RDY is detected as a request to transmit the program-generated interrupt/boot data and parity. Upon recognition of the request, the serial transmission logic gates the data from PGTE and a copy of the PGP parity bit to the input of the serial shift register. Also supplied to the shift register input are leading 0's, a "start" bit (which TYPE bit (0 for a program-generated request), a framing bit (which is 0 unless forced to 1 by bit maintenance logic), and trailing 0's. In all, a character of 16 bits is constructed. After recognition of the request, the 16-bit character is loaded into the shift register, the ACTIVE flip-flop is set, and the shifting of the character onto the IIST bus, one bit at a time, begins.

A counter in the serial transmitter control logic keeps track of the number of bits shifted out of the shift register. After 16 bits have been shifted, a PG XMIT DONE pulse is sent to the program-generated request logic, clearing PG RDY and indicating the end of the transmission. The PGTE register can then be loaded again and another cycle requested. The serial transmission logic readies itself for a new

request by clearing ACTIVE.

The data shifted from the shift register is transmitted onto the IIST bus along with the clock pulses that cause the shifting to occur. The receiving interfaces can therefore use the clock pulses to accurately strobe the incoming data and assemble it into parallel characters. The serial data and clock streams are transmitted on only one of the four pairs of data and clock lines on the IIST bus, as selected by switches on the IIST interface module. The switches must correspond to the Self-ID of the interface module so that all interfaces in the system transmit on one and only one pair of bus lines.

2.6.14.9 Interrupt/Boot Request Reception Path

The Interrupt/Boot Request Reception Path provides the means by which the IIST interface receives characters transmitted by other IIST interfaces and uses the information to cause program interrupts and processor boots. Shown in Figure 2-23 is a block diagram of a serial receiver. The reception path consists of four serial line receivers (one per each set of data/clock lines on the IIST bus), a set of flags (set by the line receivers and monitored by the program), and flag control logic (which provides masks to control setting of the flags, logic to enable the program to selectively clear the various flags, and combining gates to provide the signals used to interrupt and boot the processor). Each of these elements is discussed in detail in the following paragraphs.

2.6.14.10 Serial Line Receivers

2.6.14.10.1 Basic Receiver Functions

Each serial line receiver is connected to a unique pair of data/clock

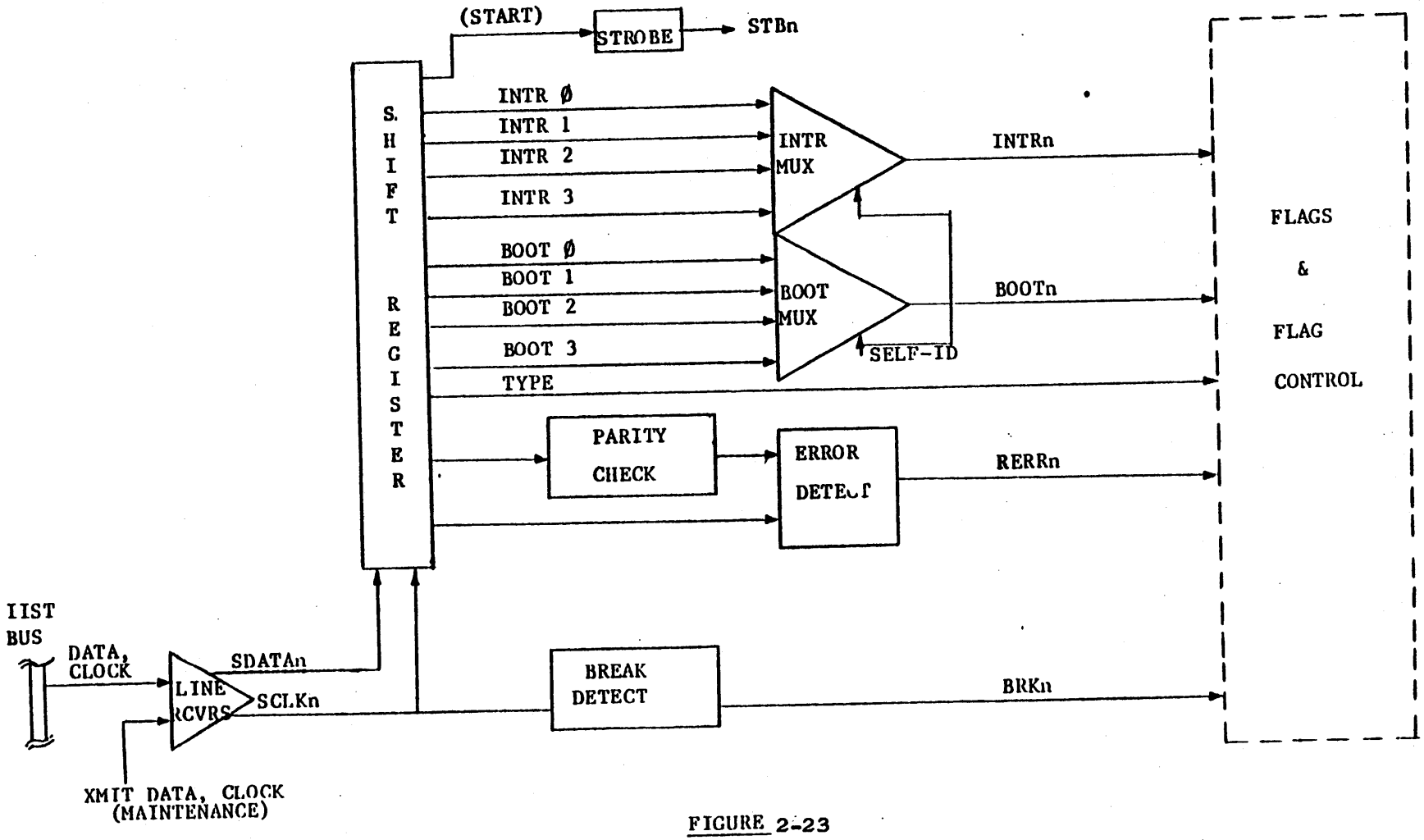


FIGURE 2-23

SERIAL RECEIVER BLOCK DIAGRAM

lines on the IIST bus. The receivers are numbered 0 through 3, corresponding to the line pairs to which they are connected. Since only one IIST interface transmits on any pair of lines, the receiver input connections serve to define the source (sending) interface for a received character. For example, the receiver connected to line-pair 2 receives only characters transmitted by IIST interface 2. This receiver, in turn, affects only the program-readable flags associated with processor 2.

Each serial line receiver, illustrated in the block diagram of Figure 2-23 performs the following functions:

1. Performs serial to parallel converson to assemble characters received on the IIST bus line to which it is attached.

2. Checks for valid parity and framing on the received characters.

3. If the character parity and framing are correct, the receiver uses the Self-ID (identification) code of the interface to select one of the four possible 2-bit INTR/BOOT fields of the character to determine which interrupt and/or boot flag associated with the line is to be set (contingent upon the state of the mask bits for the line).

4. If either character parity or framing are incorrect, the receiver sets its respective RTE (Received Transmission Error) flag and will take no action with respect to the normal interrupt or boot flags; characters with errors, therefore,

cannot cause a normal interrupt or boot to occur.

2.6.14.10.2 Summary of Receiver Operation

A summary of the events occurring at the IIST Interface Module during the reception of each serial character is as follows?

1. Before reception is started, the IIST bus and the serial receiver are assumed to be idle. That is, the Data line on the IIST bus is in the 0 state (Low) and the clock signal is continually being sent by the transmitting interface. In the receiving section, the shift register is clear, either because a previous break or reset condition has been released or because a clear pulse was generated at the end of reception of a previous character. Since the SCLK input is active and being applied to the shift register, the receiver's input shift register is actually shifting in a continuous string of 0's at this time.

2. The receiver will shift in data from the IIST bus as long as the most-significant bit of its shift register remains clear. Therefore, when the transmitting interface starts out of the transmission of a character, the serial data bits will flow out of the transmitter, over the IIST bus, and into the receiver's shift register. The transmitted character begins with two leading 0's, but these are indistinguishable from the normal 0's present in the idle state.

3. Eventually, the START bit (first non-zero bit) of the serial character will be shifted into the receiver (at the least-significant end) followed by the remainder of the character (INTR/BOOT data, Type bit, parity bit, and Frame bit), with all bits propagating from the least-significant toward the most-significant bit of the shift register. When the most-significant bit of the shift register becomes set, the entire character has been loaded into the shift register. The assertion of the most-significant bit prevents shift pulses from being further applied to the shift register. The output of the shift register is then sampled. If no error condition is present and if the selected INTR and/or BOOT bit in the shift register is set, a pulse is produced at the output of the INTR and/or BOOT multiplexers which sets the appropriate flag. The input shift register is then cleared and is ready to receive another character.

2.7 MODIFIED KB 11-C PROCESSOR

The main processor used in the PDP-11/70 Multiprocessor System is a modified KB11-C. The modified KB11-C processor is identical in operation to the unmodified KB11-C except for the following functions and capabilities:

1) Cache Flush--Under software control, it clears the cache by marking all of the words that are in the cache as invalid.

2) Cache Bypass--Data Cannot be read or written into the cache as all memory references by both the CPU and the Unibus Map are forced to main memory (although read or write hits will invalidate data at the hit locations in the cache). Data can be read or written only into main memory.
Cache Bypass can be turned off and on so that the address locations and amount of code references it effects can vary as follows:

2A) CACHE BYPASS ALL--Turned on for all code references without regard for their location in memory.

2B) CACHE BYPASS ONLY ON VIRTUAL PAGE--Turned on only for those memory references which fall within the virtual address space of a particular page.

2C) CACHE BYPASS ONLY ON UNIBUS MAP REFERENCE--Turned on only for those memory references which are mapped through particular mapping registers of the Unibus Map.

3) Ability to Access Main Memory CSR Registers--The CSR registers located on the main memory bus can be accessed directly.

4) Extension of Main Memory Timeouts have been increased and are now of longer duration for the modified KB11-C

than for the unmodified KB21-C.

5) Ability to Use ASRB Instruction as Lock--In addition to its normal ASRB function, the ASRB instruction can be used to implement a lock to prevent processors from gaining simultaneous access to certain shared locations of main memory.

2.7.1 Relationship Of Processors To Shared Memory

Shown in Figure 2-24 is a simplified Block Diagram of the PDP-11/70 Multiprocessor System showing the address, data, and control lines which interconnect the functional components of the system and the relationship of the Processor, Memory Management, Unibus Map and to the multi-ported memory. In the Multiprocessor System, the cache the Cache of each processor is connected by means of independent memory bus cables to an individual port on the multiported memory. Up to four processors can be connected to four ports on the memory. Some regions in memory are shared, others are private to each processor.

The data lines connecting the cache to the main memory and to the Massbus Controllers are 36 bits wide, and comprise 32 bits of data plus 4 parity bits. The remaining data lines are 16 bits wide.

The cache because of its function and position relative to the other functional components of the system, acts as a clearing house for all requests for access to Main memory. Requests for main memory access within each processor can come from three sources: processor, Unibus Map, and Massbus Controllers. When more than one of the above require memory access concurrently, priority in the code is given to the request according to the following structure:

TO BE SUPPLIED

1st priority: Unibus Map

2nd priority: Massbus Controllers

3rd priority: Processor

In addition, concurrent requests for memory access by Massbus Controllers are arbitrated in the Cache.

Once the access request is raised to the memory by the cache, however, actual access to main memory via the ports is controlled by the Multiplexor Board of the memory. Port contention (when two processors request access to memory via two different ports simultaneously) is resolved on a "round-robin" basis by the Multiplexor Board of the memory.

Since all ports have equal priority, when one or more ports make a request and the memory is ready to accept requests, an arbitration cycle is initiated. At the conclusion of the arbitration cycle the requesting ports are serviced round-robin in sequence regardless of which port made the first request, starting with the port immediately after the last port to have completed a memory cycle. After the memory has responded to all requests of one arbitration cycle, and while its servicing the last request, the memory becomes available to accept new requests and may start another arbitration cycle.

2.7.2 Cache Operation (Cache Bypass not generated)

When a 22-bit address arrives from the processor or Unibus, bits (9:2) (the index field) are immediately used as an index into the 256 by 30 bit Address Memory, which contains the high order bits (address field) of the addresses presently stored in the Cache

and their Valid bits. At the end of the Address Memory access time, two tags are available for use. Each tag consists of a 12-bit address field, a Valid bit, and two parity bits. The two address fields go directly to two comparators, where they are compared with the 12 high order bits of the incoming address. (The stored address tags are checked for correct parity while the address fields are compared. The following discussion assumes that no address tag parity errors are detected and that Cache Bypass is not generated).

2.7.2.1 Read Hit - Assume that one of the address comparisons results in a match and that the corresponding Valid bit is set. This condition is called a "hit" and means that the word requested is in the Fast Data Memory. The appropriate select signal is therefore sent to the Fast Data Memory.

Ten bits are required to select one of the 1024 words in Fast Data Memory. Eight of these bits are the same index bits used to index into the Address Memory. These bits select a set of two blocks, one block in each group. Also required is bit 1 of the incoming address, which selects either the high or low word of the blocks within the set. The last selection is provided by the comparison signal from the address field comparators, which determine which of the two groups (or equivalently, which of the two blocks within the set) has the desired data. After this last signal arrives, the data is available from the data memory and will be sent to the processor or Unibus as required.

2.7.2.2 Read Miss - If neither address field from the Address Memory matches the address field of the incoming address, then the requested data is not in fast memory. This is called a miss condition. When the Cache controller determines that there is no match, it must start a Main Memory cycle to fetch the required block. The block address will be the 20 high order bits of the incoming address.

During the Main Memory access time, the Cache controller can decide where to put the incoming data when it arrives. The index field determines which block within a group is replaced. The controller determines in which group the new block will be placed by examining an internally generated Random bit. When the data block arrives from Main Memory, it is written into the selected block of fast memory, while the requested word is passed along to the processor or Unibus. At the same time, the address field of the block is loaded into the corresponding location in Address Memory along with a set Valid bit. (A set Valid bit is loaded into Address Memory whenever the Fast Data Memory is loaded as a result of a read miss.)

2.7.2.3 Write Hit - During a write cycle initiated from the processor or Unibus, the initial sequence of events in the Cache is the same as during a read cycle: the address comes in, the Address Memory is accessed, and the address fields are compared. If the address fields match and the corresponding Valid bit is set, a hit is indicated and the new data is written into the

appropriate word or byte of fast memory, as selected by the index, word, and byte fields of the address and the comparator outputs. Since the PDP-11/70 Cache is implemented using write-through, the data is also written into Main Memory.

2.7.2.4 Write Miss - If, during a write operation from the processor or Unibus, a miss is indicated by the address comparison in the Cache, a write cycle is performed to the specified address in Main Memory. The contents of the Address Memory and the Fast Data Memory are left unaltered.

2.7.2.5 Cache Flush

2.7.2.5.1 Purpose - The caches of the processors in the Multiprocessor System are unsynchronized and operate independently of each other, while directly accessing a common or shared memory. The Cache Flush is used by the operating system to rid the cache of any stale or inappropriate data (data that does not accurately reflect the data contained in main memory at the same locations) each time a process is started or resumed. This function provides the ability to clear the local independent cache under software control, and prevents inappropriate or stale data from being present in the cache that might have been caused by process migration when process task execution may shift from one processor to another as process/system activity dictates, or by Indirect I/O (i.e., when the execution of a process that has requested an I/O transfer and the service of the I/O request occurs on different processors.

The result of the cache flush is to mark all of the words that may have been in the cache as invalid and later requires a refill of the cache at the appropriate times. Total cache invalidation is necessary to prevent stale data when processes are allowed to run as they are on the multiprocessor system serially on all CPU's. Because all context switching is performed by the Executive, the cache flush can occur without task intervention.

2.7.2.5.2 Typical Cache Flush Operation - The cache flush function is implemented by having two sets of valid bits, one which is in on-line use, and a second off-line that has been previously cleared. A typical cache flush operation that would prevent stale data from a cache from being transmitted is as follows:

1) Read Location X on CPU A

2) Issue CPU A Cache Flush Command (Upon receipt of the flush command) (shown being generated on logic print CCBX, M8182). The cleared offline set of valid bits is switched on line and the original set off line immediately invalidating all the words in the CPU A Cache. The original, now off line set of validation bits, is then cleared.

3) Write Location X on CPU B.

4) Issue CPU B Cache Flush Command. CPU B Cache is invalidated as described in step 2 above.

5) Read Location X on CPU A.

The cache flush performed by invalidating all the locations in the cache during the initial transition from CPU A to B ensures that the next reference to location X from processor A will correctly access main memory and not the cache. Variations to the above can occur, depending upon what conditions exist when the

Cache Flush command is issued. For example, if the clear cycle on the second valid set of bits has not yet been completed when a Cache Flush command is issued, the cache will be turned off until the second set of valid store bits has been cleared. When the off line set is cleared, it will be switched on line and the cache flush operation will resume. (This logic is presented on sheets CCBD, CCBF, and CCBX of the M8182 Board and ADMJ of the M8183 Board.) A summary of the various conditions that can exist as reflected in the Cache Control Register Bits when a Cache Flush command is issued and their effect upon the Cache Flush operation is shown in Table 2-4A.

TABLE 2- TRUTH TABLE FOR CACHE CONTROL REGISTER BITS EFFECTING CACHE FLUSH

CACHE CONTROL REGISTER BITS			TYPE OF CACHE FLUSH GENERATED			
INHIBIT VALID STORE SWITCHING (BIT 14)	VALID STORE IN USE (BIT 13)	FLUSH CACHE (BIT 8)	SWITCH ON-LINE SET OF VALID BITS WITH OFF-LINE SET	HARDWARE CLEARS ON-LINE VALID STORE SET OF BITS	HARDWARE CLEARS OFF-LINE VALID STORE SET OF BITS	HARDWARE CLEARS BOTH ON-LINE AND OFF-LINE SET OF VALID BITS (DOUBLE FLUSH)
0	(0-Valid Set A in use On-Line) Complemented by software- 0 to 1	1	Valid set B switched On-Line; A switched Off-Line		X	
0	(1-Valid Set B in use On-Line) Complemented by software- 1 to 0	1	Valid set A switched On-Line; B switched Off-Line		X	
1	Complemented by software (1 to 0, or 0 to 1)	1				X
1	0-Valid Set A in use On-Line (not complemented)	1		On-Line set A of Valid bits cleared		
1	1-Valid set B in use On-Line (not complemented)	1		On-Line set B of Valid bits cleared		

2.7.2.6 Cache Bypass - This function permits the software to force all memory references from the Processor or Unibus to main memory. Data is not taken from or copied into the cache, although read or write hits will invalidate cache data at the hit locations in the cache. It allows shared data in the memory to be accessed by several parallel processes without fear of stale data appearing in any of the caches and reduces the amount of cache flushes required every time a processor is dealing with a shared data memory region.

The purpose of a cache bypass is to prevent stale data in a cache (without resorting to a cache flush) when simultaneous processes use common shared data in memory. It ensures that an accurate copy of the current data in main memory will be obtained when needed. When a cache bypass is implemented, all memory references are forced to main memory, and the cache does the following: checks the tag store for a hit or a miss, if a miss occurs the cache does nothing, if a read or write hit occurs, the cache invalidates that location in the Tag store of the cache. Read or write misses will not change the contents of the cache.

Bypass can be turned on for different lengths of time and made to apply to either whole sections of addresses or only a page or pages of contiguous virtual or Unibus mapped addresses, depending upon program requirements. The three specific variations under which Cache Bypass occurs (the bypass effect is always the same, i.e., all references are forced to mainmemory, but the sections of code

that the cache bypass will apply to can vary depending upon where the references are going) are: CACHE Bypass ALL, Cache Bypass only on a Virtual Page, and Cache Bypass only on a Unibus Map reference.

2.7.2.6.1 Cache Bypass All - This function is used for extremely short sections of code that must have access to the current contents of the shared memory in situations where a Cache Flush would impact processing speed. Bypass in this case is untestricted, i.e., it applies to all references no matter where the code is coming from or going to in memory. It is used for short segments of code, such as interrupt service routines, that must have guaranteed access to main memory. This function is controlled by bit 9 of the Cache Control Register.

2.7.2.6.2 Bypass Only On A Virtual Page - It is possible to specify that when a Page Descriptor Register (PDR) is programmed to map a virtual page into a physical page all memory references to that page will bypass the cache. Limiting the Bypass to specific locations in memory preserves that contents of the cache (as much as possible) so that maximum cache advantage can be realized for subsequent cache operations. It permits shared read-write data to be non-cached while still maintaining the advantages of the cache for all other data and instruction regions in memory. In this case performance degradation is reduced by confining the virtual address space over which cache bypass is effected to the region containing shared data.

Cache Bypass only on a virtual page is controlled by BYP bit 15 in the Page Descriptor Register (PDR). When this bit is set and relocation is enabled, any CPU references to the virtual page mapped by that PAR/PDR pair will go directly to main memory. This logic is generated on the System Address Path Board, M8177 sheets SAPD, SAPE, SAPF and SAPJ, from where it is sent to the cache control board to initiate the cache bypass cycle.

2.7.2.6.3 Bypass Only On A Unibus Map Page - This bypass is performed only on Unibus transfers through a page of the Unibus map when eighteen bits are being converted to a twenty-two bit physical memory addresses. The control of a bypass on reference though the Unibus Map is identical to that of a Bypass on a Virtual Page except that the control bit for the bypass of the Unibus Mapping Page is a bit in the high word of each Unibus mapping register (BYP-bit 15). When this bit is set in the High Word of the Mapping Registers and mapping is enabled any Unibus memory references through the mapping register in which bit 15 is set will intiate a cache bypass cycle. This logic is generated on the Unibus Map board M8181, MAPD; from where it is sent to the cache control board to complete the cache bypass cycle.

2.7.2.6.4 Summary Of Cache Bypass Operations - A summary of Cache Bypass operations showing the effect on the Cache with Cache Bypass turned off and on is shown in Table 2 -4B.

