EK-KD11K-TD-PRE

# KD11K PROCESSOR (PDP-11/60) TECHNICAL DESCRIPTION MANUAL SECTION 6 MAINTENANCE FEATURES

PRELIMINARY

1st Edition, January 1978



Printed in U.S.A.

CONTENTS

## CONTENTS (CONT)

## CONTENTS (CONT)

## CONTENTS (CONT)

## TABLE

NOTICE

The technical information in this document has been reviewed and approved by engineering and field service. This information will be published in final typeset form as section 6 of the KD11-K Processor (PDP-11/60) Technical Description Manual.

# CHAPTER 1
# INTRODUCTION

## 1.1 SCOPE

This section of the KD11K Technical Description Manual provides a detailed description of the PDP-11/60 Maintenance Features. These built-in features are both software (system diagnostics, microdiagnostics, etc.) and hardware (parity, error status registers, microbreak, single clock, etc.).

PDP-11/60 troubleshooting is accomplished using three major approaches. Each of these approaches implementes microdiagnostics to isolate faulty modules. The approaches are:

1. Self Diagnosis -- Internal diagnostics check the processor, cache, and 28K of main memory automatically upon loading but may be disabled by altering micro-switches on the M9301-YH Module.

2. On-Line Diagnosis -- More than 90 internal registers may be accessed through the console, using the Maintenance, Examine and Deposit Procedure. Error flags and control bits stored in some of the internal registers are treated as data, to be used for system error analysis.

While the system is in operation, extensive parity checking is performed on data and memory addresses. The system employs one parity bit for every eight bit byte. When single-bit parity errors are detected, error information is recorded (Error Logging). Byte parity checking is provided:

a. For all memory (cache and core); on addresses and data in cache, and on data in core

b. Between the processor and cache for cache data and tag.

3. Diagnostic Control Store (DCS) Module -- A hex board containing its own microdiagnostics in a 2K X 52 bit ROM array, with 14 error indicator LEDs (32 indicator LEDs total), four switches, control logic and interface; employed to check the basic processor modules (except the M7875 Cache module) which has its own macrodiagnostics). The DCS Module can be utilized in the following manner:

a. DIAG key switch on Console - When the PDP-11/60 processor is in the halt state; the DIAG key switch, in conjunction with the CNTRL key switch, initiates the DCS module's microdiagnostics. This provides a "confidence check" on the system and gives a go/no-go indication on the console by displaying a predefined

octal number (1.2.3.3.2.1), with decimal points lit.

b. Verify/Norm Control switch on DCS module - This switch, when put in the Verify position, allows the DCS module to perform a diagnostic check of itself. This can be initiated from the console (pressing the DIAG key while holding in the CNTRL key) or by the Run/Stop control switch located above the Verify/Norm Control switch on the DCS module. Errors are indicated by the Error LED and an octal code is displayed in the remaining 12 error LEDs on the Module. If the Verify pass is successful, the console displays the octal number 2.1.2.1.2.1, with decimal points lit. Also, the Error and EOP (End of Pass) LEDs illuminate if the Verify routine is successful.

c. MED (Maintenance, Examine and Deposit) Instruction Access - To initiate a confidence check of the processor, by the DCS Module's diagnostics, the Maintenance, Examine and Deposit procedure may be employed. This is performed by loading the NUA (Next Micro-Address) Register, (Write Function Code 350), with a 11410 (starting address of the DCS Module's diagnostics). This procedure initiates the micro-diagnostics in the same manner as utilizing the DIAG/CNTRL key switches.

The main purpose of these maintenance aids (to be discussed in detail in the following Sections) is to allow the Field Service engineer to maintain the 11 Family's repair philosophy of faulty module replacement. With these maintenance features, faulty module isolation is made possible.

## 1.2 MAINTENANCE EQUIPMENT REQUIRED

Table 1-1 lists special/standard equipment required to maintain a PDP-11/60.

## Table 1-1 Maintenance Equipment Required

| Equipment or Tool | Manufacturer | Model, Type or Part No. | DEC Part No. |
|---|---|---|---|
| Oscilloscope | Tektronix | 465* | |
| Digital Voltmeter (DVM) | Weston (or the like) | 6000 | |
| Volt/Ohmmeter (VOM) | Triplett | | 29-13510 |
| Unwrapping Tool | Gardner-Denver (DEC Catalog #H812A) | 505 244-475 | 29-18387 |
| Hand Wrap Tool | Gardner-Denver (DEC Catalog #H811A) | A-20557-29 | 29-18301 |
| Diagonal Cutters | Utica | 47-4 | 29-13460 |
| Diagonal Cutters | Utica | 466-4 (modified) | 29-19551 |
| Miniature Needle Nose Pliers | Utica | 23-4-1/2 | 29-13462 |

| | | | |
|---|---|---|---|
| Wire Strippers | Millers | 101S | 29-13467 |
| Solder Extractor | Solder Pullit | Standard | 29-13451 |
| Soldering Iron (30W) | Paragon | 615 | 29-13452 |
| Soldering Iron Tip | Paragon | 605 | 29-19333 |
| 16-Pin IC Clip | AP Incorporated | AP923700 | 29-10246 |
| 24-Pin IC Clip | AP Incorporated | AP923714 | 29-19556