TABLE 2-4B EFFECT ON CACHE OF READ AND WRITE HITS AND MISSES

SOURCE OF CYCLE	TYPE OF OPERATIONAL CYCLE	EFFECT ON CACHE	
		CACHE BYPASS OFF	CACHE BYPASS ON
CPU OR UNIBUS	Read Miss	Updated	No Change
	Read Hit	No Change	Invalidated
	Write Miss	No Change	No Change
	Write Hit	Updated	Invalidated
Massbus	Read Miss	No Change	No Change
	Read Hit	No Change	Invalidated
	Write Miss	No Change	No Change
	Write Hit	Invalidated	Invalidated

2.7.3 Use of ASRB to Test and Set Memory Lock (If Unlocked)

2.7.3.1 Purpose

In order to synchronize and serialize access to blocks of critical shared data locations in memory and guarantee that these blocks in memory will be accessed by only one processor at a time, a new lock implementation function has been added to the standard PDP-11/70 ASRB instruction. This ASRB lock function is used together with cache bypass and the DATIP cycle of the MKA11 memory to test and set (assuming that the semaphore or flag in the lock byte at the memory destination location was found to be unlocked) the lock byte that controls access and ownership of a block of shared memory data locations.

2.7.3.2 Detailed Discussion

For example, in order to execute their tasks, two tasks running on two different processors might require access to shared data locations (the shared data locations could contain certain portions of the executive operating code containing I/O routines, data dependent routines, etc.,) . Blocks of shared data memory locations have a lock byte associated with them which controls access and ownership of that particular block of memory locations. For a processor to access and own these data locations, the processor must first test the lock byte to see if that block of memory is accessible, i.e., in a locked or unlocked state or, in other words, to check if it is presently owned by another processor. Then, if it not owned by another processor, to set the lock byte in order to indicate that it is now owned by the testing processor. If the check on the lock byte indicates that it is locked or owned by another processor (blocked lock) then the testing processor will not be able to access or own those data locations and will have to wait until the owning processor relinquishes his lock on them by setting the flag in the lock byte to an unlocked condition (1).

Test and set of the lock byte is accomplished during a DATIP (read-modify-write) uninterruptible cycle to memory. During the DATIP cycle, all other processors are locked out from memory in order to permit the testing processor to test and set the destination or lock byte of a particular block of shared data memory locations without interference from other processors. Cache bypass is also in effect during the read-modify-write (DATIP-DATO) of the destination or lock byte operand in order to ensure that the lock byte is read from and written into main memory without recourse to the cache.

2.7.3.1 SIMPLIFIED VIEW OF ASRB TEST AND SET OF MEMORY LOCK

Assume that Task A is running on CPU A and Task B on CPU B. Task a issues a directive which requires the use of some executive code whichis contained in a shared data region of memory whose access is controlled by a lock byte. CPU A enters Kernel state and is vectored to the shared data region containing the Executive code. Before accessing the shared data region, the processor must gain ownership of the block of memory data it desires to access. It does this by a Lock/Unlock routine involving the new function of the ASRB instruction.

Processor A issues an ASRB instruction (a P class instruction which is always followed by a BCS (Branch if Carry bit C conditioncode of the Processor Status Words is 1 after ASRB has been executed). The ASRB is ANDED with the DATIP signal on the Cache Control Board. The ANDING of ASRB with the DATIP signal causes the cache control board to raise MAIN C1 L to the memory indicating that the type of memory cycle being initiated is a DATIP (the DATIP cycle is a read that is always followed by a write). Whenever a DATIP cycle is indicated to the memory, a PAUSE/LOCK flip-flop is set on the Multiplexor board of the memory in order to ensure that no other processor can access memory through another port while a processor is testing and setting (if destination in memory was found unlocked) the lock byte or

destination operand. (The PAUSE/LOCK flip-flop is set at the initiation of the Read-Modify-Write cycle (DATIP) and is not cleared until completion of the write.)

The lock byte is now read from memory into the processor and the ALU performs an ASRB on the lock byte whose least significant digit is a flag or semaphore indicating the locked or unlocked state of the lock byte. (0 equals locked; 1 equals unlocked.) As a result of the execution of the ASRB, the condition code C bit (Carry) of the Processor Status Word of Processor A is set to 1 or 0 depending upon the state of the least significant bit of the lock byte or destination operand. If it is 1, the C bit becomes 1; if it is 0, the C bit becomes 0. Assuming that the least significant digit of the lock byte is 1 (unlocked) then the C bit of the PSW word becomes set or 1 upon execution of the ASRB by the ALU indicating there was a carry, and that the location controlled by the lock byte is not locked or owned by another processor. Processor A is then able to execute a BCS branch (branch if carry bit set to 1) . Processor A writes the now changed lock byte (its least signfiicant digit is now 0 indicating a locked state, i.e., owned by Processor A) back to the lock byte location in memory . Processor A can n ow access and execute the desired code in the shared data locations. The DATIP cycle is now completed and the PAUSE/LOCK flip-flop is cleared. permitting the Multiplexor to return ACK and service requests from other processors in round robin fashion.

Now if Processor B attempts to access the same shared data location that has been locked by Processor A (least significant digit or flag of lock byste is o) he will not be permitted to do so. When he issues the ASRB and goes through th the test and lock/unlock routine with the lock byte his C condition code bit in his PSW will be 0, therefore he will be unable to branch to execute the desired code. As long as processor A does not relinquish his lock on that memory location by setting the flag in the lock byte to a 1, processor

B will be unable to access those locations in memory. While waiting for the blocked lock to clear, Processor B may perform other operations such as restoring old priorities although operations of this type are determined by the kind of software that is running.

When processor A has finished with the shared data routine, he will execute a lock/unlock routine which cause the least signfificant digit or flag of the lock byte to become 1 unlocking it. Thus when Processor B tests and sets the lock byte he will be able to unlock it , establish ownership of that block in memory and branch to execute the desired code.

When the ASRB instruction is not used as part of the special test and lock/unlock routine (i.e., anded with the DATIP signal) it retains its identity as a normal Arithmetic Shift Right Byte Instruction. A simplified logic diagram showing the Lock/Unlock routine is contained in Figure 2-24A.

TO BE SUPPLIED

Figure 2-24 Simplified Logic Diagram of ASRB Lock/Unlock Routine

TO BE SUPPLIED

Figure 2-24A (Sheet 2 of 2)

2.7.4 Accessing Main Memory Control Status Registers (CSR's) - The MKA11 Multiported memory has Control and Status registers (CSR's) located on the main memory bus. These registers are used to store operating mode commands and error information. There are two CSR's in each memory box. The selection of the CSR memory box addresses is determined by switches on the memory Address Interface Module. (Refer to Table2-5). Each memory box must have a unique CSR addresse ; no two memory boxes should have the same CSR address. In the multiprocessor system each CPU has its own I/O page and interrupt vector addresses. It is possible, therefore, for devices to have the same CSR or vector addresses if they reside on different processor busses. Through the CSR the program can control the operation of the memory and read error messages for maintenance purposes. In the standard PDP-11/70 system the addresses of these registers are located in the upper 4k page references of memory. (The upper 4k is normally used for I/O device registers and CPU registers.) In the multiprocessor system however, the Unibus Map and Address Memory Board have been modified to map I/O page references to the Memory Parity CSR addresses (772100 through 772136) to addresses on the main memory bus (160772100 through 160772136). This is accomplished by permitting the Unibus map to decode addresses in this range as

ADDRESS	MEMORY BOX (PORT BUFFER MODULE) SWITCHES		
	S1-1	S1-2	S1-3
177772100 THROUGH 177772102	ON	ON NOTE: 1=ON 0=OFF	ON
177772104 THROUGH 177772106	OFF	ON	ON
177772110 THROUGH 177772112	ON	OFF	ON
177772114 THROUGH 177772116	OFF	OFF	ON
177772120 177772122	ON	ON	OFF
177772124 THROUGH 177772126	OFF	ON	OFF
177772130 17772132	ON	OFF	OFF
177772134	OFF	OFF	OFF

Table 2-5, Switch Settings for Memory Port CSR Address Selections

main memory addresses and request a cache bypass cycle. When the CSR address is received, the cache will assert address lines (bits 22 through 24) as 1's leaving address lines (bit 18 through 21) as 0's and pass through Unibus addresses (bits 00 through 17) unchanged as main memory bits 00 through 17. If a cache hit occurs the contents of that location in cache will be invalidated. In cases where all 16 CSR addresses are not required, the Unibus Map Board contains switches which can be used to select the CSR addresses to which the Map will respond. The switches can be set to permit the Map to respond to either first 2, 4, 8 or all 16 CSR addresses. An eight box maximum system would, however, require the use of all 16 CSR addresses as 2 CSR addresses are required for each box of memory. (See Table 2-6.)

Each memory box has a unique CSR address. The selection of the CSR address is determined by switches on the Boat. Buffer module. (See Table 2-5)

NOTE:
Switch 1-8 on M8158 is off.

2.7.5 Main Memory Timeouts

The main memory timeouts for the MKA11 memory are 20 microseconds. These timeouts are longer than ones used with the unmodified processor and the single-ported memory. The time-out period for main memory transfers was increased to allow for the multi-ported memory's worst case access time (approximately 20 us.) which is longer that that of the PDP-11/70's single ported memory. Increasing the time-out period was accomplished by lengthening the main memory time-out flip-flop's cycle as shown on logic print CCBD, M8182.

NUM MKA11 MEMORY BOXES	CSR ADDRESSES USED	CSR ADDRESSES MAPPED	CSR ACCESS ADDRESS FROM CONSOLE	UBUS BOARD MEMORY CSR ADDR SELECTION SWITCHES AND SETTINGS					
				S1-6	S1-5	S1-4	S1-3	S1-2	S1-1
1	772100 AND 772102	772100 AND 772102	177772100 177772102	OFF	ON	OFF	ON	OFF	ON
2	772100 THROUGH 772104	772100 THROUGH 772104	177772100 THROUGH 177772104	OFF	ON	OFF	ON	OFF	OFF
3	772100 THROUGH 772112	772100 THROUGH 772116	177772100 THROUGH 17772116	OFF	ON	OFF	OFF	OFF	OFF
4	772100 THROUGH 772116	772100 THROUGH 772116	17772100 THROUGH 177772116	OFF	ON	OFF	OFF	OFF	OFF
5	772100 THROUGH 772122	772100 THROUGH 772136	177772100 THROUGH 177772136	OFF	OFF	OFF	OFF	OFF	OFF
6	772100 THROUGH 772126	772100 THROUGH 772136	177772100 THROUGH 177772136	OFF	OFF	OFF	OFF	OFF	OFF
7	772100 THROUGH 772132	772100 THROUGH 772136	177772100 THROUGH 177772136	OFF	OFF	OFF	OFF	OFF	OFF
8	772100 THROUGH 772136	772100 THROUGH 772136	177772100 THROUGH 177772136	OFF	OFF	OFF	OFF	OFF	OFF

Table 2- 6Unibus Map Board CSR Memory Address Selection Switches and Settings

2.7.6 Processor Registers

The modified KB11-C processor contains registers that are identical to the unmodified KB11-C. They control processor operations or provide information relative to these operations. These registers are listed below along with a brief summary of their operation and purpose.

Address	Register
17 777 570	Switch and light Registers
17 777 760	Lower Size Register
17 777 762	Upper Size Register
17 777 764	System ID Register
17 777 766	CPU Error Register
17 777 770	Microprogram Break Register
17 777 772	Program Interrupt Request Register
17 777 774	Stack Limit Register
17 777 776	Processor Status Word

For complete details refer to Section II, Chapter 3, of the KB11-C Processor manual, including the multiprocessor change to it.

Processor Status Word 17 777 776

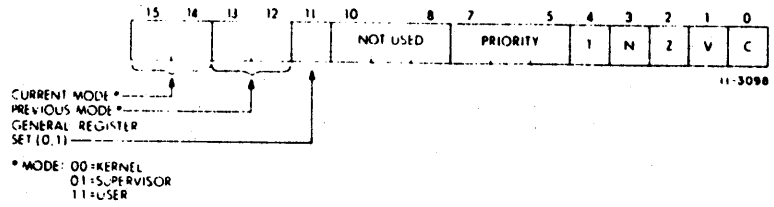


Figure 2-25 PSW Register

15-14: Current Mode

Specifies the current processor mode as follows:

1) When PS(15:14) = 0, the processor is in Kernel mode; all operations are legal.

2) When PS(15:14) = 01, the processor is in Supervisor mode; HALT, RESET, and SPL instructions either trap to location 4 (HALT) or are treated as NOP (RESET and SPL); SUPER address space is used if memory management is enabled.

3) PS(15:14) = 10 is an illegal mode; if memory management is enabled, a memory management abort occurs (refer to Section IV of the KB11-C Processor Manual).

4) When PS(15:14) = 11, the processor is in User mode; HALT, RESET, and SPL instructions either trap to location 4 (HALT) or are treated as NOP (RESET and SPL), USER address space is used if memory management is enabled.

13-12: Previous Mode

Specifies the processor mode prior to the last trap, interrupt, or loading of the PS.

11: Register Set

Specifies which general register set is used; if PS11 = 0, register set 0 is selected; if PS11 = 1, register set 1 is used.

10-08: Unused

07-05: Priority

Sets the processor priority; this priority determines which levels of programmed and external device interrupt requests are honored.

04: Trace

When PS04 = 1, the processor traps to the trace trap vector address (14 octal) after each instruction fetch; this facility is used to debug programs.

03: N Condition Code

This bit is set when the result of the last data manipulation is negative.

02: Z Condition Code

This bit is set when the result of the last data manipulation is 0.

01: V Condition Code

This bit is set when the result of the last data manipulation causes an arithmetic overflow.

00: C Condition Code

This bit is set when a carry occurs during data manipulation.

Stack Limit Register 17 777 774

The stack limit allows program control of the lower limit for permissible stack addresses. This limit may be varied in increments of 400 (8) bytes or 200 (8) words, up to a maximum address of 177 400 (almost the top of a 32K memory).

The normal boundary for stack addresses is 400. The stack limit option allows this lower limit to be raised, providing more address space for interrupt vectors or other data that should not be destroyed by the program.

The Stack Limit register has the following format:

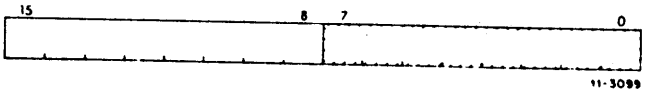


FIGURE 2-26 SL Register

The Stack Limit register can be addressed as a word at location 17 777 774, or as a byte at location 17 777 775. The register is accessible to the processor and console, but not to any bus device.

The 8 bits, 15 through 8, contain the stack limit information. These bits are cleared by INIT. The lower 8 bits are not used. Bit 8 corresponds to a value of 400 (8) or 256 (10).

Program Interrupt Request Register (PIR) 17 777 772

A request is booked by setting one of the bits 15 through 9 (for PIR 7 - PIR 1) in the Program Interrupt register at location 17 777 772. The hardware sets bits 7-5 and 3-1 to the encoded

value of the highest PIR bit set. This Program Interrupt Active (PIA) should be used to set the processor level and also index through a table of interrupt vec for the seven software priority levels. The following figure shows the layout of the PIR register.

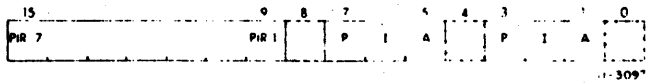


Figure 2-27 PIR Register

When the PIR is granted, the processor will trap to location 240 and pick up PC in 240 and the PSW in 242. It is the interrupt service routine's responsibility to queue requests within a priority level and to clear the PIR bit before the interrupt is dismissed.

Microprogram Break Register 17 777 770

This register is used for maintenance purposes only. It is used with maintenance equipment to provide timing synchronization and testing facilities. The microaddress in this register generates a sync pulse at T1 at pin F13K2 each time the cycle corresponding to the address is executed (TIGB PB SYNCH H); a pulse whose width is the same as that of the cycle is also generated at A10E1 (PDRC PB CMP H).

CPU Error Register 17 777 766

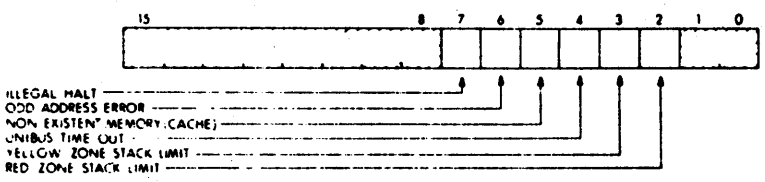


Figure 2-28 CPU Error Register

This register identifies the source of the abort or trap that used the vector at location 4.

7: Illegal Halt

Set when trying to execute a HALT instruction when the CPU is in User or Supervisor mode (not Kernel).

6: Odd Address Error

Set when a program attempts to do a word reference to an odd address.

5: Non-existent Memory

Set when the CPU attempts to read a word from a location higher than indicated by the System Size register. This does not include Unibus addresses.

4: Unibus Timeout

Set when there is no response on the Unibus within approximately 10 μs.

3: Yellow Zone Stack Limit

Set when a yellow zone trap occurs.

2: Red Zone Stack Limit

Set when a red zone abort occurs.

System I/D Register 17 777 764

This read only register contains information uniquely identifying each system. **It is read on the Internal Data Bus (INTD).**

Upper Size Register 17 777 762

This register is an extension of the lower size register, which is reserved for future use. It is read only and its contents are always read as zero.

Lower Size Register 17 777 760

This read only register specifies the memory size of the system. It is defined to indicate the last addressable block of 32 words in memory (bit 0 is equivalent to bit 6 of the Physical Address).

Refer to the KB11-C Processor Manual, Section IV, Chapter 5.

CPU General Registers 17 777 717 - 17 777 700

The general registers can be used as accumulators, index registers, auto-increment registers, auto-decrement registers, or as stack pointers for temporary storage of data. Arithmetic operations can be from one general register to another, from one memory or device register to another, or between memory or device register and a general register.

R7 is used at the machine's program counter (PC) and contains the address of the next instruction to be executed. It is a general register normally used only for addressing purposes and not as an accumulator for arithmetic operations.

The R6 register is normally used as the processor stack p inter indicating the last entry in the appropriate stack (Kernel stack,

Supervisor stack, and User stack). When the central processor is operating in Kernel mode it uses the Kernel stack, in Supervisor mode, the Supervisor stack, and in User mode, the User stack. When an interrupt or trap occurs, the PDP-11/70 automatically saves its current status on the Processor stack selected by the service routine.

The remaining 12 registers are divided into two sets of unrestricted registers, R0-R5. The current register set in operation is determined by the processor status word bit 11.

Refer to the KB11-C Processor Manual, Section II, Paragraphs 2.1.3 and 2.1.4.

2.7.7 Cache Registers

Table 2-7lists the six registers located in the Cache along with their addresses. A simplified block diagram of Cache register logic is shown in Figure 2-29.

Table 2-7
Cache Registers

Register	Address	Access
Low Error Address	17 777 740	Read only
High Error Address	17 777 742	Read only
Memory System Error	17 777 744	Read/selective clear
Control	17 777 746	Read/write
Maintenance	17 777 750	Read/write
Hit/Miss	17 777 752	Read only

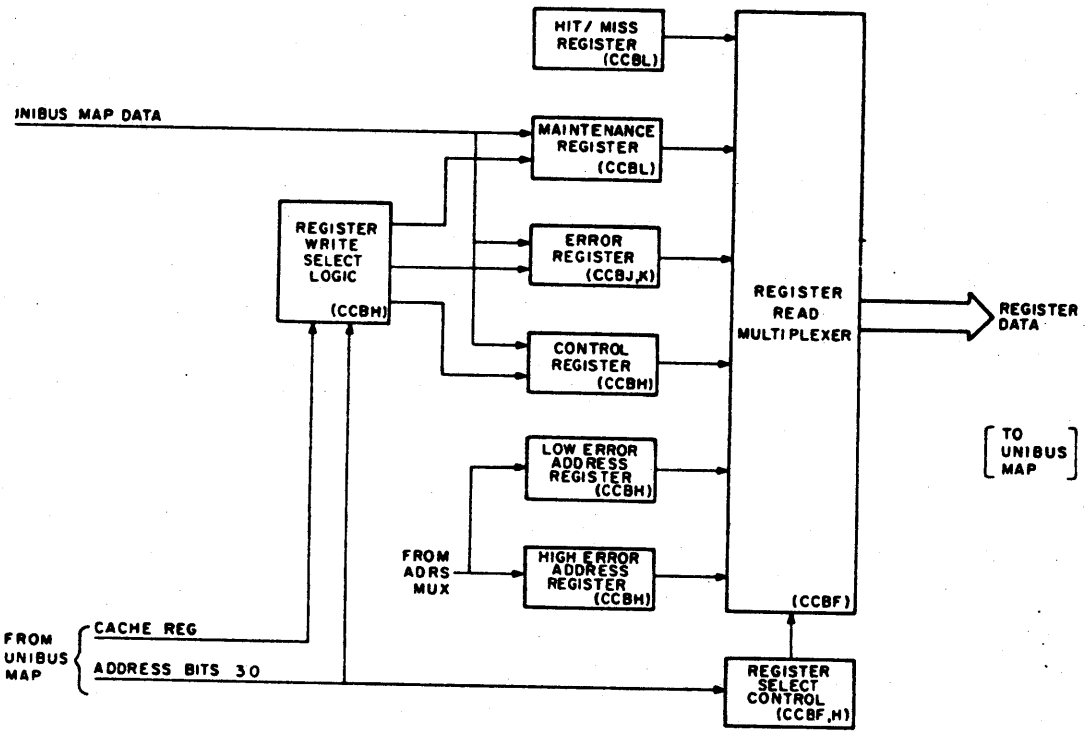


FIGURE 2-29 Block Diagram of Cache Register Logic

2.7.7.1 Low Error Address Register 17 777 740 - This register, illustrated in Figure 2-30, contains the 16 low order bits of the 22-bit physical address being accessed when an error occurred. The least significant bit is bit 0. The high order bits of the addresses are contained in the High Error Address Register.

All bits are read only. The bits are undetermined after a power-up. They are not affected by a Console Start of RESET instruction.

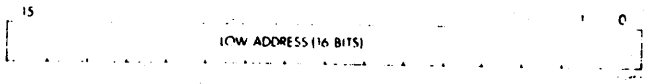


FIGURE 2-30 Low Error Address Register

2.7.7.2 High Error Address Register 17 777 742 - This register, illustrated in Figure 2-31 contains the six high order bits of the 22-bit physical address being accessed when an error occurred. The type of memory cycle being performed when the error occurred is indicated by register bits 15 and 14, which store the operation control bits (C1 and C0) of the memory cycle. Table 2-8 lists the register bits.

Bit 15 permits logging the extended main memory address used to access the MKA11 CSR's when an error occurs. This bit sets to 1 only if a memory error occurs during an access to the MKA11 CSR registers. When this bit is 1, it indicates that the lower order bits of the High Address register along with the contents of the LOW ERROR ADDRESS REGISTER define the exact CSR location that was being accessed when the error occurred.

All the bits are read only. The bits are undetermined after a power-up. They are unaffected by a Console Start or RESET instruction.

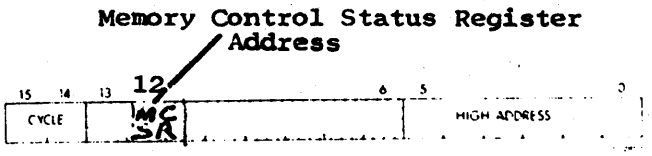


Figure 2-31 High Error Address Register

TABLE 2-8
High Error Address Register

BIT	NAME	FUNCTION
0-5	Address	Highest 6 bits of the 22-bit address of the first error. Bit 5 is the most significant bit.
6-11		Not used
12	Memory Control Status Register Address	Used only if a memory error occurs during access to MKA11 CSR addresses. When set, indicates that the lower order bits of the High Address register along with the contents of the Low Error Address Register define the exact CSR location that was being accessed when the error occurred.
13		Not used
14-15	Cycle Type	Indicates type of memory cycle which was being requested when the parity error occurred. Types of memory transmission cycles are:

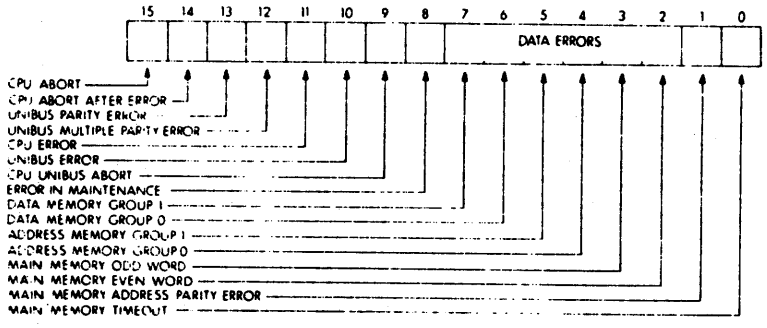
BIT 15	BIT 14	CCYLE TYPE
0	0	Data In (Read with restore)
0	0	Data In Pause (Read without restore. Always followed by write.)
1	0	Data Out (Word write.)
1	1	Data Out Byte (Word Byte write.)

2.7.7.3 Memory System Error Register 17 777 744 - The Memory System Error Register, illustrated in Figure 2-32, keeps track of hard and soft errors within the memory system.

A soft error is an error which does not result in the processor receiving erroneous data: a soft error causes a trap. An error which causes the processor to receive erroneous data is a hard error; this type of error causes an abort.

Table 2-9 defines the bits in the Memory System Error Register. All the bits are read/write. The bits are cleared on power-up or by Console Start. They are unaffected by a RESET instruction.

When writing to the Memory System Error Register, a bit is unchanged if a 0 is written to it, and it is cleared if a 1 is written to it. Thus, the register is cleared by writing the same data back to the register. This guarantees that if additional error bits were set between the read and the wirte, they will not be inadvertently cleared.



32
FIGURE 2- MEMORY SYSTEM ERROR REGISTER

Table 2-9
Memory System Error Register

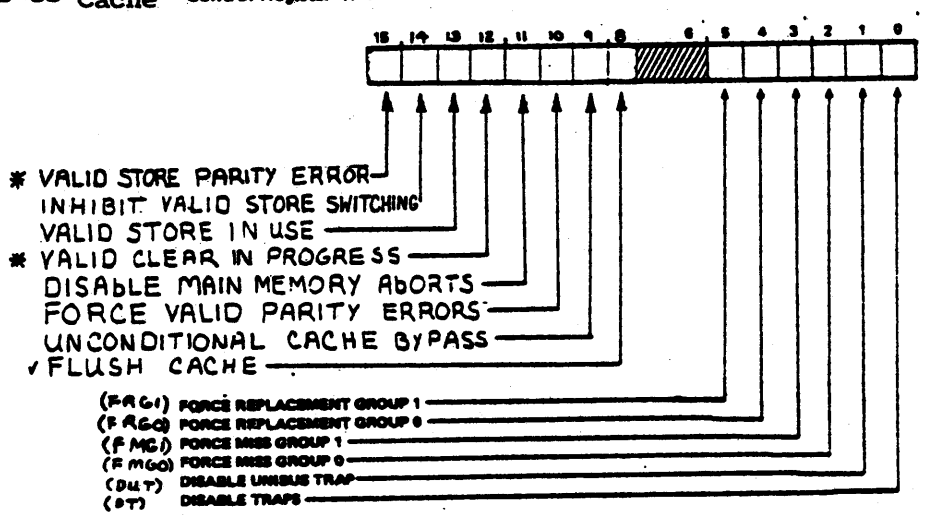
Bit	Name	Function
15	CPU Abort	Set if an error occurs which causes the Cache to abort a processor cycle.
14	CPU Abort After Error	Set if an abort occurs with the Error Address Register locked by a previous error.
13	Unibus Parity Error	Set if an error occurs which results in the Unibus Map asserting the parity error signal on the Unibus.
12	Unibus Multiple Parity Error	Set if an error occurs which causes the parity error signal to be asserted on the Unibus with the Error Address Register locked by a previous error.
11	CPU Error	Set if any memory error occurs during a Cache cycle from the processor.
10	Unibus Error	Set if any memory error occurs during a Cache cycle from the Unibus.
9	CPU Unibus Abort	Set if the processor traps to vector 114 because of a Unibus parity error on a DATI or DATIP cycle by the processor on the Unibus.
8	Error in Maintenance	Set if an error occurs when any bit in the Maintenance Register is set. The Maintenance Register will then be cleared.
7-6	Data Memory	These bits are set if a parity error is detected in the Fast Data Memory in the Cache. Bit 7 is set if there is an error in Group 1, bit 6 for Group 0.
5-4	Address Memory	These bits are set if a parity error is detected in the Address Memory in the Cache. Bit 5 is set if there is an error in Group 1, bit 4 for Group 0.
3-2	Main Memory	These bits are set if a parity error is detected on data from Main Memory. Bit 3 is set if there is an error in either byte of the odd word, bit 2 for the even word. An abort occurs if the error is in the word needed by a CPU reference. A trap occurs if the error is in the other word, or if it is a Unibus reference.
1	Main Address Parity Error	Set if there is a parity error detected on the address and control lines on the Main Memory Bus.
0	Main Memory Time-out	Set if there is no response from Main Memory. For CPU cycles, this error causes an abort. When a Unibus device requests a non-existent location, this bit will not set.

2.7.7.4 Control Register 17 777 746 - This bit register, illustrated in Figure 2-33 controls fourteen important internal functions; these are outlined in Table 2-10 The Control Register allows running the PDP-11/70 in a degraded mode; this may be desirable if parts of the Cache are malfunctioning, it is possible to force all operations through Group 1. Setting bit 4 or bit 5 allows the internally generated Random bit to be overridden and causes data, fetched from Main Memory as a result of a read miss to be replaced in the specified group. If bits 5 and 2 of the Control Register are set and bits 4 and 3 are cleared, the CPU will not be able to read data from Group 0, and all Main Memory data replacements will occur within Group 1. In this manner, half of the Cache will be operating. Bus system throughout will not decrease by 50 percent, since the statistics of read hit probability will still provide reasonably fast operation. If Group 1 is malfunctioning, bits 4 and 3 should be set and bits 5 and 2 cleared so that only Group 0 is operating. If all of the Cache is malfunctioning, bits 3 and 2 should be set. The Cache will be bypassed, and all references will be to Main Memory.

Control Register bits 5 and 4 can also be used to keep a desired routine in the Fast Data Memory. For example, if bit 5 is cleared and bit 4 is set prior to execution of a desired routine, the routine will be loaded into Group 0. If bit 5 is cleared and bit 4 is set when the desired routine is not being executed, the routine will remain protected in Group 0 for future reference. The routine can be protected in Group 0 while it is being executed if bit 5 is set and bit 4 is cleared.

CACHE REGISTERS

Figure 2-33 Cache Control Register 17 777 746



NOTES
* READ ONLY
✓ WRITE ONLY

2.7.7.4.1 Summary Of Cache Control Register Bit Functions - Table 2- 10 summarizes the functions of the bits of the Cache Control Register.

2.7.7.4.2 Disable Trapping With Bits 1 And 0 - Bits 1 and 0 can be set to disable trapping. With these bits set, the processor will not spend time performing trap service routines each time a non-fatal error occurs. Overall system operation will produce correct results; however, more Main Memory Bus cycles may be performed.

2.7.7.4.3 Use Of The Control Register In Troubleshooting - The control register can also be used in troubleshooting. For example, by setting register bits 3 and 2, the Cache is effectively disabled. If the system operates with these bits set and does not operate if they are cleared, a malfunction in the Cache is indicated.

Bits (5:0) are read/write. The bits are cleared on power-up or by Console Start. They are unaffected by a RESET instruction.

TABLE 0

SUMMARY OF CACHE CONTROL REGISTER BIT FUNCTIONS (Sheet 1 of 3)

BIT	NAME	BIT FUNCTION
15	Valid Store Parity Error	Inicates parity error on the valid store in use has occurred. Cache begins Slow cycle (bypassing cache and all references are forced to main memory) and trap to 114 occurs. Cleared by writing 1 into it by means of software, or on power up or console start.
14	In hibit Valid Store Switching	Inhibits switching to other valid bits (Valid Store) when Cache Flush command is issued. Cache will be turned off until cache hardware has completed clearing valid store in use, and then cache will resume normal operation. Set by writing 1 into it by means of software. Cleared by writing 0 into it by means of software or on power up or by console start.
13	Valid Store In Use	Indicates which valid store is in use. 0 is valid set A. 1 is Valid Set B.
12	Valid Clear In Progress	Set while cache hardware is clearing off-line valid store set. Read only and is cleared by power-up, flush cache, or a change in state of the valid store in use, or on-line because of software. NOTE : Hardware clear cycle of a Valid Store set will take about 52 microseconds
11	Disable Main Memory Aborts	Disables Main Memory Aborts if Enable Disabling Of Main Memory Aborts (bit 0, EDMA of Maintenance Register) is also set. Aborts due to Main Memory Parity Errors will be disabled when both these bits are set. Can be set only if MR 00, EDMA is set. If MR , 00, EDMA, is clear, this bit will be held in clear condition , over-riding all software attempts to set it. NOTE: DMMA, bit 11, and EDMA, bit 0, are for maintenance use only and are intended to help in the isolation of main memory parity errors. They should never be set for normal operation as they can permit the cpu to receive bad data from mainmemory without CPU being notified of errors.

Table 2-10 Summary of Cache Control Register Bit Functions (Sheet 2of 3)

BIT	NAME	BIT FUNCTION
10	FORCE VALID PARITY ERROR	Forces parity check logic for valid store in use to generate a bad parity indication. Results in trap to 114 unless Disable Traps bit is set. Two halves of valid store can be tested by using Force Miss and Force Replacement bits to disable one-half of cache at a time. This bit is read and write and is cleared by power-up, console start, software, or any memory system error.
9	Unconditional Cache Bypass	Forces all references to main memory while invalidating cache on read or write hits. Cache is only effected on write or read hits.
8	CACHE FLUSH	Cause total contents of cache to be invalidated by replacing current on-line valid bits with cleared off-line set of valid bits. On-line set that has been switched off-line is then cleared.
6 and 7		Not used
4 and 5	Force Replacement	Setting these bits forces data replacement within a group in the Cache by Main Memory data on a read miss. Bit 5 selects Group 1 for replacement; bit 4 selects Group 0.
3 and 2	Force Miss	Setting these bits forces misses on reads to the Cache. Bit 3 forces misses on Group 1; bit 2 forces misses on Group 0. Setting both bits forces all cycles to Main Memory.

Table 2-10 Summary of Cache Control Register Bit Functions (Sheet 3 of 3)

BIT	NAME	BIT FUNCTION
1	Disable Unibus Trap	Set to disable traps to vector 114 when parity error signal is placed on Unibus.
0	Disable Traps	Set to disable traps from soft errors (non-fatal errors that do not cause abort).

2.7.7.5 Maintenance Register 17 777 750 - This register, illustrated in Figure 2-34 is used for memory system maintenance. Table 2-11 lists the functions of the register bits.

The Maintenance Register is read/write. It is cleared on power-up or by Console Start. It is also cleared whenever any memory system error is detected.

This register is for maintenance use only.

Figure 2-34 Maintenance Register 17 777 750

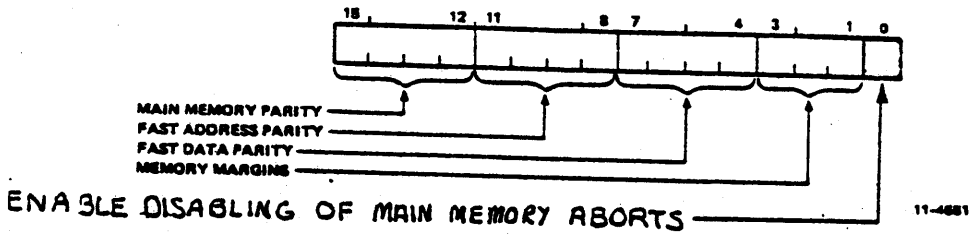


Table 2-11 Maintenance Register

Bit	Name	Function
15–12	Main Memory Parity	Setting these bits causes the four Main Memory parity bits to be checked as 1s. There is one bit per byte; there are four bytes in the data block. Bit Set — Byte 15 — Odd word, high byte 14 — Odd word, low byte 13 — Even word, high byte 12 — Even word, low byte
11 8	Fast Address Parity	Setting these bits causes the four parity bits for fast address memory to be wrong. Bits 11 and 10 affect Group 1; bits 9 and 8 affect Group 0.
7 4	Fast Data Parity	Setting these bits causes the four parity bits to be checked as 0s. Bit Set — Byte 7 — Group 1, high byte 6 — Group 1, low byte 5 — Group 0, high byte 4 — Group 0, low byte
3 1	Memory Margins	These bits are encoded to do maintenance checks on Main Memory. Bit 3 / Bit 2 / Bit 1 0 0 0 — Normal operation 0 0 1 — Check wrong address parity 0 1 0 — **Early MDR Load** 0 1 1 — **Late Refresh** 1 0 0 — **Reserved** 1 0 1 — **Reserved**
00	**Enable Disabling of Main Memory Aborts (EDMA)**	**When set, enables setting of Disable Main Memory Aborts bit (CCR 11-DMMA). When both bits are set, main memory parity error will not cause an abort ,instead, main memory aborts will result in traps to 114. Used to isolate failures in main memory system.**

NOTE:

Bits CCR 11 (DMMA), and MR 00 (EDMA) are for maintenance used only and should never be used during normal operation, since they can permit the CPU to receive bad data from main memory without aborting the operation or notifying the CPU.

2.7.7.6 Hit/Miss Register 17 777 752 - The Hit/Miss Register, illustrated in Figure 2-35, indicates whether the six most recent references by the CPU were hits or misses. A one indicates a read hit; a zero indicates a read miss or a write. The lower numbered bits are for the more recent cycles.

All the bits are read only. The bits are undetermined after a power-up. They are not affected by a RESET instruction.

This register is for maintenance use only.

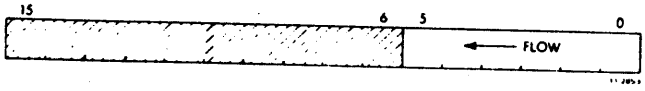


Figure 2-35 Hit/Miss Register

2.7.7.7 Use Of Cache Registers - When a memory system error is detected, the processor traps to location 114, If location 114 is used as a trap catcher, the operator can examine the Memory System Error Register to determine the type of error which has occurred. The Low Error Address and High Error Address Registers can then be examined to determine where in the program, and during what type of cycle, the error occurred. If statistics on the hit ratio are desired, the Hit/Miss Register can be read. The Control Register can be read to determine what the control conditions were at the time the error occurred.

If location 114 is not used as a trap catcher, the above tasks must be performed by the trap service routine.

If bit 14 (CPU Abort After Error) or bit 12 (Unibus Multiple Parity Error) of the Memory System Error Register is set, the address stored in the Low Error Address and High Error Address Registers is the address of the first error and not the address at which the most recent error occurred. The address at which the most recent error occurred must be reconstructed from the contents of the SP (which points to the virtual address incremented by 2) and the appropriate Memory Management PAR.

The contents of the Memory System Error Register and the High and Low Error Address Registers indicate the failing section of the memory system.

For example, if a Main Memory parity error bit is set in the Error Register, all the information required to determine the failing 16K section of memory is present. The Low and High Error Address Registers indicate the 32K section of memory in which the error occurred. The Error Register indicates whether the error occurred on the odd or even addressed word. If, for instance, the error occurred in the odd addressed word, the 16K section containing odd addressed words should be replaced.

If an FDM parity error bit is set in the Error Register, the bad chip is on the M8184 (DTM) module. KNowing which group failed and the state of address bit A01 (from the Low Error Address Register), it can be determined which of the four word sections of the FDM (Group 0 even and odd, Group 1 even and odd) has failed.

If an Address Memory parity bit is set in the Error Register, the problem is on the M8183 (ADM) module. The Error Register indicates whether the error occurred in the TAg 0 or Tag 1 Address Memory.

If the Main Memory Address Parity bit is set, there may be a problem in the parity generator (Drawing ADMJ) or in a memory controller parity checker. A failure in the Main Memory Bus address and control lines is the most likely cause for this error.

If the Main Memory time-out bit is set, the most probable cause is a memory controller failure. Another possible cause is a misconfiguration of the System Size Register in the processor.

2.7.8 Memory Management

Refer to Figure 2-, Memory Management receives the (VA) from the processor. It generates the (PA), which is received by the Cache or by the Unibus. As a result of its management functions. Memory Management informs the processor of traps and aborts.

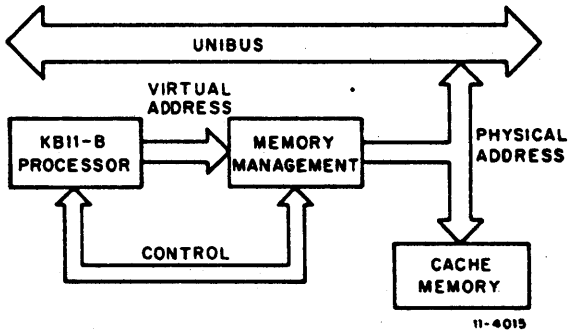


Figure 2- Block Diagram

2.7.8.1 Purpose - Memory Management receives all Virtual Addresses generated by the program, relocates them if necessary and then transmits the physical addresses to the Cache or to the Unibus. Address modification is the main function of Memory Management. This modification of addresses is called Relocation because it consists of adding a fixed constant to every virtual address. **Construction of the Physical Address is shown in Figure 2-37.**

Memory Management also allows the user to protect one section of memory from access by programs located in another section. It divides the memory into sections - called pages (collection of contiguous addresses). Each individual page has a protection or access key associated with it that defines access to the page. With the Memory Management unit, a page can be keyed non-resident (memory neither readable nor writable) or read-only (no write operations to memory). These two types of protection, in association with other features, enable the user to develop a secure computer operating system. With the non-resident key, memory not specifically assigned to a program can be made unavailable to it.

It is often desirable to load a program into one area of physical memory and then execute it as if it were located in another area of memory, e.g., when several user programs are simultaneously stored in memory. When any one program is running, it must be accessed by the processor as if it were located in the set of addresses beginning at 0. This process is called Relocation. When the processor accesses virtual address 0, a base address is added to the address; thus, the relocated 0 location of the program

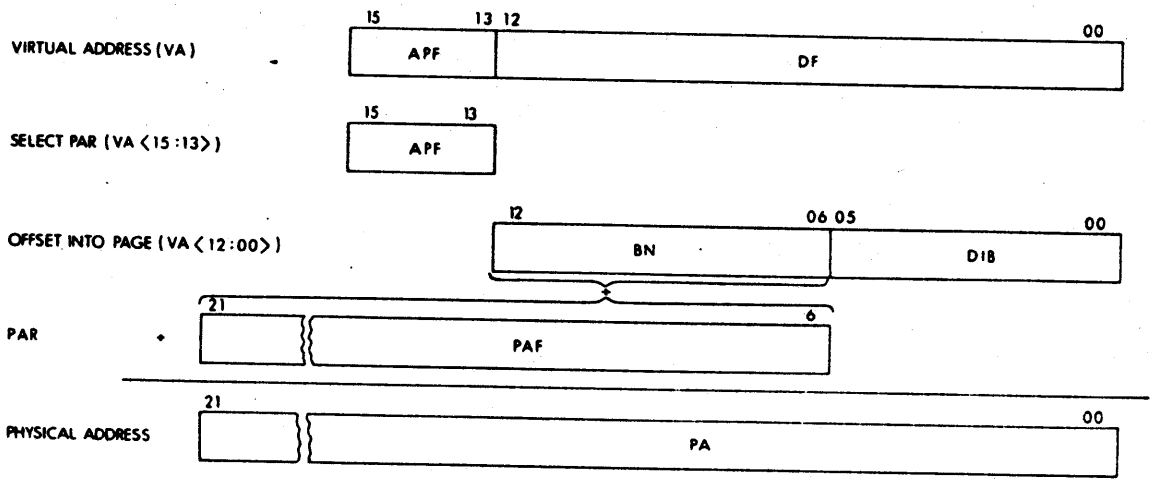


Figure 2-37 Construction of PA

is accessed. Typically, this same base address is added to all references while the program is running. A different base address is used for each of the other programs in memory.

Memory Management specifies relocation on a page basis, which allows a large program to be loaded into discontiguous pages in memory. This ability eliminates the need to shuffle programs to accommodate a new one. It also minimizes unusable memory fragments, allowing more users to be loaded in a specific memory size.

A program and its data may occupy as many as 16 pages in the memory. The size of each page may vary and can be any multiple of 32 words, up to 4096 words in length. This feature allows small areas in memory to be protected, i.e., stacks, buffers, etc., and also allows the last page of a program, exceeding 4K words, to be of adequate length to protect and relocate the remainder of the program.

As a result, the memory fragmentation problem inherent with fixed-length pages is eliminated. The base address of each page can be any multiple of 32 words in the Physical Address space, thus ensuring compacted core. Finally, the variable page size enables pages to be dynamically changed at run time.

The Memory Management unit provides two bits of active page status information: an "accessed" bit and a "written into" bit. These bits can be used by the operating system program to determine whether the page has been accessed and, if so, whether it was written into. The accessed bit can be used by operating system

programs to determine which page should be overlaid with the new program page in systems that swap programs back and forth from a disk. The written into bit can be used to determine whether the page to be overlaid must be swapped back to the disk or whether it is identical to a copy already there.

The Page Address Register (PAR) specifies the starting address of the memory page which that PAR describes.

Memory management divides the 32K Virtual Address space into eight 4K sections called Virtual pages. The lowest Virtual Address in each page is a whole multiple of 4096. The three high order bits of VA (VA (15:13) are the page numbers (0-7) and select a PAR/PDR pair within the current mode (Kernel, Super or User) and Instructions or Data space (I or D).

This PAR/PDR pair in turn defines the Physical Page. The PAR contains the base address of this page, which may be on any whole multiple of 32 words. A block consists of 32 words, and a physical memory page may consist of up to 128 blocks. The Page Length Field (PLF) of the PDR (bits 14:08) determines the allowable length of the page. A page may expand upward (from lower to higher addresses) or downward. Expansion direction is determined by bit 3 of the PDR (ED).

In addition to its relocation function, Memory Management has supervisory or memory protection functions.

The Page Description Register (PDR) is read at the same time as its corresponding PAR during relocation and contains all the information required for the supervisory functions.

2.7.8.2 Memory Management Registers - A brief description of the Memory Management registers follows. For a detailed description of their operation, refer to Section IV of the KB11-C Processor Manual.

Memory Management Register 0 (MMR0)

MMR0 contains error flags, the page number whose reference caused the abort, and various other status flags. The register is organized as follows:

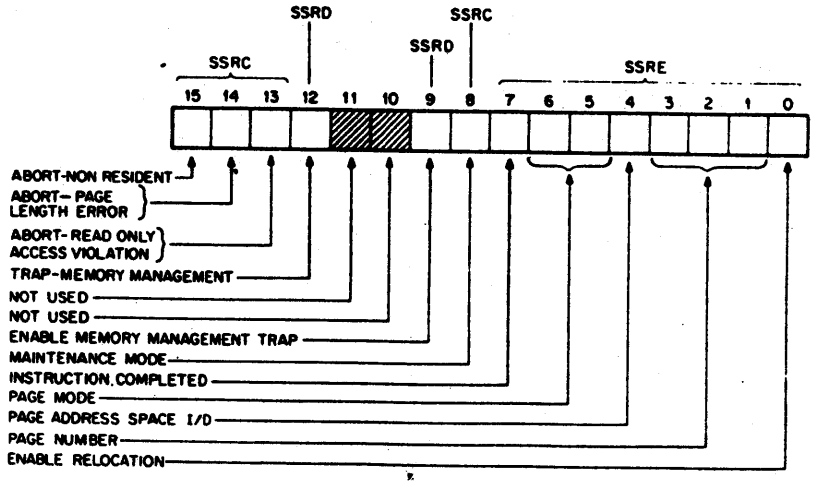


Figure 2-38 Memory Management Register 0

Setting bit 0 of this register enables address relocation and error detection. This means that the bits in MMR0 become meaningful.

Bits 15-12 are the error flags. They may be considered to be in a "priority queue" in that "flags to the right" are less significant

and should be ignored. That is, a "nonresident" fault-service routine would ignore length, access control, and memory management flags. A "page length" service routine would ignore access control and memory management faults, etc.

Bit 15-13, when set (error conditions), cause memory management to freeze the contents of bits 1-7 and Memory Management registers 1 and 2. This has been done to facilitate error recovery.

These bits may also be written under program control. No abort will occur, but the contents of the Memory Management registers will be locked up as in an abort.

15: Abort-Nonresident

Bit 15 is the abort-nonresident bit. It is set by attempting to access a page with an Access Control Field (ACF) key equal to 0, 3, or 7. It is also set by attempting to use memory relocation with a processor mode of 2.

14: Abort-Page Length

Bit 14 is the abort-page length bit. It is set by attempting to access a location in a page with a block number (virtual address bits, 12-6) that is outside the area authorized by the Page Length Field (PLF) of the Page Descriptor Register (PDR) for that page. Bits 14 and 15 may be set simultaneously by the same access attempt. Bit 14 is also set by attempting to use memory relocation with a processor mode of 2.

13: Abort-Read Only

Bit 13 is the abort-read only bit. It is set by attempting to write on a Read-only page. Read only pages have access keys of 1 or 2.

12: Trap-Memory Management

Bit 12 is the trap-memory management bit. It is set whenever a memory management trap condition occurs; that is, a read operation which references a page with an Access Control Field (ACF) of 1 or 4, or a write operation to a page with an ACF key or 4 or 5.

11 and 10 - Spares

Bits 11 and 10 are spare and are always read as 0; they should never be written. They are unused and reserved for possible future expansion.

9: Enable Memory Management Traps

Bit 9 is the enable memory management traps bit. It is set or cleared by doing a direct write into MMR0. If bit 9 is 0, no memory management traps will occur. The A and W bits will, however, continue to log memory management trap conditions. When bit 9 is set ot 1, the next memory management trap condition will casue a trap, vectored through Kernel virtual address 250.

Note that if an instruction which sets bit 9 to 0 (disable memory management trap) causes a memory management trap condition in any of its memory references prior to and including the one actually

changing MMR0, then the trap will occur at the end of the instruction anyway.

8: Maintenance/Destination Mode

Bit 8 specifies that only destination mode references will be relocated using memory management. This bit may be used only for maintenance purposes.

7: Instruction Completed

Bit 7 indicates that the current instruction has been completed. It will be set to 1 during T bit, parity, odd address, and time out traps and interrupts. This provides error handling routines with an aid to determine whether the last instruction will have to be repeated in the course of an error recovery attempt. Bit 7 is read only (it cannot be written). It is initialized to a 1. Note that EMT, TRAP, BPT, and IOR do not set bit 7. Refer to Section IV, of the KB11- Processor Manual.

5 and 6: Processor Mode

Bits 5 and 6 indicate the CPU Mode (Kernel/Supervisor/User) associated with the page causing the abort (Kernel=00, Supervisor=01, User=11, Illegal mode=10). If an Illegal mode is specified, bits 15 and 14 will be set.

4: Page Address Space

Bit 4 indicates the type of address space (I or D) the Unit was in when a fault occurred (0=I Space, 1=D Space). It is used in conjunction with bits 3-1, Page Number.

3 to 1: Page Number

Bits 3-1 contain the page number of a reference causing a memory management fault. Note that pages, like blocks, are numbered from 0 upward.

0: Enable Relocation

Bit 0 is the enable relocation bit. When it is set ot 1, all addresses are relocated by the unit. When bit 0 is set to 0, the memory management unit is inoperative and addresses are not relocated or protected.

Memory Management Register 1 (MMR1) 17 777 574

MMR1 records any autoincrement/decrement of the general purpose registers, including explicit references through the PC. MMR1 is cleared at the beginning of each instruction fetch. Whenever a general purpose register is either autoincremented or autodecremented, the register number and the amount (in 2s complement notation) by which the register was modified, is written into MMR1.

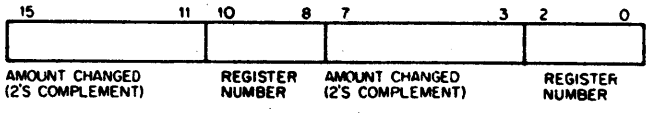


Figure 2-39 Memory Management Register 1

Memory Management Register 2 (MMR2) 17 777 576

MMR2 is loaded with the 16-bit virtual address (VA) at the beginning of each instruction fetch, or with the address trap vector at the beginning of an interrupt, T bit trap, parity, odd address, and timeout aborts and parity trap. Note that MMR2 does not get the trap vector on EMT, TRAP, BPT and IOT instructions. MMR2 is read only; it cannot be written, MMR2 is the Virtual Address Program Counter.

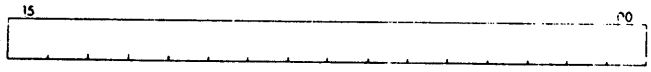


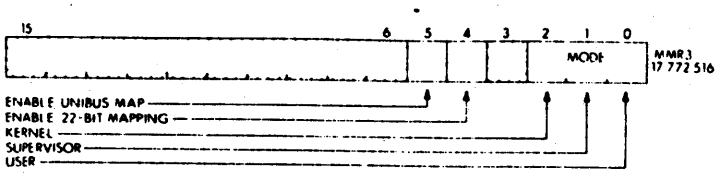
Figure 2-40 Memory Management Register 2

Memory Management Register 3 (MMR3) 17 772 516

Memory Management Register 3 (MMR3) enables or disables the use of the D space PARs and PDRs and 22-bit mapping and Unibus mapping. When D space is disabled, 11 references use the I space registers; when D space is enabled, both the I space and D space registers are used. Bit 0 refers to the User's registers, bit 1 to the Supervisors, and bit 2 to the Kernels. When the appropriate bits are set, D space is enabled; when clear, it is disabled. Bit 03 is read as zero and never written. It is reserved for future use. Bit 04 enables 22-bit mapping. If memory management is not enabled, bit 04 is ignored and 16-bit mapping is used.

If bit 4 is clear and memory management is enabled (bit 0 of MMR0 is set), the computer used 18-bit mapping. If bit 4 is set and

memory management is enabled, the computer uses 22-bit mapping. Bit 5 is set to enable relocation in the Unibus map; the bit is cleared to disable relocation. Bits 6 to 15 are unused. On initialization, this register is set to 0 and only I space is in use.



Bit	State	Operation
5	0	Unibus map relocation disabled
	1	Unibus map relocation enabled
4	0	Enable 18-bit mapping } if bit 0 of MMR0
	1	Enable 22-bit mapping } is set
2	1	Enable Kernel D Space
1	1	Enable Supervisor D Space
0	1	Enable User D Space

Figure 2-41 Memory Management Register 3

Page Address Registers (PAR)

The Page Address Register (PAR) contains the Page Address Field (PAF), 16-bit field, which specifies the starting address of the page as a block number in physical memory.

The addresses of these registers are listed in Table 2-12.

The Page Address Register may be alternatively thought of as a Relocation register containing a relocation constant, or as a

Table 2-12 PAR/PDR Unibus Addresses

Kernel

I Space			D Space		
No.	PAR	PDR	No.	PAR	PDR
0	17 772 340	17 772 300	0	17 772 360	17 772 320
1	17 772 342	17 772 302	1	17 772 362	17 772 322
2	17 772 344	17 772 304	2	17 772 364	17 772 324
3	17 772 346	17 772 306	3	17 772 366	17 772 326
4	17 772 350	17 772 310	4	17 772 370	17 772 330
5	17 772 352	17 772 312	5	17 772 372	17 772 332
6	17 772 354	17 772 314	6	17 772 374	17 772 334
7	17 772 356	17 772 316	7	17 772 376	17 772 336

Supervisor

I Space			D Space		
No.	PAR	PDR	No.	PAR	PDR
0	17 772 240	17 772 200	0	17 772 260	17 772 220
1	17 772 242	17 772 202	1	17 772 262	17 772 222
2	17 772 244	17 772 204	2	17 772 264	17 772 224
3	17 772 246	17 772 206	3	17 772 266	17 772 226
4	17 772 250	17 772 210	4	17 772 270	17 772 230
5	17 772 252	17 772 212	5	17 772 272	17 772 232
6	17 772 254	17 772 214	6	17 772 274	17 772 234
7	17 772 256	17 772 216	7	17 772 276	17 772 236

User

I Space			D Space		
No.	PAR	PDR	No.	PAR	PDR
0	17 777 640	17 777 600	0	17 777 660	17 777 620
1	17 777 642	17 777 602	1	17 777 662	17 777 622
2	17 777 644	17 777 604	2	17 777 664	17 777 624
3	17 777 646	17 777 606	3	17 777 666	17 777 626
4	17 777 650	17 777 610	4	17 777 670	17 777 630
5	17 777 652	17 777 612	5	17 777 672	17 777 632
6	17 777 654	17 777 614	6	17 777 674	17 777 634
7	17 777 656	17 777 616	7	17 777 676	17 777 636

Base register containing a base address. Either interpretation indicates the basic importance of the Page Address Register (PAR) as a relocation Tool.

There are six sets of eight PARs, one set for Kernel Data Space, one for Kernel Instruction Space, one for Supervisor Data Space, one for Supervisor Instruction Space, one for User Data Space, and one for User Instruction Space.

Page Descriptor Registers (PDR)

The Page Descriptor Register (PDR) contains information relative to page expansion, page length, access control, and Cache Bypass.

There are six sets of eight PDRs which are allocated in the same manner as the PARs. The addresses of these registers are listed in Table 2-12.

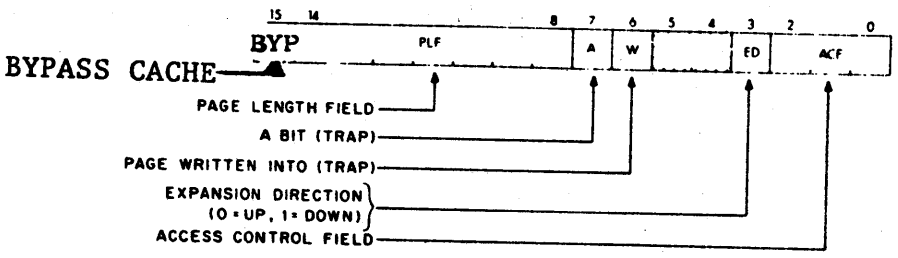


Figure 2-42 Page Descriptor Register

2 to 0: Access Control Field (ACF)

This three-bit field, occupying bits 2-0 of the Page Descriptor Register (PDR) contains the access rights to this particular page. The access codes or "keys" specify the manner in which a page may be accessed and whether or not a given access should result in a trap or an abort of the current operation. A memory reference which causes an abort is not completed while a reference causing a trap is completed. In fact, when a memory reference causes a trap to occur, the trap does not occur until the entire instruction has been completed. Aborts are used to catch a "missingpage fault," prevent illegal access, etc,; traps are used as an aid to gather memory management information.

In the context of access control, the term "write" is used to indicate the action of any instruction that modifies the contents of any addressable word.

The modes of access control are as follows:

000 Nonresident Abort all accesses

001 Read only Abort on write attempt, memory management trap on read

010 Read only Abort on write attempt

011 Unused Abort all accesses - reserved for future use

100 Read/write Memory management trap upon completion of a read or write

101 Read/write Memory management trap upon completion of a write

110 Read/write No system trap/abort action

111 Unused Abort all accesses - reserved for future use

It should be noted that the use of I space provides the user with a further form of protection, execute only.

7: A Bit

This bit is used by software todetermine whether any accesses to this page met the trap condition specified by the Access Control Field (AFC) (A = 1 is affirmative). The A Bit is used in the process of gathering memory management statistics.

6: W Bit

This bit indicates whether this page has been modified (i.e., written into) since either the PAR or PDR was loaded. (W = 1 is affirmative). The W Bit is useful in applications that involve disk swapping and memory overlays. It is used to determine which pages have been modified and hence must be saved in their new form, and which pages have not been modified and can be simply overlaid.

Note that A and W bits are "reset" to "0" whenever either PAR or PDR is modified (Written into).

3: Expansion Direction (ED)

Bit 03 of the Page Description Register (PDR) specifies in which direction of the page expands. If ED = 0 the page expands upwards from Block Number 0 to include blocks with higher addresses; if ED = 1, the page expands downwards from Block Number 127 to include blocks with lower addresses. Upward expansion is usually used for program space while downward expansion is used for stack space.

14 to 08: Page Length Field (PLF)

This 7-bit field specifies the block number, which defines the boundary of that page.

The block number of the virtual address is compared against the page length field to detect length errors. An error occurs when expanding upwards if the block number is greater that the page length field.

15: Bypass Cache (BYP)

When the BYP bit 15 is set in the Page Descriptor Register (PDR) and relocation is enabled, any CPU references to the virtual page mapped by that PAR/PDR pair will bypass the Cache and go directly to main memory. If read or write hits occur the contents of that location in cache will be invalidated. Read or write misses will not disturb the contents of the cache. When relocation is disabled by the installation of a jumper at W3 on the Cache Control

Board, bit 15 will have no effect on cache operation.

5, and 4: Reserved Bits

Bits 5, and 4 are spare and are always read as 0, and should never be written. They are unused and reserved for possible future expansion.

2.7.9 Unibus Map

2.7.9.1 Purpose - The Unibus Map is the interface between the Unibus and Cache. It responds as a slave device to Unibus signals and converts 18-bit Unibus addresses to 22-bit Cache addresses.

The top 4K word addresses of the 128K Unibus addresses are reserved for CPU and I/O registers and are called the Peripherals Page (see Figure 2-43.) The lower 124K addresses are used by the Unibus Map to reference physical memory.

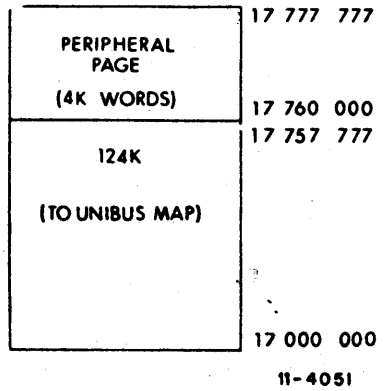


Figure 2-43 Unibus Address Space

The Unibus Map is the interface to memory from the Unibus. The operation is transparent to the user, if it is disabled.

2.7.9.2 Relocation Disabled - If the Unibus Map relocation is not enabled, an incoming 18-bit Unibus address has 4 leading zeros added for referencing a 22-bit Physical Address (PA). The lower 18 bits are the same. No relocation is performed.

2.7.9.3 Relocation Enabled - There are a total of 31 mapping registers for address relocation. Each register is composed of a double 16-bit PDP-11 word (in consecutive locations) that holds the 22-bit base address. These registers have Unibus addresses in the range 17 770 200-17 770 372.

If Unibus Map relocation is enabled, the 5 high order bits of the Unibus address are used to select one of 31 mapping registers. The low order 13 bits of the incoming address are used as an offset from the base address contained in the 22-bit mapping register. To form the PA, the 13 low order bits of The Unibus address are added to 22 bits of the selected mapping register to produce the 22-bit PA. The lowest order bit of all mapping registers is always a zero, since relocation is always on word boundaries.

The Unibus address is decoded by the Unibus Map. If it is a Cache register address (17 777 740 - 17 777 752), MAPB CACHE REG L is sent to the Cache; this signal, in addition to Unibus address bits

MAPA ADRS (03:01) H allows the Cache to select the register required for the current data transaction.

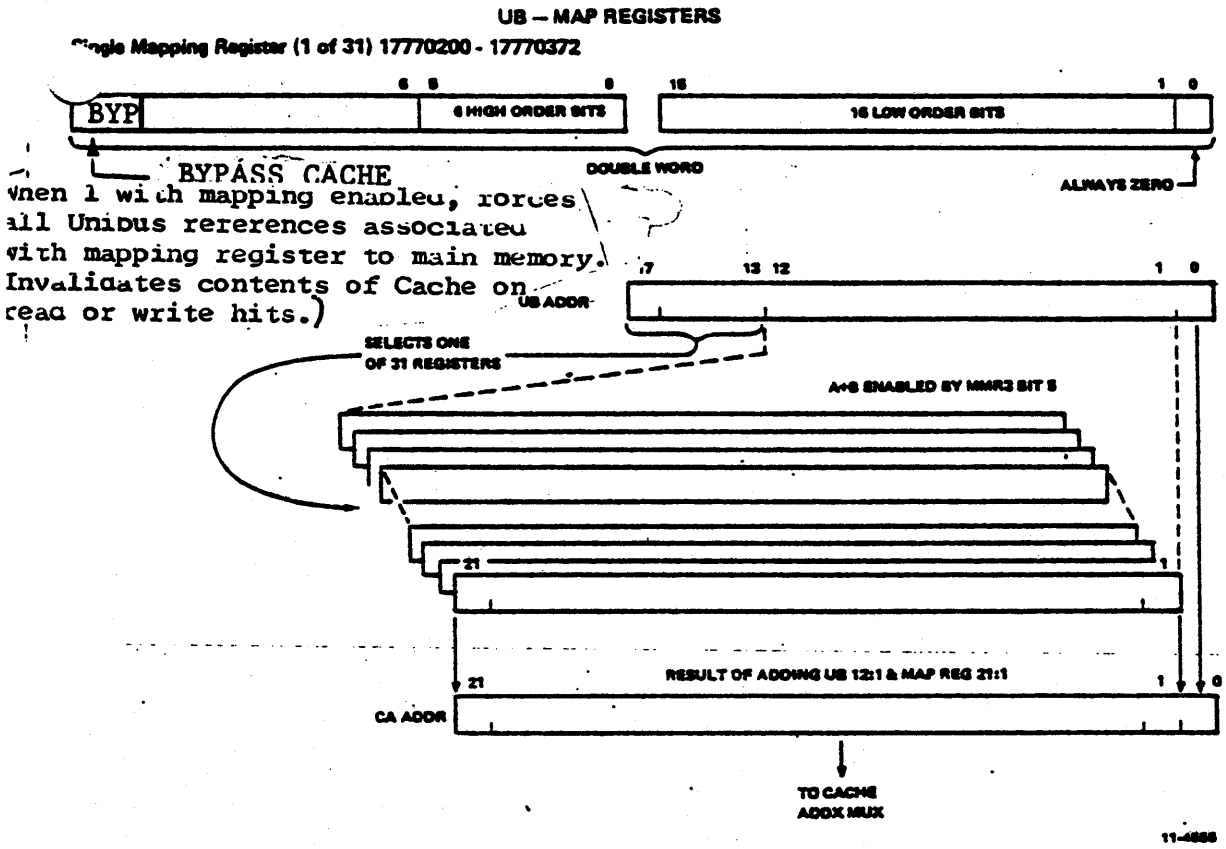
If the address is a valid Cache address (as determined by the limit jumpers), it is either sent to the Cache unmodified (if the Map is not enabled). The Map is enabled if bit 5 of Memory Management Register 3 is set (Unibus address 17 772 516).

The Map responds as a slave to the two major types of Unibus transactions: DATI (or DATIP) which requires a read from memory, and DATO (or DATOB), which requires a write into memory. The Map does not distinguish between DATI (data-in) and DATIP (data-in, pause), not between DATO (data-out) and DATOB (data-out byte): it transmits Unibus control bit C0, which distinguishes DATI from DATIP and DATO from DATOB, to the Cache.

2.7.9.4 Unibus Map Registers 17 770 200-17 770 336 - There are a total of 31 mapping registers for address relocation. Each register is composed of a double 16-bit PDP-11 word (in consecutive locations) that holds the 22-bit base address. The Unibus map address conversion is shown in Figure 2-44.

If Unibus Map relocation is enabled, the five high order bits of the Unibus address are used to select one of the 31 mapping registers. The low order 13 bits of the incoming address are

Figure 2-44 Unibus Map Address Conversion



MAPPING

16 BIT MAPPING

- FIXED MAPPING FROM VIRTUAL TO PHYSICAL ADDRESSES. (VIRTUAL ADDRESS IS IDENTICAL TO PHYSICAL ADDRESS.)
- ASSOCIATED WITH 28K OF MEMORY.
- TOP 4K ADDRESSES CAUSE UNIBUS CYCLES TO ADDRESSES 17 760 000-17 777 777.
- ENABLED AFTER POWER UP, CONSOLE START, OR RESET INSTRUCTION.

18 BIT MAPPING

- 32K VIRTUAL ADDRESSES FOR EACH OF THREE MODES; KERNEL, SUPERVISOR, USER)
- 32K ADDRESSES ARE MAPPED TO 128K OF PHYSICAL ADDRESSES
- TOP 4K ADDRESSES CAUSE UNIBUS CYCLES 17 760000--17777777

22 BIT MAPPING

- PERMITS ACCESS TO ALL 2 MEG (WORDS) OF MEMORY
- TOP 128K ADDRESSES CAUSE UNIBUS CYCLES 17000000 to 17777777

used as an offset from the base address contained in the 22-bit mapping register. To form the physical address, the 13 low order bits of the Unibus address are added to 22 bits of the selected mapping register to produce the 22-bit physical address. The lowest order bit of all mapping registers is always a zero, since relocation is always on word boundaries.

For a detailed description of Unibus Map operation, refer to section V, Chapters 1 and 3 of the KB11-C Processor Manual.

2.7.9.4.1 15 Bypass Cache (BYP) - When the BYP bit 15 is set in the Unibus Mapping Register and mapping is enabled any Unibus references using that mapping register will bypass the cache and go directly to main memory. If read or write hits occur the contents of that location in the cache will be invalidated. Read or write misses will not disturb the contents of the cache. When mapping is disabled, the Bypass Cache bit has no effect.

2.7.9.4.2 Register Access - Table 2-13 shows the correspondence between the Unibus addresses that select each mapping register and the two addresses used for reading or writing the same register.

Note that register 37 is selected by Unibus addresses (17) 760 000 - (17) 777 777. Since these addresses are higher than the maximum allowed by the upper limit jumpers, register 37 cannot be used as a mapping register. It can, however, be read and written into by using addresses (17) 770 374 and (17) 770 376.

Table 2-13 Access to Unibus Map Registers

Register No.	Unibus Address Read or Write		Unibus Address for Memory Reference When Mapping
	Low Order Bits	High Order Bits	
0	17 770 200	02	17 000 000 – 17 017 777
1	17 770 204	06	17 020 000 – 17 037 777
2	17 770 210	12	17 040 000 – 17 057 777
3	17 770 214	16	17 060 000 – 17 077 777
4	17 770 220	22	17 100 000 – 17 117 777
5	17 770 224	26	17 120 000 – 17 137 777
6	17 770 230	32	17 140 000 – 17 157 777
7	17 770 234	36	17 160 000 – 17 177 777
10	17 770 240	42	17 200 000 – 17 217 777
11	17 770 244	46	17 220 000 – 17 237 777
12	17 770 250	52	17 240 000 – 17 257 777
13	17 770 254	56	17 260 000 – 17 277 777
14	17 770 260	62	17 300 000 – 17 317 777
15	17 770 264	66	17 320 000 – 17 337 777
16	17 770 270	72	17 340 000 – 17 357 777
17	17 770 274	76	17 360 000 – 17 377 777
20	17 770 300	02	17 400 000 – 17 417 777
21	17 770 304	06	17 420 000 – 17 437 777
22	17 770 310	12	17 440 000 – 17 457 777
23	17 770 314	16	17 460 000 – 17 477 777
24	17 770 320	22	17 500 000 – 17 517 777
25	17 770 324	26	17 520 000 – 17 537 777
26	17 770 330	32	17 540 000 – 17 557 777
27	17 770 334	36	17 560 000 – 17 577 777
30	17 770 340	42	17 600 000 – 17 617 777
31	17 770 344	46	17 620 000 – 17 637 777
32	17 770 350	52	17 640 000 – 17 657 777
33	17 770 354	56	17 660 000 – 17 677 777
34	17 770 360	62	17 700 000 – 17 717 777
35	17 770 364	66	17 720 000 – 17 737 777
36	17 770 370	72	17 740 000 – 17 757 777
*37	17 770 374	76	17 760 000 – 17 777 777

*Note: Can be read or written into, but not used for mapping.

2.8 MKA11 MEMORY

The PDP-11/70 and PDP-11/74 Multiprocessors use the MKA11 MOS multi-ported memory as their main memory . the MKA11 provides common main memory which can be shared by two, three or four CPU's. Each processor's cache interfaces to one of the memory's four ports via a main memory bus. Arbitration circuitry, internal to the memory, allows each of the ports to access the shared main memory space on an equal priority basis. (Shown in Figure 2-45 is a block diagram of a dual-ported memory.)

Each MKA11 memory has a storage capacity of 64K to 512K 16-bit words. Larger main memory capacities are obtained by adding additional MKA11 memories to the system and daisy chaining the memory's ports on the main memory busses. Up to eight memories can be daisy-chained in this fashion, expanding the shared main memory space to the maximum addressable 16M words.

The MKA11 memory stores data in dynamic MOS random access storage arrays. Data is organized on the arrays as 39-bit double word blocks. double word consists of four eight bit data bytes and seven error correction code (ECC) check bits. Three types of storage arrays are available, two with capacities of 16K 39-bit double words (32K words)

MKA11 Memory Block Diagram

(DUAL-PORTED)

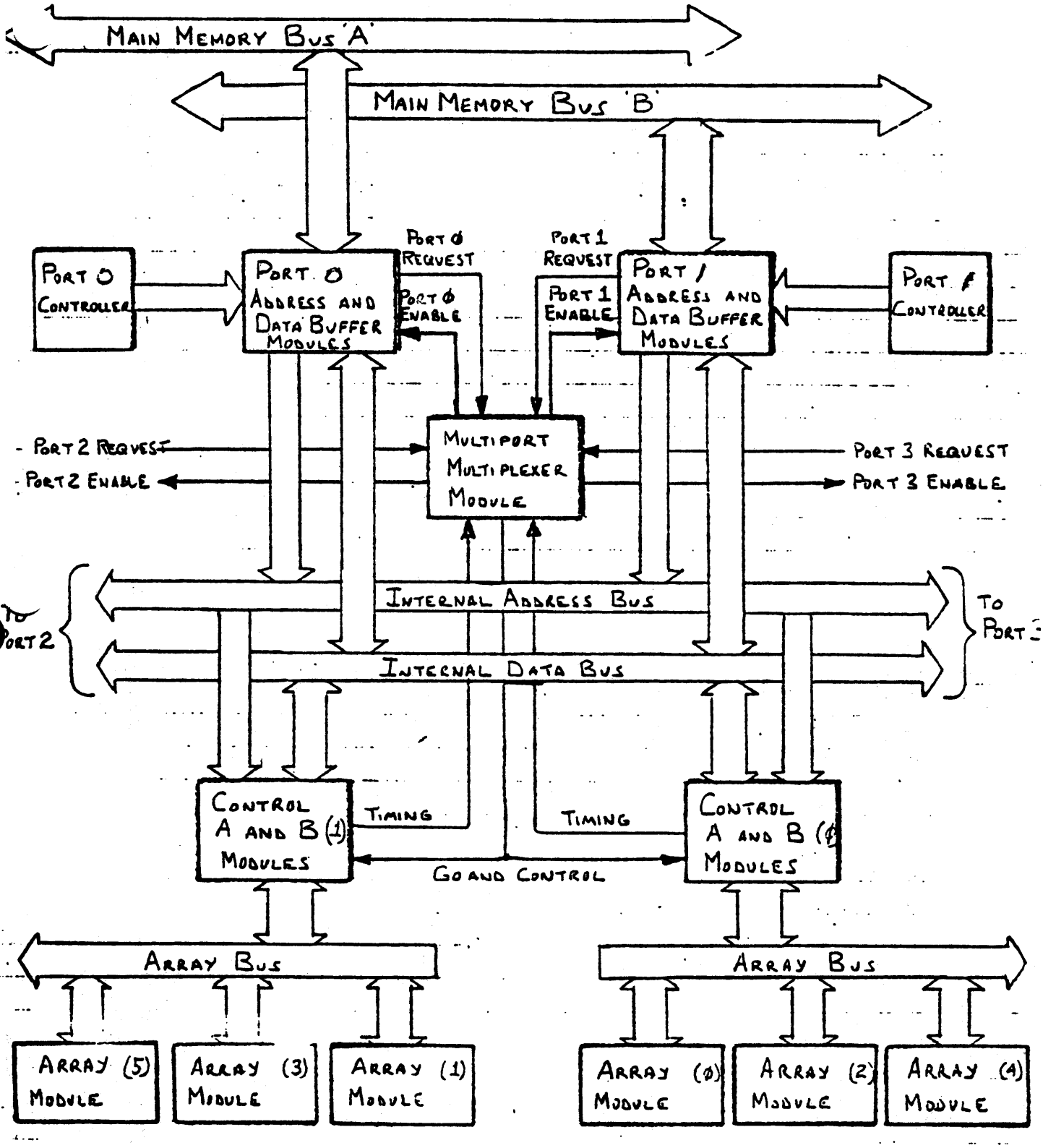


Figure 2-45

and one with a capacity of 64K 39-bit double words (128K words). The three types of arrays may be mixed within a memory frame. Up to 16 sto age arrays fit into the MKA11 memory frame.

The seven error correction code check bits stored with each double world uniquely define the pattern of data in the four data bytes. When data is read from an array, the memory uses the ECC check bits to detect any errors in the stored data. An error in any one bit of the 32 bits of data will be corrected before transferring the data to cache. If several bits are in error, a parity trap will be forced by the memory.

Control and status registers (CSRs) at each port store operating stat and error information. The CSRs log error information, indicating the type of error (single bit or double bit) and the storage array which contains the error. Starting address, memory capacity, interleaving information are also stored in each port's CSR. For diagnostic purpos the CSR's can be used to control memory functions in order to test the memory and to change the interleaving and address structure.

2.8.1 Addressing and Data Transfers

Main memory locations in the MKA11 are four bytes wide, each double word has a unique 23 bit address determined by the starting address and the interleaving scheme selected for the memory. Cache memory accesses main memory by palcing a 23 bit address (A24:A02) on its main memory bus. Figure 1-1 shows the address and data format on the main memory bus.

During a read operation, cache addresses a double word with A24:A02

and receives from memory a 35 bit double word (four bytes plus four parity bits) formatted as in Figure 1-1. When the cache writes data to main memory, it can access individual bytes. Byte addresses (0 through $1777\ 777\ 777_8$) are obtained by combining the byte mask bits with the double word address (A24:A02). Write operations can be directed to one, two, three, or four bytes of the addressed double word; address bits A24:A02 select the double word location and the byte mask bits 0,1,2, and 3 select data bytes 0,1,2, and 3 respectivel

The control bits, C1 and C0, define the type of memory cycle the memory will perform. The four cycles are:

C1	C0	
0	0	Read
0	1	Read-Pause-Write
1	0	Write
1	1	Exchange

NOTE:

The Read-Pause-Write (DATIP-DATO) is a read-modify-write memory cycle that is used in conjunction with the ASRB instruction when the ASRB is being used to implement a lock on a shared data location.

2.8.2 Major Components

The MKA11 memory is packaged as five major assemblies: the memory frame (or memory box), the port controllers (one for each port), the power supply, the remote AC power control box, and the battery back-up unit. These five assemblies are pictured in Figure 2-11A,

2.8.2.1 Memory Frame

The memory frame consists of an H9060 card cage with a 29 slot hex height back plane forming the back of the box and a squirrel cage cooling fan mounted on the bottom of the box. The back plane forms the memory's internal busses, seven mate-n-lok connnectors at the back (pin side) of the backplane provide the connection between the power supply and the memory frame. The memory's Address and Data Buffer modules, Control A and B modules, Multiplexer module, and MOS Array modules insert into the memory Box on guides and plug into the backplane edge connectors. Air is drawn downward past these modules and exhausted out the bottom by the fan.

The memory frame mounts in a 48 centimeter (19 inch) cabinet and takes up about 61 centimeters (24 inches) of cabinet space.

2.8.2.2 Port Controllers

The Port Controllers contain the memory's operator and maintenance control switches and indicator lights. There is one Port Controller for each port. The Port Controllers are approximately 10 by 13 centimeters (4 by 5 inches); up to four controllers mount in one control panel. The control panel is 48 centimeters (19

inches) wide for cabinet mounting. If the memory has fewer than four ports, the unused mounting locations of the control panel are covered with blank panels.

2.8.2.3 Power Supply and Power Control

The MKA11's power supply is an H7420. This power supply contains three 7014251 ± 12 volt, +5 volt regulator modules and two H7441 +5 volt regulator modules if the memory is a multiport, or one H7441 +5 volt regulator module if the memory is a single port. Four cooling fans force air across the transformer and regulator modules for cooling. The H7420 measures approximately 25 by 58 centimeters (10 by 23 by 8 inches) and mounts behind the memory frame at the rear of the cabinet.

(870)

The remote AC Power Control switches primary power to the power supply's transformer. It is housed in a small, 15 by 20 by 5 centimeter (6 by 8 by 2 inch), box which mounts on the side of the cabinet.

2.8.2.4 Battery Back-up Unit

The H7750 Battery Back-up Unit is housed in a 13 by 48 by 25 centimeter chassis (5-1/4 by 19 by 10 inches). The chassis contains three 12 volt batteries and battery charging circuitry. The batteries will supply power to retain the data in the volitile MOS memory chips during line transients or brown outs for a minimum of five minutes. The battery back-up unit mounts on a hinged door at the rear of the equipment cabinet.

2.8.2.5 Main Memory Busses

Each Main Memory Bus consists of four BCC6R cables. These ribbon cables contain 40 conductors each. Two of the bus cables carry the main memory data between the processor and the Port Data Buffer and two of the bus cables carry the address and control signals between the processor and the Port Address Buffer.

The Main Memory Bus between the processor and the memory Port closest to the processor extends from the back of the processor chassis and plugs into connectors at the front of the Port Data Buffer and Address Buffer modules. Other memory boxes are daisy-chained together by bus cables connecting each Port Data Buffer and Address Buffer modules to the same port of adjacent memory boxes.

Since the Main Memory Bus is characteristically a transmission line, it must be properly terminated at both ends. Four type H873 terminators are plugged into the bus cable connectors on the Port Data Buffer and Address Buffer modules of the last memory port on the bus. The terminators are secured to the port buffer modules with two small screws which also provide dc power to the terminators.

2.9 MASSBUS OPERATION

The Massbus consists of a Control Bus and a Data Bus. The Control Bus is an asynchronous bus that is used for register transfers to or from the various registers located in the Massbus Device. The Data Bus is a synchronous bus that is used for block data transfers to or from the Massbus device. The operation of these two busses is completely independent. The Massbus controller can be initiating a register transfer over the Control Bus at the same time that data is being transferred over the Data Bus.

Two electronic switches are in the RH01. One switch is implemented with the Control Bus and the other with the Data Bus. Each switch has three positions; neutral, switched to A and switched to B. The control element of each switch is made up of a pair of high speed flip-flops, each driven by a different phase of a master high speed clock.

Once the switches have switched to a given port, additional logic in the RH01 determines the direction of transfer for each of the Massbus signals. The combination of these factors (the switch

position and the transfer direction) enable the Massubs signals to be gated through the RH01.

2.9.1 RH01 Massbus Dual-Port Adapter

2.9.1.1 Purpose - The RH01 Massbus Dual Port Adapter, provides a means of dual porting single-ported Massbus peripherals. This permits sharing or switching Massbus peripherals between two computers for either resource sharing or for high availability redundant systems. The RH01 may be used with any Massbus device, such as the TU16, TU45, RS03, RS04, RP04, and DR70. As many as eight devices can be shared on the common Massbus of the RH01. The RH01 is also designed to allow local drives to be used on the same Massbus Controller. A local drive can be used by its processor withou interfering with simultaneous operation of one of the shared drives by the other processor. The RH01 can be used with devices having transfer rates as high as 1 MHz and will not reduce the data throughout of the Massbus. A maximum of 180 ns latency will be added to all read or write operations out of device control registers on the common Massbus. Shown in Figure 2-46 is a simplified system block diagram showing the RH01 Adapter connected to two different Massbus Controllers and to Massbus peripheral devices.

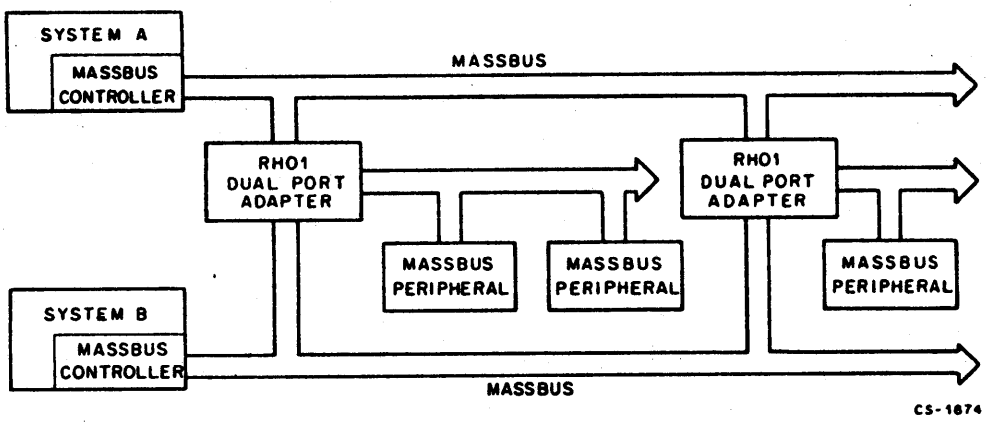


Figure 2-46 System Block Diagram

2.9.1.2 Operation - Switching of the RH01 can be controlled either manually or programmably. A four position switch on the front panel allows the controller to be locked on either the A or B port, to be put in the program control mode, or to be switched to the OFF position to allow maintenance on any device. When in the programmable mode, the RH01 can be essentially transparent to the user program. When an attempt is made by either port to read or write any of the device registers on the common Massubs, the RH01 will automatically switch from the unseized state to the requesting port. Once the controller has been seized by a port, if it is not used or released in the maximum amount of time (selectable at installation from 1 second to 4 minutes), the RH01 will automatically release from that port and become available to the other port. If an attempt is made to connect to one port while the Rh01 is connected to the other port, the request will be stored and the Rh01 will switch as it becomes available.

2.9.3 System Considerations

The RH01 will increase the length of time required for register transfers between the controller and a shared drive by approximatly 200 ns when there is no interference from the other port. When one port is attempting to do a register transfer while the RH01 is seized by the other port in Dual Access Mode, the seized port will be forced to wait momentarily while the RH01 responds to the other transfer. The amount of this additional delay should be fairly short because thr RH01 will abort the transfer on the unseized port.

The Massbus Controller has a nonexistent drive timeout error that is caused by the failure to receive a response from the drive within 1.5 μS. In systems with long delays due to the length of the Massbus cable, the 1.5 may not be long enough to compensate for the additional time added in the RH01. If nonexistent drive errors occur while running the drive through an Rh01 that is properly selected, the timeout should be lengthened in the controller by changing the timing components on the nonexistent drive one-shot.

The RH01 will not slow down the data transfers between the controller and the drive. However, in dual access systems, consideration must be given to the increased access time which results when the controller must wait for a shared drive while it is being used on the other port. This waiting time can be minimized by using individual RH01's on each drive rather than sharing a string of drives on a single RH01. When using only one RH01, the entire string of

drives is denied to the other port when any one drive is being used. Using an RH01 for each drive allows the drives to be individually selected by each port.

2.9.3.1 RH01 Electronic Switches - Two electronic switches are in the RH01. One switch is implemented with the Control Bus and the other with the Data Bus. Each switch has three positions; neutral, switched to A and switched to B. The control element of each switch is made up of a pair of high speed flip-flops, each driven by a different phase of a master high speed clock.

Once the switches have switched to a given port, additional logic in the RH01 determines the direction of transfer for each of the Massbus signals. The combination of these factors (the switch position and the transfer direction) enable the Massbus signals to be gated through the RH01.

4.5.8 Attention Summary Register (ASR) - When the RH01 has an attention condition (indicating a request was generated for the RH01 while it was seized by the other port) it simulates an interrupt from the device that was requested by asserting its Attention bit when the Attention Summary register is read. When the RH01 is requested, the decoded drive address is stored in an 8-bit register for the requesting port. If the Attention flip-flop for that port is set, the contents of the register will be ORed with the rest of the Attention Summary register through an open collector multiplexor. The multiplexor is connected to the internal bus between the drive and controller.

4.5.9 Attention Summary Control (ASC) - The major function of this logic is to clear the Attention Summary register and Attention flip-flop in the RH01. When the Attention Summary register is

being written, the bits being written are compared to the bits that are set in the Attention Summary register. If a bit is being written to a one, the Attention Summary register and Attention flip-flop for that port will be cleared. The logic also contains a circuit for clearing the attention condition when the RH01 is powered up.

NOTE:

If a request had been generated for the RH01 while it was seized by the other port, the Attention and Request bits for that port would have been set. As soon as the Data Bus switches to that port, the Attention signal will be enabled to the controller that generated the request.

2.10 DT07 Unibus Switch

2.10.1 Purpose

The DT07 Unibus Switch is an electronic device that allows a single Unibus peripheral or group of peripherals to be switched from one PDP-11 processor to another. It provides for the sharing of peripheral devices between processors and facilitates on-line system back-up and dynamic reconfiguration for systems where high availability is required.

The DT07 family contains two-, three-, and four-port versions, allowing connection of from two to four PDP-11 processors to a common (switched) Unibus. Figure 2-47 is a block diagram of a dual-processor system using a 2-port DT07. The DT07 family is characterized by the following features:

* Each switch port is a hex-size module which connects to a processor Unibus via a standard SPC slot, becoming one of that processor's complement of devices, allowing each port to be powered independently from the other ports.

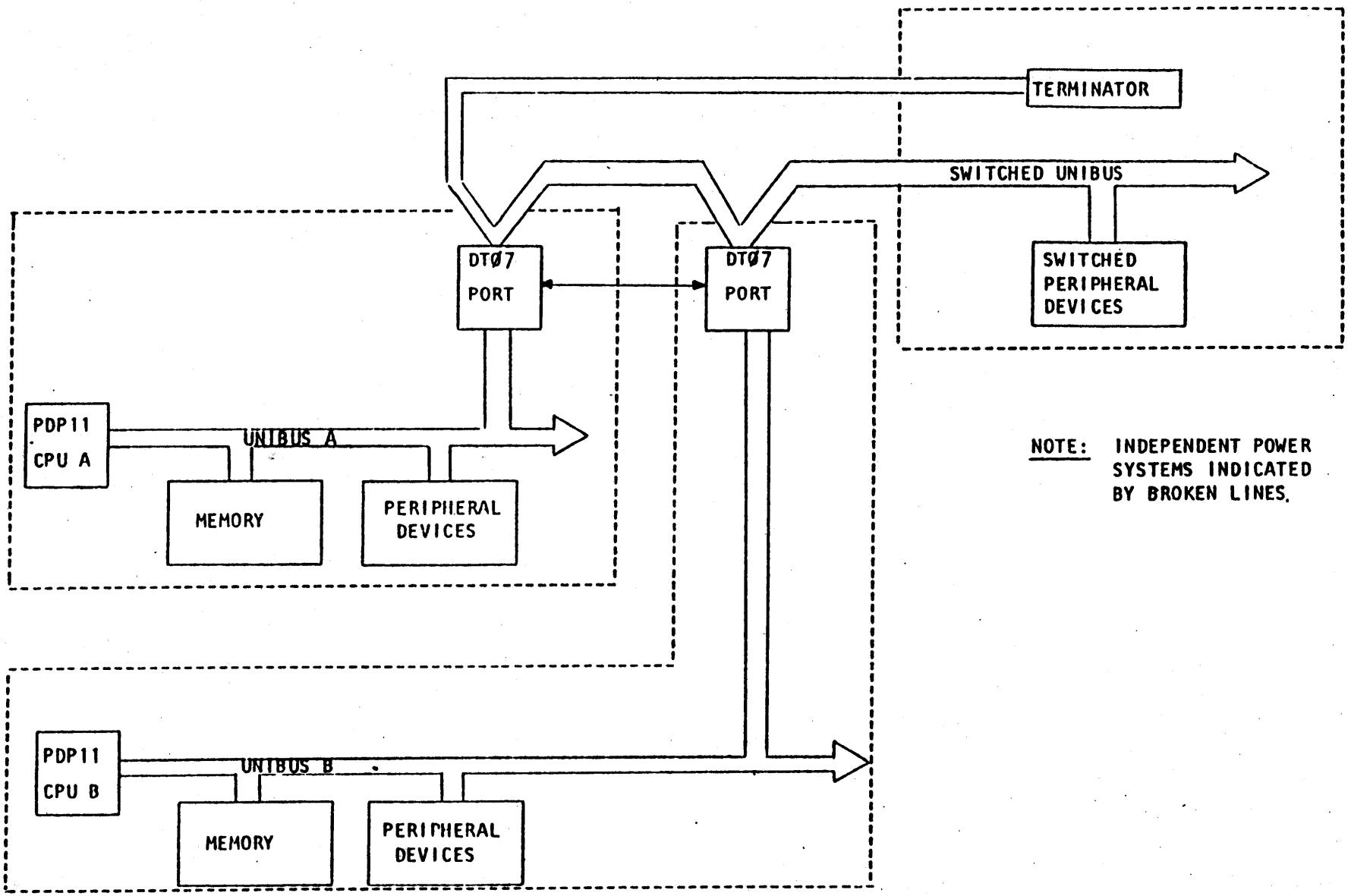
* Each switch port module functions as an isolation circuit. When its power is off, it will not affect the switched Unibus. If its power is on when power fails on the switched bus, the processor is notified via an interrupt but operations on the main bus are not affected.

* Switching is controlled programmably by direct computer command or by an optional manual control panel and is synchronized to assure that the switch changes position without interfering with any operations on the processor Unibus, thereby allowing a switch to occur while a program is running. Only one processor can be connected at any one time.

* Failsoft operation is supported in the programmable mode by providing an internal watchdog timer whose function is to cause the switched Unibus to be disconnected from a failing primary processor and then connected to a backup processor when a request from the backup is not actively rejected by the primary within a predetermined period of time.

* Connection of each port module to the switched Unibus is by means of Y-connected cables which can be removed from the module without breaking continuity of the switched bus, thus allowing for on-line repairability of the DT07 subsystem.

* Some versions allow a head-of-bus terminator assembly which is mounted with the switched peripherals and powered by the same supply powering those peripherals, thereby disassociating bus termination from all port modules. Other versions contain the terminator for the head of the switched Unibus on the port module itself, to be powered from the supply powering the switched peripherals.



NOTE: INDEPENDENT POWER SYSTEMS INDICATED BY BROKEN LINES.

FIGURE 2-47 DUAL PROCESSOR SYSTEM WITH 2-PORT SWITCH

2.10.2 OPERATION

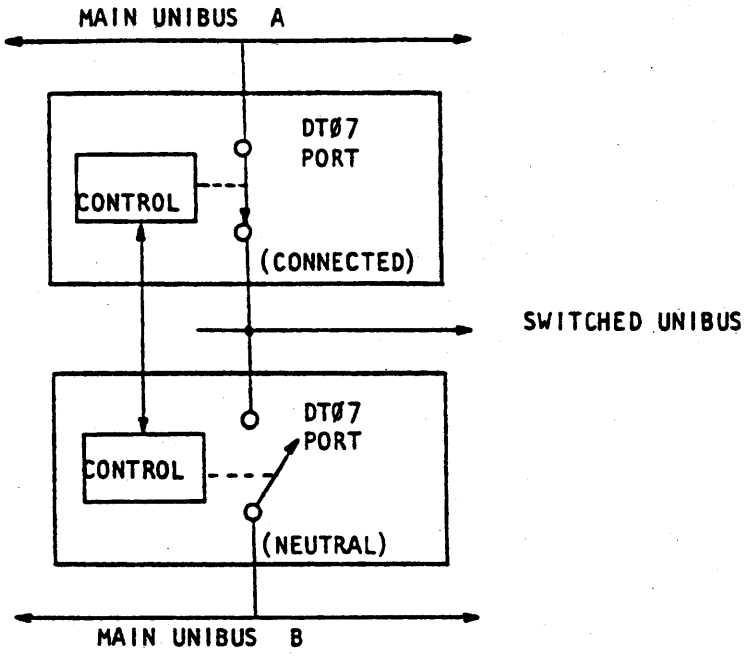
2.10.2.1 Basic Switch Functions

Each port in a DT07 subsystem can have only two switch positions: Connected and Neutral, defined as follows:

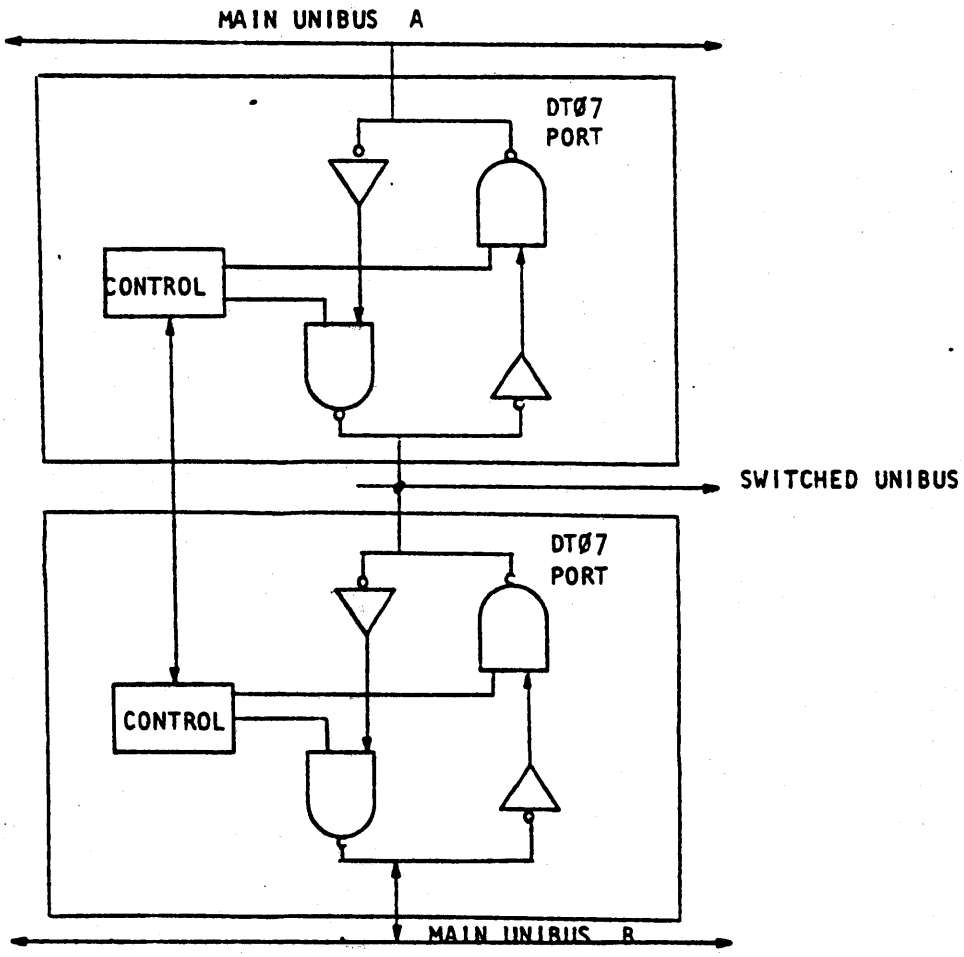
> Connected: In this position the switched bus is logically connected to the processor associated with that port, and all of the devices on the switched bus are available to that processor. Only one port of a switch subsystem can be in the Connected position at any one time; i.e., the common bus can only be used by one processor at a time.
>
> Neutral: In this position the switched bus is not logically connected to the processor, and operations on the common bus have no effect on the processor or its Unibus. When all ports are in the Neutral position devices on the switched bus can then be serviced or repaired without disturbing operations on the processor buses.

Figure 2-48 illustrates the role of the DT07 hardware with respect to the Unibuses in a dual-processor system. In (a) is shown an idealized analog of a 2-port switch, where each port is a set of switches or relays making or breaking an electrical connection between a main Unibus and the switched Unibus. In the Connected position, the switched Unibus is part of the main Unibus; in the diagram, therefore, full communication can take place between Main Unibus A and the switched Unibus. In the Neutral position, the switched Unibus is totally isolated from the main Unibus associated with that port; so in the diagram, no connection exists between Main Unibus B and the switched Unibus. Communication between the control logic in each port assures that only one port is in the Connected position at any one time.

Section (b) of Figure 1-2 shows a simplified diagram of the actual DT07 implementation. Rather than relays, electronic circuits are used. Each Unibus line has connected to it a receiver and a driver. Under direction of the Control section, a signal received on one bus may be driven onto the other bus. The Control section can break the connection by disabling the drivers for both buses, placing the port in the Neutral position. Electronic ciruits, rather than relays, are used in the DT07 because of their inherent reliability and their ability to "repeat" the bus signals, allowing a group of devices on the switched bus to appear as only one electrical load on the main bus.



(A) IDEALIZED RELAY MODEL



(B) ELECTRONIC STRUCTURE OF DTØ7

FIGURE 2-48 DTØ7 SIMPLIFIED FUNCTIONAL DIAGRAM

2.10.2.2 Control

With the optional manual control panel, included as part of a BA511-K Switched-Bus Foundation box, the operator can enable either manual or programmable control of each switch section. When in manual mode, control is derived from a toggle switch that either enables or disables the bus signal-flow through the switch section. In the programmable mode (which is always enabled when the manual control option is absent) switch operation is under processor control. Each processor can request control of the switched Unibus and, when all other processors have relenquished control (either purposely or by timeout of their watchdog timers), can become connected to the switched bus. If two or more processors request use of the switched bus simultaneously, a priority arbitration circuit within each switch section specifies which processor will be serviced first. The priority circuit assures that no more than one processor at a time is connected to the switched bus.

In the typical system, with each switch section running in the programmable mode, the processor designated as the primary will initially request and gain control of the switched bus by setting a programmable request bit. The backup processor(s) will be programmed to periodically request use of the switched Unibus; the primary processor is notified of this request via a program interrupt and its watchdog timer is started. The primary processor can then behave in one of three ways: (1) it can actively dismiss the request from the other processor and retain control of the switched bus, thereby indicating that it is still functioning; (2) it can actively release control of the swiched bus if it has no further use for it, allowing control to be passed to the requesting processor; or (3) if the primary processor has failed it cannot actively respond to the request, thereby allowing the watchdog timer in the port module to expire and forcing a release from the primary processor and connection to the secondary.

CHAPTER 3
OPERATION AND PROGRAMMING

3.1 SCOPE

This chapter summarizes the functions of the operating controls and indicators used on the PDP-11/70 Multiprocessor System. It also includes a description of the jumpers and switches that are used for selection of device and vector addresses and various other options that effect operation.

3.4 BOOT/IIST SWITCH PANEL Switches and Indicators

Summaries of the operating functions of the BOOT/IIST panel and switches indicators are contained in Tables 3-1 and 3-2 respectively. The panel itself is shown in Figure 3-1.

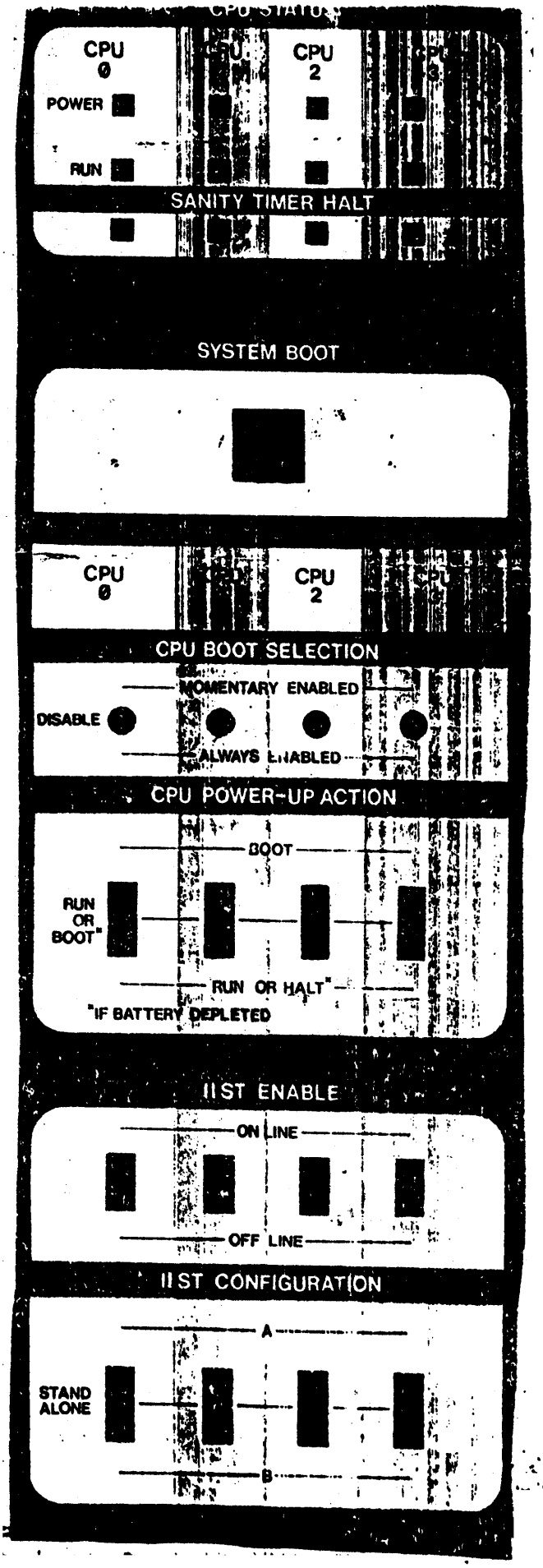


Figure 3-1 BOOT/IIST PANEL

Table 3-1 BOOT/IIST CPU STATUS PANEL INDICATORS (Sheet 1 of 2)

INDICATOR	CONDITION	PURPOSE
POWER CPU 0 - CPU 3 -	ON	Indicates processor associated with lighted indicator is powered up and does not have AC LOW or DC LOW asserted on either Unibus or Memory Bus.
RUN CPU 0 - CPU 3 -	ON	Indicates processor associated with lighted indicator is in RUN state and either signal TMCA CONF (1) L or TMCF PAUSES L is not asserted.
	OFF	Indicates processor associated with OFF indicator is halted or in pause (TMCA CONF or TMCF PAUSES L is asserted.)
SANITY TIMER HALT CPU 0 - CPU 3 -	ON	Indicates Sanity Timer has timed out and processor associated with lighted indicator has received IIST HALT L signal from its IIST Interface Module (IIST HALT L signal causes UBCF STOP L signal to be generated on M8176 Board).

INDICATOR	CONDITION	PURPOSE
SANITY TIMER	ON (Continued)	NOTE: HALT indicator lights when Sanity Timer times out only if switch S1 on IIST Interface Module is ON.

Table 3-1 BOOT/IIST CPU Status Panel Indicators (Sheet 2 of 2)

Table 3-2 BOOT/IIST Panel Switches (Sheet 1 of 4)

SWITCH	SWITCH POSITION	FUNCTION
SYSTEM BOOT	Pressed momentarily and released.	Sends a boot pulse to the external boot input of all M9312 Boot modules of processors CPU 0 through CPU 3 enabled by the CPU BOOT SELECTION switches.
CPU BOOT SELECTION	MOMENTARY ENABLED (Momentary contact with spring return to DISABLE).	Enables boot pulse circuit to M9312 module of associated processor (CPU 0 through CPU 3) only while manually held in this position.
	(CPU 0 through CPU 3) DISABLE	No BOOT results when SYSTEM BOOT switch is pressed to any processor whose CPU BOOT SELECTION switch is in DISABLE position.
	ALWAYS ENABLED	Boot pulse will always be sent to M9312 module of associated processor (CPU 0 through 3) when SYSTEM BOOT switch is pressed.
CPU POWER-UP ACTION	BOOT (CPU 0 through CPU 3)	Causes boot pulse to M9312 of associated pro-

SWITCH	SWITCH POSITION	FUNCTION
	BOOT (CPU 0 through CPU 3) (Cont)	cessor (CPU 0 through CPU 3) on any power-up of that processor.
	RUN OR BOOT (IF BATTERY DEPLETED)	Causes boot pulse to M9312 module of associated processor (CPU 0 through CPU 3) during power-up sequence only if memory back-up battery is depleted, otherwise CPU goes through normal power-up.
	RUN OR HALT (IF BATTERY DEPLETED)	No boot on power-up (power-up boot is disabled). CPU goes through normal power-sequence. If memory back-up battery is depleted, CPU traps to 0 and halts.
IIST ENABLE	ON LINE (CPU 0 through CPU 3)	Enables IIST Interface Module On IIST Bus (drivers and receivers) of associated processor (CPU 0 through CPU 3) to transmit and to receive from itself and from any other processor's IIST that is grouped on its system as determined by IIST CONFIGURATION switches.

Table 3-2 BOOT/IIST Panel Switches (Sheet 2 of 4)

Table 3-2 BOOT/IIST Panel Switches (Sheet 3 of 4)

SWITCH	SWITCH POSITION	FUNCTION
	OFF LINE (CPU 0 through CPU 3)	Disables IIST Interface Module (drivers and receivers) of associated processor (CPU 0 through CPU 3) from transmitting to and receiving from itself and from any other processor's IIST that is on its system as determined by IIST CONFIGURATION switches NOTE This switch position overrides the IIST configuration switches.
IIST CONFIGURATION	SYSTEM A (CPU 0 through CPU 3)	Enables IIST Interface Module of associated processor so that it can transmit to and receive from any other processor's IIST that is also in SYSTEM A position.

Table 3-2 BOOT/IIST Panel Switches (Sheet 4 of 4)

SWITCH	SWITCH POSITION	FUNCTION
	STAND ALONE	Disables IIST Interface Module of associated processor so it cannot receive from or transmit to any other IIST Interface Module. NOTE Unlike OFF LINE position, associated processor's IIST Interface Module in STAND ALONE position can transmit to and receive from itself.
	SYSTEM B	Enables IIST Interface Module of associated processor so it can transmit to and receive from any other processor's IIST that is also in SYSTEM B position.

3.5 IIST Interface Module (78717) Switches

The M8717 IIST interface module contains various switches used to select the device address, the interrupt vector address, the Self-ID number, and system Lockup (Halt) options. These switches, located on the DIP-Switch packs at I.C. locations E109, E86, and E5 on the M8717, are described in the following paragraphs. Refer to Figure 3-2 for the locations of the switches on the M8717 module and Figure 3-3 for typical examples of switch settings.

3.5.1 Device Address Select Switches

The device address of the IIST (e.g., the Unibus address of the ACR register) is selectable via switches 1 through 10 on the switch pack at E109 and switch 3 on the switch pack at E86, representing address bits 12 through 3, respectively. Address bits 13 through 17 are not selectable since the device address must lie in the Unibus peripheral address page; these bits must all be 1 in order to access the IIST device registers. Bit 1 of the address is used internally to select between the ACR and ADR registers and therefore does not come under switch control. A switch in the OFF position selects the corresponding address bit to be a 0 a switch in the ON position selects the address bit to be a 1. An example of the switch configuration required to select the device address of 764000, is shown on sheet 1 of Figure 3-3.

3.5.2 Interrupt Vector Address Select Switches

The interrupt vector transferred to the processor is selectable via switches 4 through 10 on the switch pack at E86, representing vector bits 8 through 2, respectively. A switch in the OFF position represents a 1 and a switch in the ON position represents a 0.

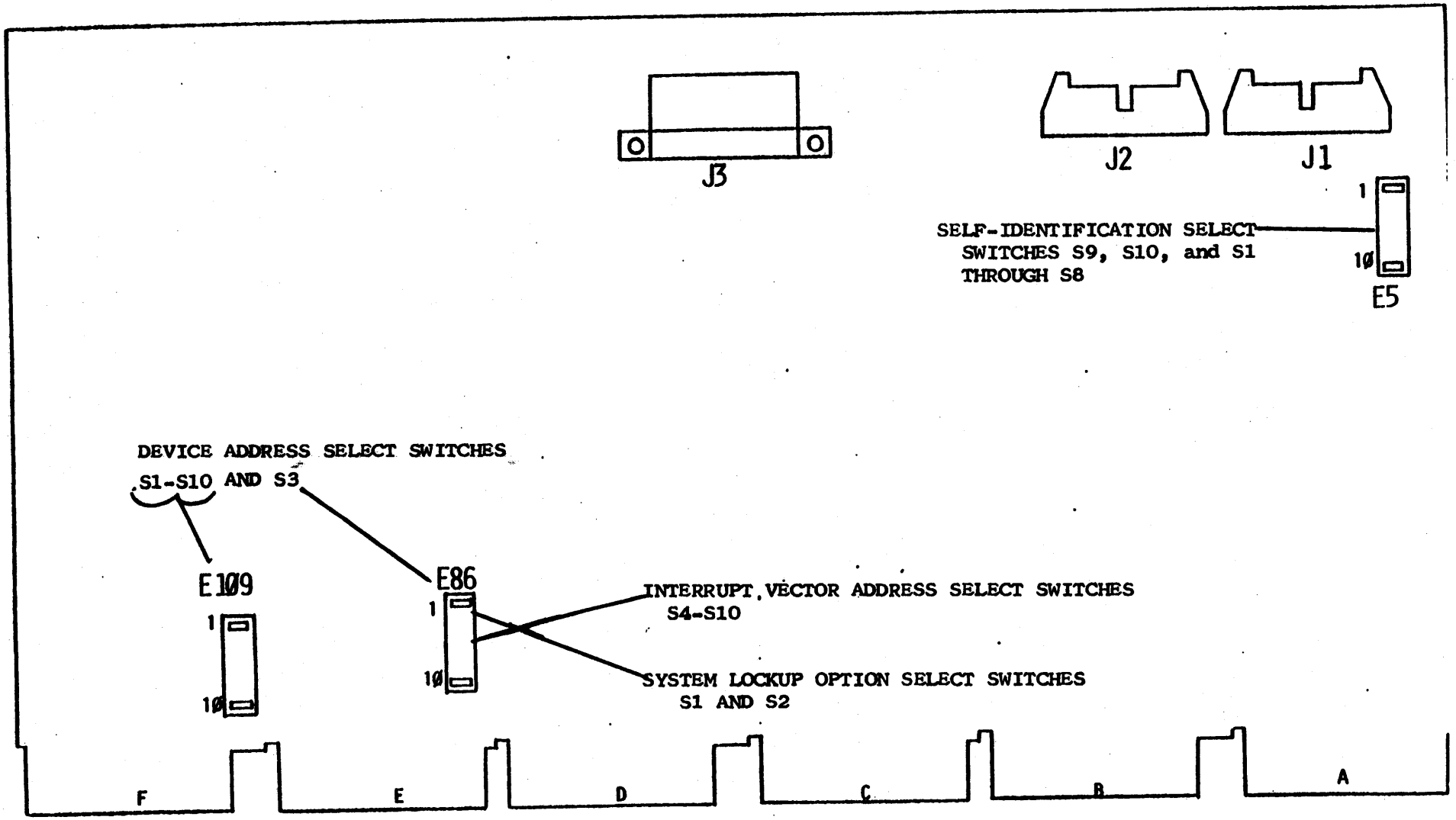


FIGURE 3-2

M8717 MODULE PHYSICAL LAYOUT

SWITCH	LOCATION	SWITCH SETTING LOGIC			PURPOSE
		SWITCHES	SETTING	LOGIC LEVEL	
Device Address Select Switches	IIST Interface Module	S4 1-10 at E106 and S3 at E115	OFF	0	Selects device address of IIST (i.e., address of Access Control Register - ACR)
			ON	1	

NOTE:

EXAMPLE OF Standard IIST DEvice Address--764000:

ADDRESS BITS-	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
LOGIC LEVEL--	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	
SWITCHES						1	2	3	4	5	6	7	8	9	10	3	
POSITION						OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

ALWAYS 1 (NOT SELECTABLE)

Not under switch control. Selects between ACR and ADR Register.

DEVICE ADDRESS SWITCHES

Figure 3-3 IIST Address, Self-Identification, and Lock-Up Switches (Sheet 1 of 4)

SWITCH OR INDICATOR	LOCATION	SWITCH SETTING LOGIC			PURPOSE
		SWITCHES	SETTING	LOGIC LEVEL	
Interrupt Vector Address Select Switches	IIST Interface Module (On switch pack at E115)	SW4, 9, and 10	OFF	0	Selects the Interrupt Vector that is transferred to the Processor.
			ON	1	
		SW5--8	OFF	1	
			ON	0	

NOTE:
Example of Standard IIST VECTOR ADdress Selection--170

VECTOR ADDRESS BITS--	8	7	6	5	4	3	2
SWITCHES AT E115 --	4	5	6	7	8	9	10
SWITCH POSITION	OFF	ON	OFF	OFF	OFF	ON	OFF
LOGIC LEVEL --	0	0	1	1	1	1	0

1 7 0

INTERRUPT VECTOR ADDRESS SWITCHES

Figure 3-3 IIST Address, Self-Identification, and Lock-Up Switches (Sheet 2 of 4)

SWITCH ~~OR INDICATOR~~	LOCATION	SWITCH SETTING LOGIC			PURPOSE
		SWITCHES	SETTING	LOGIC LEVEL	
Self-Identification Select Switches	IIST Interface Module (On Switch Pack at E19)	SW9-10	ON OFF	0 1	Selects Self-identification code of IIST INTERface (processor) Selects bits 1 and 0, respectively, of the ID code appearing in the ACR register.
		SW1-8			Controls the gating of the serial transmit data and clock signals on to the IIST Bus.

NOTE:
Four possible Processor or IIST self-identification codes (0 through 4) may be chosen. Only the switch settings shown in the following table are valid and no others may be used.

SELF ID CODE	SWITCH NUMBER									
	1	2	3	4	5	6	7	8	9	10
0	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON
1	OFF	OFF	ON	ON	OFF	OFF	OFF	OFF	ON	OFF
2	OFF	OFF	OFF	OFF	ON	ON	OFF	OFF	OFF	ON
3	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF	OFF

SELF-IDENTIFICATION SWITCHES

Figure 3-3 IIST Address, Self-Identification, and Lock-Up Switches (Sheet 3 of 4)

SWITCH ~~OR INDICATOR~~	LOCATION	SWITCH SETTING LOGIC			PURPOSE
		SWITCHES	SETTING	LOGIC LEVEL	
System Lock-up Option Select Switches	IIST Interface Module (On switch pack at E115)	SW 1 and 2			Selects method used to freeze the state of the processor and its Unibus

NOTE:
The method for freezing processor operation when the Sanity Timer times out can be selected by means of the System Lock-Up option Select switches.

SWITCH	SETTING	CONDITION GENERATED
SW1 SW2	ON ON	Direct hardware Lock line to the Processor that freezes the Processor when Sanity Timer times out.
SW2 SW1	OFF OFF	When Sanity Timer times out system is locked via an NPR (Non-Processor Request) request on the Unibus with subsequent holding of Bus mastership.
SW1 SW2	OFF ON	No System Lock-up is generated.

SYSTEM LOCK-UP OPTION SWITCHES

Figure 3-3 IIST Address, Self-Identification, and Lock-Up Switches (Sheet 4 of 4)

3.5.3 Self-Identification Select Switches

The Self-ID code associated with the IIST interface is selectable via the switches on the switch pack at E5. Switches E5-9 and E5-10 select bits 1 and 0, respectively, of the 2-bit ID code appearing in the ACR register. Switches E5-1 through E5-8 control the gating of the serial transmit data and clock signals onto the IIST bus. Switch configurations for the four possible Self-ID codes (0 through 3) are given on Sheet 3 of Figure 3-3.

3.5.4 System Lockup Option Select Switches

The method used to halt (lockup) the processor is selectable via switches 1 and 2 on the switch pack at E86. When switch E86-1 is ON, the direct hardware Halt line to the processor (or the backplane HALT REQ signal) is used. When switch E86-2 is OFF, the system is halted via an NPR request on the Unibus, with subsequent holding of bus mastership. Sheet 4 of Figure 3-3 shows the settings of the system lock-up switches.

3.6 FRONT PANEL SWITCHES AND INDICATORS OF Memory Port Controller

The operating functions of the memory port controller switches and indicators that are used to control the operation of each individual port on the MKA11 multiported memory are shown in Table 3-3 Location of the switches and indicators on the memory box panel are shown in Figure 3-4.

3.7 RH01 DUAL-ACCESS ADAPTER

3.7.1 Switches and Indicators

The Indicator and Control panel of the RH01 dual access adapter is shown in Figure 3-5 The operating functions of the switches and indicators are described in Table 3-4.

Figure 3-3 RH01 Indicator and Control Panel

SWITCH	INDICATOR	FIGURE 2-1 Reference Number	INDICATOR CONDITION	SWITCH POSITION	PURPOSE
STARTING ADDRESS		1		Depends upon memory starting address desired.	Selects octal address (in 16K X 32 bit blocks) of first memory location.
INTER LEAVE		2		0= No external interleaving. 1= Not used. 2= First box of two-way interleaved system. 3= Second box of two-way interleaved system. 4= First box of four-way interleaved system. 5= Second box of four-way interleaved system. 6= Third box of four-way interleaved system. 7= Fourth box of four-way interleaved system.	Selects type of external interleaving, number of memory boxes externally interleaved and order in which interleave is performed.
	PANEL SELECTED	3	Lighted		Indicates starting address and external interleaving selected by controller.
ADDRESS/INTERLEAVE		4		ALLOW PROG CONTROL	Permits Operating System to select starting address and type of interleaving through Control Status Register (CSR).
				FORCE PANEL	Enables controller's Starting Address and Interleave switches.

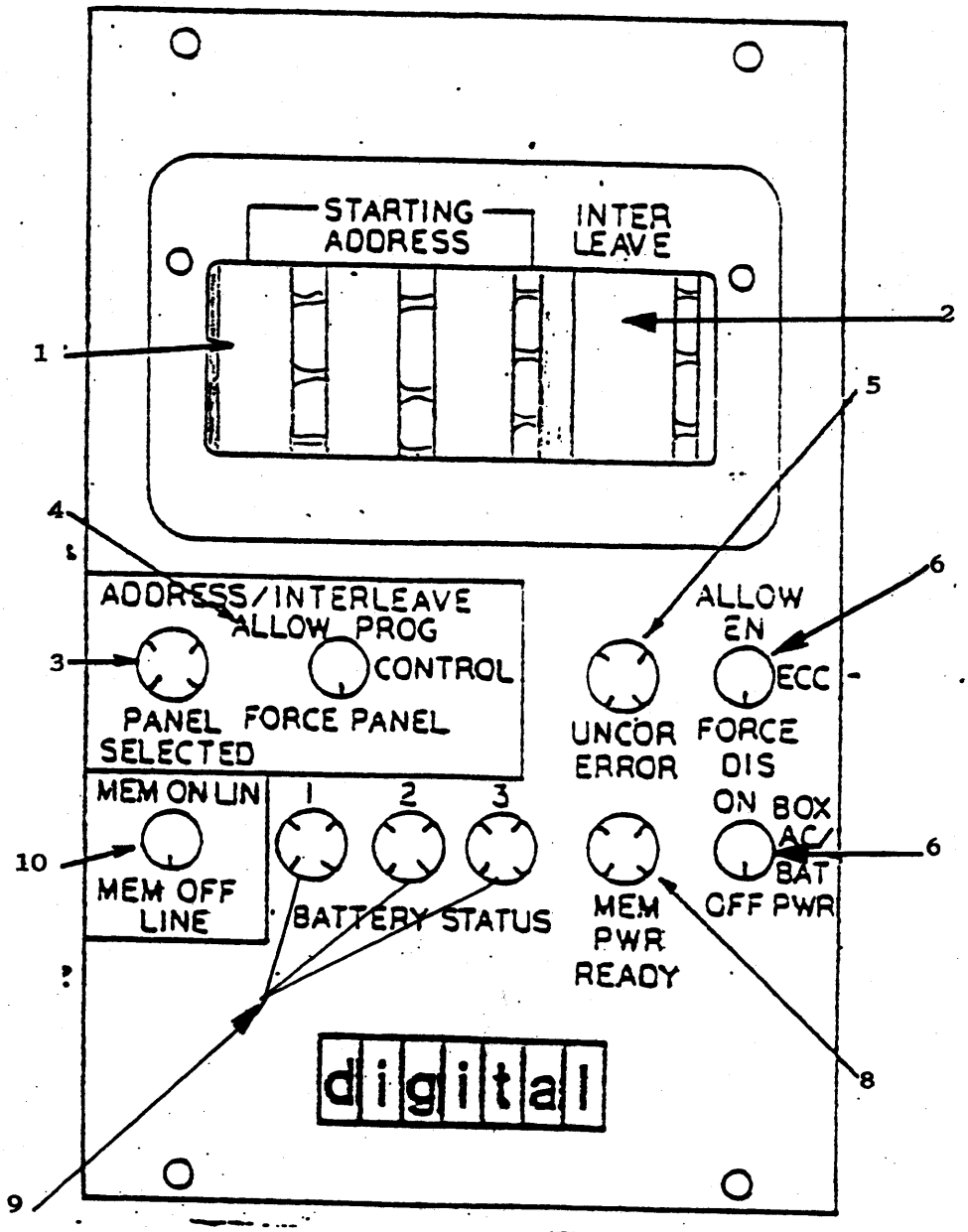
TABLE 3-3 FRONT PANEL SWITCHES AND INDICATORS OF PORT CONTROLLER (Sheet 1 of 3)

SWITCH	INDICATOR	FIGURE 2-1 REFERENCE NUMBER	INDICATOR CONDITION	SWITCH POSITION	PURPOSE
	Uncorrected Error	5	Lighted with ECC Enabled		Indicates uncorrectable multiple bit error detected.
			Lighted with ECC Disabled		Indicates single or multiple bit error detected.
Allow ECC Enable/ Force ECC Disable		6		Allow ECC Enable	Allows correction of all single bit data errors.
				Force ECC Disable	Disables all error correction (except in protected memory blocks).
On Line/ Off Line Memory		10		On Line	Places memory on line with processor, and it will respond to commands from Bus Master.
				Off Line	Places Memory Off-Line with Processor and it wil not respond to commands from Bus Master.
NOTE: BATTERY STATUS indicators work only on first port controller of every memory box. Are not operable on any other port controllers of the same memory box.	1 2 3 Battery Status	9	Off		Battery discharged or disconnected from memory.
			Slow Blink		Battery charging
			Fast Blink		Battery supplying power to memory.
			Lighted (Constantly)		Battery fully-charged.
NOTE: Works only on first port controller.	MEM PWR READY	8	Lighted		Indicates memory is on and DC voltages within operating limits.

TABLE 3-3 FRONT PANEL SWITCHES AND INDICATORS OF PORT CONTROLLER (Sheet 2 of 3)

SWITCH	INDICATOR	FIGURE 2-1 Reference Number	Indicator Condition	Switch Position	Purpose
Box AC/ Bat Pwr NOTE: This switch operates only on first port controller of every memory box. Does not operate on any other port controllers of the same memory box.		7		On	Switches primary AC and/or Battery back-up power to power supply and regulators.

TAB:E 3-3 FRONT PANEL SWITCHES AND INDICATORS OF



NOTE:
Numbers reference descriptions of switch and indicator functions contained in Table 3-.
One port controller is required for every port on every box of memory.

Figure 3-4 Port Controller (MKA11 Memory)

Figure 3-5 RH01 Indicator and Control Panel

SWITCH	INDICATOR	INDICATOR STATE	SWITCH POSITION	FUNCTION
PORT SELECT	NOTE The port selection on the RH01 must not be charged while data transfers are in progress through the RH01. Changing the port selection while transfers are taking place can result in errors		OFF CAUTION Power to the RH01 Is NOT turned OFF in this position.	Disables access from both controllers to shared device, (normally used during maintenance, device appears not to exist in this mode.) Attempts to access shared drive with RH01 in OFF results in nonexistent drive errors.
			A PORT	Permits access to the shared device only through processor A. Processor B cannot access the device. Nonexistent drive errors occur, if B attempts access.
			DUAL ACCESS MODE (3 possible states: unseized, seized by Port A, or Port B)	Permits both processors A and B to access device under program control requirements. 1) UNSEIZED- Not connected to either controller. Allows either controller to read or write any registers in shared drive(s). 2) SEIZED STATE-From unseized state, RH01 becomes seized by controller when it writes into any register or reads Control/ Status 1 (CS1) register in shared drive(s). Can be seized by either controller. When seized, attempted register transfer from other port does not result in nonexistent drive error. Attempted read of a register while RH01 is seized by other controller causes RH01 to respond with all zeroes. If write attempted, RH01 discards data, but does not cause non-existent drive error. If controller it is se ed by issues

Table 3-4 RH01 Switches and Indicators

SWITCH	INDICATOR	INDICATOR STATE	SWITCH POSITION	FUNCTION
				release command or Releast Timer in RhO1 times out, RH01 returns to unseized state.
			B PORT	Permits access to the shared device only through processor B, processor A cannot access the device. Nonexistent drive errors occur, if A attempts access.
SWITCH SELECT (Push Button)			PRESSED	When it is pressed. Changes the mode of the RH01 to whatever mode the port select switch is in (rotating the port select switch has no effect until this switch is pressed.) As soon as it is pressed, LED that is adjacent to the port select switch position should light.
	OFF (LED)	LIGHTED		Indicates RH01 in OFF.
	A PORT (LED)	LIGHTED		Indicates RH01 in A port position
	DUAL ACCESS (LED)	LIGHTED		Indicates device can be accessed by both A and B processors under program control.
	B PORT (LED)	LIGHTED		Indicates device can be accessed by B controller.

Table 3-4 RH01 Switches and Indicators (Cont)

SWITCH	INDICATOR	INDICATOR CONDITION	SWITCH POSITION	FUNCTION OR INDICATION
	A BUSY (LED)	LIGHTED		Indicates Port A has seized the device
	B BUSY	LIGHTED		Indicates Port B has seized the device

NOTE:
In dual-access mode, A and B BUSY Led's will alternately light indicating which processor is using the shared device, In manual mode (Port Select switch in either A or B Port) only the LED on the selected port will light when that port is using the drive.

Table 3-4 RH01 Switches and Indicators (Cont)

3.7.2 Manual Mode of Operation

The RH01 is in the Manual Mode when the port Select Switch on the front panel is set in either the PORT A or PORT B position. In the Manual Mode, the RH01 is completely transparent to the selected port. The RH01 simply repeats transfers between the controller and the shared drive(s). Any attempt to access the shared drive(s) from the port that is not selected is ignored and will result in a nonexistent drive error.

3.7.3 Dual Access Mode of Operation

When the RH01 is in the Dual Access Mode, access to the shared drive(s) is allowed from both ports under program control. In the Dual Access Mode, the RH01 can be in one of three possible states:

a. Unseized

b. Seized by Port A

c. Seized by Port B

3.7.3.1 Unseized State

The unseized state occurs when RH01 is not connected to either controller. In the unseized state, the RH01 will allow either controller to read or write any of the registers in the shared drive(s).

3.7.3.2 Seized State

From the unseized state, the RH01 will become seized by a controller anytime the controller either:

a. Writes into any register in the shared drive(s)

b. Reads the Control/Status 1 (CS1) register in the shared drive(s).

The seized state is similar in operation to manually selecting a given port. The major functional difference between the two operations is the RH01 response to an attempted register transfer from the other controller, If the RH01 becomes seized Dual Access Mode, an attempted register transfer from the other port does not result in a nonexistent drive error. If a controller attempts to read a register in a shared drive while it is seized by the other controller, the RH01 will respond by returning all zeroes. If the controller attempts a write, the RH01 will simply discard the data but will not cause a nonexistent drive error.

Once the RH01 has been seized by either controller, it will return to the unseized state if either the controller it is seized by issues a release command (octal 13) or the Release Timer in the Rh01 times out.

While RH01 is seized by one controller, the other controller may request the RH01 by writing into any register (e.g. writing zeroes into the CS1). If the RH01 is seized by the other controller, the request will be stored and the RH01 will connect as soon as it is released. Once the RH01 does connect, it will assert the Attention signal to the controller whose request has been stored. This Attention signal will cause an interrupt if the Interrupt Enable bit in the controller has been set. In the previous examples, where the RH01 has become seized from the unseized state, the RH01 connects in the same cycle as the register transfer and no interrupt is generated.

3.7.4 Hardware Registers

The RH01 will respond to two hardware registers; the Attention Summary register and the Control Status register.

3.7.4.1 Attention Summary Register

The Attention Summary register is a composite register consisting of the attention bits from each of the devices on the Massbus. Only the lower byte of this register is used (refer to Figure 3-6). When this register is read, each device will assert its attention bit in the register corresponding to its unit number on the Massbus. For example: if drive zero and drive one both have an attention condition, this register will read back with bits 0 and 1 set. The Attention Summary register in the RH01 is cleared by writing a one in the bit position to be cleared. This register is not cleared by initialize.

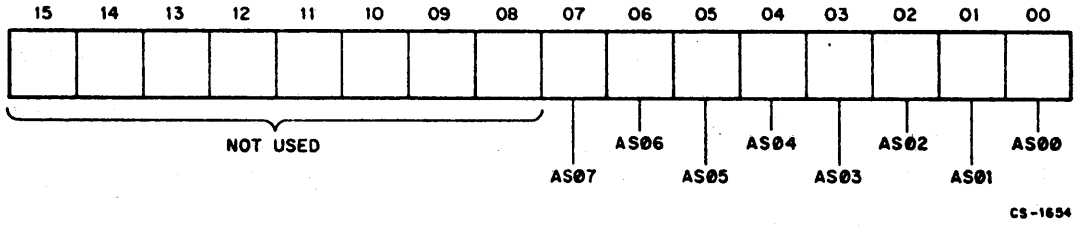


Figure 3-6 Attention Summary Register

Under certain conditions, the RH01 will interrupt the Massbus Controller by asserting the attention signal. Since the RH01 is not designated wit a device number on the Massbus, there is no corresponding bit position in the Attention Summary register to set. The RH01 must therefore simulate an interrupt from the drive that was requested. This simulation is accomplished by setting the bit in the Attention Summary register corresponding to its drive number.

An attention condition in the RH01 is not cleared by either Reset or Controller Clear. To clear the attention condition in the RH01, the bit in the Attention Summary register that has been set must be cleared by writing a one in that bit position.

3.7.4.2 Control/Status 1 Register (CS1)

The Control/Status 1 register of any Massbus device has the format shown in Figure 3-7

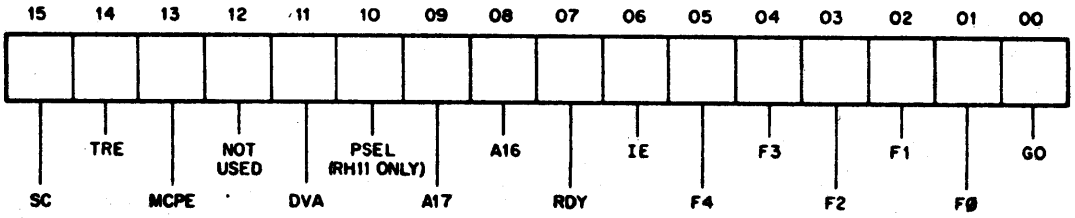


Figure 3-7 Control and Status 1 Register

Bits 00:05, 11 and 12 are stored in the Massbus device. The remaining bits are stored in the Massbus controller. For a detailed explanation of each bit in CS1, reference should be made to the maintenance manual of the Massbus device being used on the RH01.

When the CS1 register is read, bits 13:15 and 06:10 are read from the controller and bits 00:05 and 11:12 are read from the selected Massbus device. Although the function bits (01:05) are stored in the drive, each time the CS1 register is written the function bits are also written into a duplicate register in the controller. The controller then knows the direction of a data transfer command. The duplicate register in the controller is transparent to the user. When the function bits are read, they are read from the register in the Massbus device and not from the duplicate register in the controller.

Similarly, the RH01 maintains a duplicate of the function bits so the direction of the data transfer can be determined. The register always stores the last command issued to the shared drive. The duplicate register in the RH01 cannot be read. The contents of the CS1 register are read from the shared drive and not the RH01. Each time the CS1 in the shared drive is written, a copy of the information is stored in the duplicate CS1 of the RH01. The only command in the CS1 that the RH01 will respond to directly is the release command (octal 13).

3.7.4.2.1 Reading The Control/Status Register

Reading the Control/Status 1 register (CS1) will cause the RH01 to become seized only if the RH01 was unseized when the register was read. Reading the CS1 register is generally a test to see if the drive is available. The Drive Available (DVA) bit in the drive is hardwired to a one. Therefore, if the CS1 can be read through the RH01 (the RH01 is not seized by the other port) it will be read as a one. If the RH01 is seized by the other port, reading any register will result in all zeros and therefore the Drive Available bit will not be set. These features make it possible to look for an available unit by scanning a number of drives on individual RH01's.

3.7.5 Programming

After a shared device has become seized through the RH01, (either manually or programmably) the RH01 will be completely transparent to the software on that port until the drive is released.

Aside from any higher level software that may be necessary for interprocessor communication, the modifications required on a single port device handler for use with a dual port RH01 are minimal.

The major addition necessary to the handler is the software required to obtain control of the RH01 before executing the remainder of the handler code and releasing the RH01 (if desired) after the handler is finished.

If the RH01 is to be used programmable (dual access mode) it is recommended that an algorithm similar to the one shown in Figure 3-6 be used to seize and release the RH01. If the RH01 is to be used manually, no additional software should be necessary since any attempt to use the drive from the Port that is not selected will result in an error.

3.7.5.1 Programming Example

Figure 3-8 and 3-9 illustrate the programming steps necessary to seize the RH01. Two methods are shown; one using an interrupt driven code (Figure 3-8) and the other using a test on the Device Available bit (Figure 3-9). The method using an interrupt driven code is strongly recommended since testing the Device Available bit repeatedly will unnecessarily slow down register transfers from the other controller.

NOTE

In the interrupt driven code, it is necessary to do a single test on the Device Available bit after requesting the RH01. This is necessary because if the RH01 had been unseized when the request was made, it would have switched immediately without generating an interrupt.

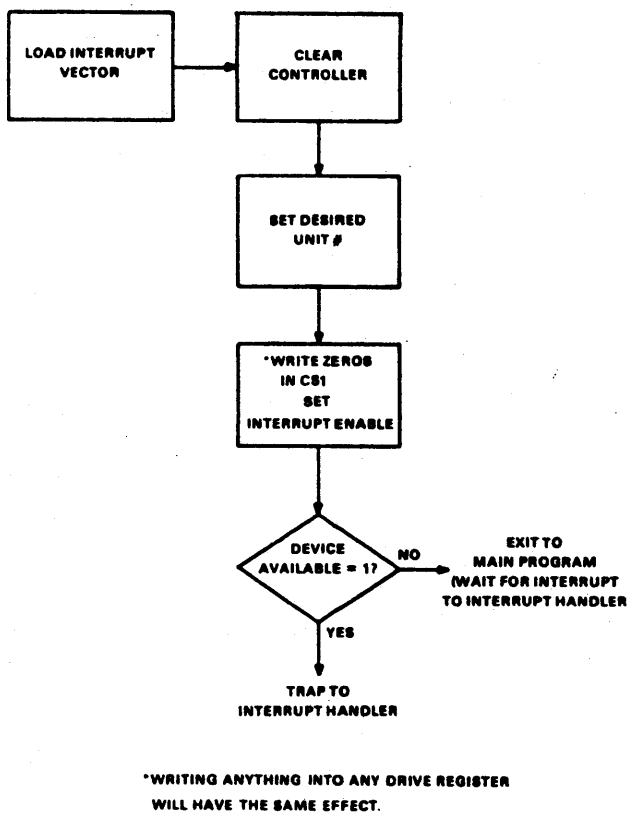


Figure 3-9 Algorithm to seize the RH01 (Interrupt Driven Code)

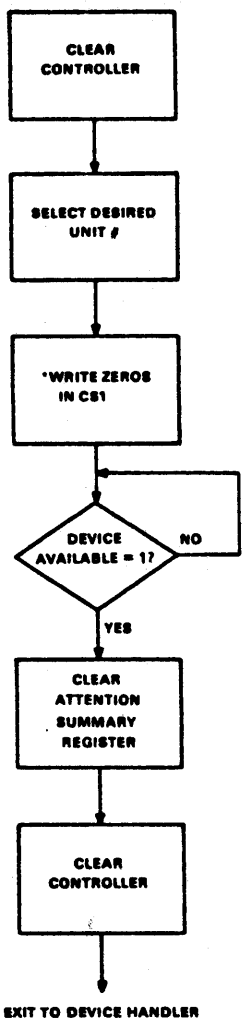


Figure 3-9 Algorithm to seize the RH01 (Non-Interrupt Driven Code)

3.8 DTO7 Bus Switch

Switches and indicators associated with the operation of the DTO7 Bus Switch are contained on the DTO7 Port Module (M8721) and the BAS11-K Maintenance /Control panel. Each DTO7 Port Module contains one toggle switch and one LED indicator. These are mounted on the handle-end of the M8721 port module.

When the Maintenance/Control panel of the BAS11-K is used, each DTO7 port module is controlled by two toggle switches and its status is indicated by one light. Figure 3-10 shows the BAS11-K panel. The lower row of switches are the four MODE switches, one for each port; the upper row contains the SELECT switches for each port. The controls for each port, labelled PORT 0 through PORT 3, are independent. Also contained on the panel is a rotary power switch. Each port module has its own identical set of controls. Switch and indicator functions are described in Table 3-5.

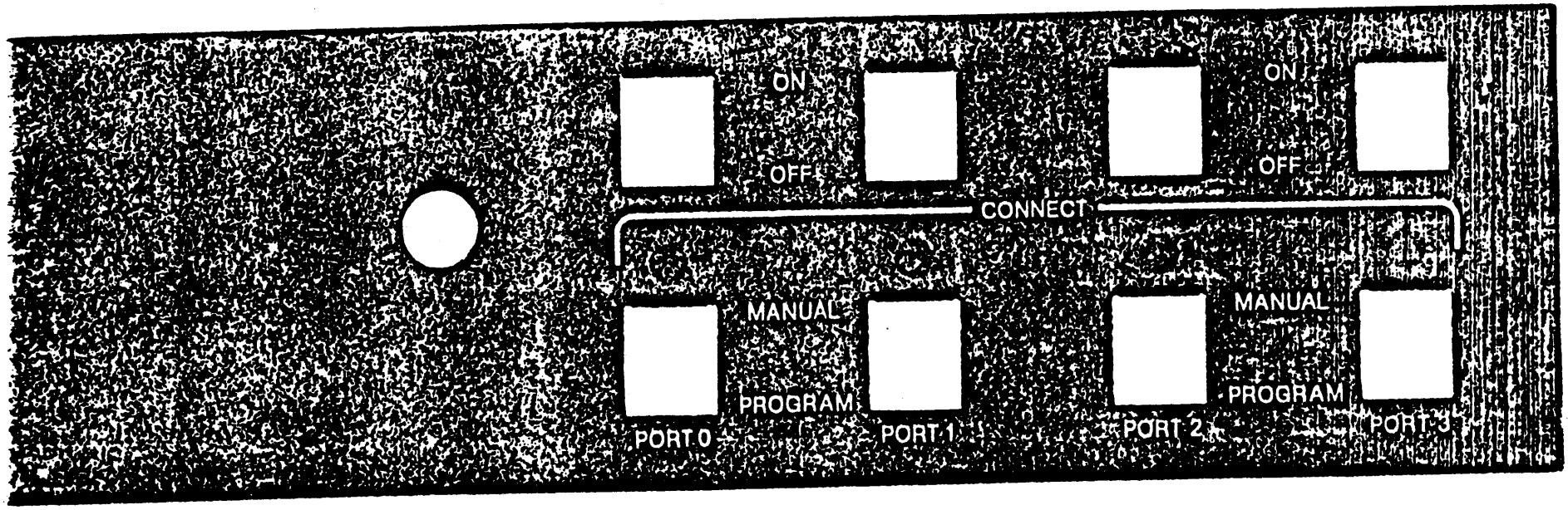


FIGURE 3-10

BAS11-K MAINTENANCE/CONTROL PANEL

Table 3-5 Control Switches and Indicators for DTO7 Bus Switch

SWITCH OR INDICATOR	LOCATION	SWITCH POSITION OR INDICATOR STATE	FUNCTION
PORT ENABLE (Toggle switch)	Port Module (M8721)	ON	Enables port module to drive or receive signals on the switched bus and inter-port control bus.
		OFF	Port Module is forced into Neutral switch position (i.e., disconnected from switched Unibus)and is inhibited from driving or receiving signals on switched bus and inter-port control bus.
CONNECT (LED)	Port Module (M8721)	Lighted	Indicates switch port is logically connected to switched bus, enabling transfers between main bus and switched bus.
		Not Lighted	Switch port is in neutral position
MODE (Switch)	BAS11-K Maintenance Control Panel (Lower row of switches	PROGRAM	Associated port module is controlled solely by program as if manual controls were not present.
		Manual	Disables PROGRAM mode. Programmed attempts to connect or disconnect port have no

Table 3-5 Control Switches and Indicators for DT07 Bus Switch

SWITCH OR INDICATOR	LOCATION	SWITCH POSITION OR INDICATOR STATE	FUNCTION
			effect. (Select switch is used to control port module.) Program can still read port status in Control-Status Register, enable interrupts, and cause programmed reset of switched Bus.
SELECT (switch)	BAS11-K Maintenance Control Panel (Upper row of switches)		Has effect only when MODE switch is in MANUAL position.
		OFF	Associated port is in neutral position. Main CPU bus is logically disconnected from the switched bus.
		ON	Moving SELECT switch to ON position causes associated port to request control of switched bus and then, when conditions permit to become logically connected to switched bus. Once manual connection is completed, requests from other ports are ignored unless port module itself is powered down. NOTE: Should SELECT switches for two or more ports

Table 3-5 Control Switches and Indicators for DT07 Bus Switch

SWITCH OR INDICATOR	LOCATION	SWITCH POSITION OR INDICATOR STATE	FUNCTION
			be ON at same time arbitration logic assures that only one port wins the connection usually port that was moved to ON first. When two switches are turned on at exactly the same time, port with highest priority (lowest port number as selected by switches on port modules) wins connection.
CONNECT (LED)	BAS11-K Maintenance Control Panel	ON	Indicates that associated port and its main Unibus are logically connected to switched bus.
POWER (3-position rotary switch)	BAS11-K Maintenance Control Panel	OFF	Turns off power supply in BAS11-K mounting box removing DC power from switched-bus equipment.
		ON	Applies power to switched-bus equipment .
		STNBY	Equivalent to OFF position, except that a contact-closure output is asserted, which may be used for special customer applications.

PAGES 3-37 to 3-44 CANCELLED

3.2 PDP-11/70 SWITCH CONSOLE

Refer to Section III, Chapter 1 of the KB11-C Processor Manual or to the PDP-11/70 Processor handbook for a detailed description of the switch console and its operation.

3.3 REMOTE SERIAL DIAGNOSTIC CONSOLE (Optional)

The KY11-R serial console is a device that replaces the switch console control panel with logic that interprets ASCII characters to perform equivalent panel functions. When installed, it is used to control and operate the individual PDP-11/70 processors making up the multiprocessor system. The ASCII data stream for the console is multiplexed over the same serial line as console terminal program I/O. The serial console is designed for remote use in conjunction with the Digital Diagnostic Center Host computer and has two serial line connections: one for a local terminal, and one for a data set.

3.3.1 Serial Console Control Panel Switches and Indicators

The functions of the control panel switches and indicators are contained in Tables 3-9 and 3-10 respectively.

3.3.2 CONSOLE COMMANDS

The console commands, which are divided up into Console Control Commands, CPU Control Basic Commands, and CPU Control ODT-11 Commands (similar to the PDP-11 debugger ODT-11 commands), are summarized in Table 3-11.

Table 3-9 Serial Console Control Panel Switches

SWITCH	SWITCH POSITION	FUNCTION
5-position key-switch (POWER OFF/LOCAL-DISABLE/ LOCAL/REMOTE-DISABLE/REMOTE)	LOCAL-DISABLE	Generally, controls access from serial interface and recognition of CONSOLE BREAK character. Forces console logic into PROGRAMII/O state. CONSOLE BREAK character recognition is disabled and all characters (including ^P) are passed to serial interface buffer. (Similar to panel lock position of hardware switch control panel.) Disables remote interface and clears Data Terminal Ready (DTR) to modem which generates carrier lost message (?CAR ER) on local terminal if carrier was present in previous key position. Standard position for running system.
	LOCAL	Used for normal system start-up, or whenever specific CONSOLE state operations are required.

NOTE

Entry to this position from REMOTE-DISABLE position disables remote ser-

SWITCH	SWITCH POSITION	FUNCTION
	LOCAL (CONT.)	ial interface and clears DTR to modem which generates carrier lost message on local terminal if carrier was present in previous key position.
		Operator can move between CONSOLE PROGRAM I/O and TALK states by using console command set. Entering TALK state temporarily enables remote serial interface and asserts DTR to modem.
	REMOTE-DISABLE	Operation similar to LOCAL-DISABLE except remote serial interface is enabled, DTR is asserted to modem and Local Copy bit is forced ON. Remote line performs normal program I/O functions, i.e., run Error-Logger, but CONSOLE state functions are disabled.
		NOTE Entry to this key position from REMOTE position clears TEST indicator if it was ON.

SWITCH	SWITCH POSITION	FUNCTION
	REMOTE-DISABLE (Continued)	Generally used with customer software running and with local terminal set to Local Copy mode to duplicate locally what is sent to remote serial interface. Local keyboard is disabled in this position to prevent interaction with remote operator. NOTE Only output to remote serial interface is copied to local terminal. If password sequence were entered both local and remote copy would cease until password sequence had been completed and system resumed echoing.
	REMOTE	Same as LOCAL position except remote serial interface is enabled and DTR is asserted to modem. Operator with ASCII terminal or host computer of Diagnostic Center may move between CONSOLE, PROGRAM I/O, and TALK states of opera-

SWITCH	SWITCH POSITION	FUNCTION
	REMOTE (Continued)	tion. Remote operator may boot system, perform program I/O functions with operating system and use TALK state to print information on or solicit inputs from the local terminal. Used almost exclusively for system-wide diagnosis in standalone condition. NOTE In this position the local terminal is initially disabled, however, enabled/disabled option bits can be programmed so that local copy-of remote activity can be obtained on the system operated local terminal locally to demonstrate a fault.
POWER-OFF	POWER-OFF	Removes all power from serial ASCII console.
POWER-FAIL (Slide Switch)	RUN 1'S	Allows CPU to power-up running with switch register set to ones.

SWITCH	SWITCH POSITION	FUNCTION
	RUN O'S	Allows CPU to power-up running with switch register set to zeros.
	HALT	Disables power-fail recovery and power-up in console state.
LAMP TEST (unmarked Momentary pushbutton)	Momentarily Depressed	Causes all control panel indicators to light.

NOTE

This switch is independent of any other control and can be pressed at any time.

Table 3-10 Serial Console Control Panel Indicators

INDICATOR	CONDITION	MEANING
POWER-ON	Lighted	Serial console device has proper voltages applied. NOTE It is possible for part of CPU to lose power and still have this indicator ON.
DISABLE	Lighted	Serial console is in panel lock condition. (Serial console is forced to PROGRAM I/O state and CONSOLE BREAK character recognition is disabled.)
REMOTE	Lighted	Keyswitch is in REMOTE-DISABLE or REMOTE position.
CARRIER	ON	Remote line data set has "carrier detected" signal asserted.

INDICATOR	CONDITION	MEANING
TEST	Lighted	Valid connection has been established with Diagnostic Center Host Computer. This indicator is turned OFF upon completion of testing. NOTE This indicator is cleared at power-on, and whenever keyswitch is returned to either LOCAL, LOCAL-DISABLE or REMOTE-DISABLE position.
FAULT	Lighted	Fault condition has been detected in serial console itself during internal operating checks or during self-test routine initiated by power-on or self-test command. If FAULT error is detected at power-on all other indicators may be on with FAULT indicator. Upon detection of fault condition, console sets FAULT indicator ON, prints error message (?CON ER) and attempts to resume normal operation.

INDICATOR	CONDITION	MEANING
FAULT	Lighted (Continued)	NOTE It is possible to continue using console even though FAULT indicator is ON, but this is not recommended. FAULT indicator can only be reset by powering down system.

Table 3-11 Summary of Console Commands

COMMAND	COMMAND MEANING	RESPONSE	ERRORS
	CONSOLE CONTROL COMMANDS		
^E	Read CPU Identity	11/70 V01-01~	None
^L	Set TALK state	^L	None
^P	Set CONSOLE state	^P<NL>CONS=~	None
^R	Load Control Register	^R~	~~SYN~~ None
^U	Clear Data Type-In	^U<NL>	None
S	Set Register Operation	None	SYN
,	Data Argument Separator	None	None
0-7	Octal Data	None	None
V	Verify Console Logic	V000377/<NL>	FAULT,IRN
Z	Set Program I/O State	Z~<NL>	None
RU	Delete Prev Data Digit	\X~	None
	CPU CONTROL BASIC COMMANDS		
(a)A	Display Address	a[ADD]~	None
C	Continue Program	~	IRN,#
(a)D	Deposit	~	SYN,IRN,MR,#
E	Examine	a[DD]~	SYN,IRN,MR,#
H	Halt CPU	[ADD]/I[S]~	#
I	Initialize CPU	~	IRN,#
S	Set Single Cycle	~	IRN
R	Reset Single Cycle	~	None
aL	Load Address	~	SYN,IRN,#
(a)M	Read Data Display	[DD]~	None
(a)N	Execute Next Instruction	[AD]~	IRN,#
R	Read Switch Register	[SWR]~	None
a,aS	Start Program	~	SYN,IRN,#
T	Read CPU Status	[STATUS]~	None
U	Read UNIBUS lines	[UNIBUS]~	None
aW	Write Switch Register	~	SYN
	CPU CONTROL ODT-11 COMMANDS		
a^D	Dump Memory	<NL>[ADD]/[DD]	SYN,IRN,MR,#
(a)LF	Open Sequential Location	NL,[ADD]/[DD]~	SYN,IRN,MR,#
(a)CR	Close Location	CR/LF	MR,#
a/	Open Location	[DD]~	SYN,IRN,MR,#
(a)@	Open Indirect Location	NL,[ADD]/[DD]~	SYN,IRN,MR,#
a,aG	Start Program in P1/O	NL	SYN,IRN,MR,#
P	Continue Program in P1/O	NL	IRN,#

NOTES:

1. "(a)" = Optional argument
2. "a" = Required argument
3. "~" = Command Acknowledge (040)
4. "AD" = Address Display in octal
5. "DD" = Data Display in octal
6. "SYN" = Syntax Error (7.1)
7. "IRN" = Illegal while Running (7.2)
8. "MR" = Memory Reference Error (7.3)
9. "#" = CPU Time-Out (7.6)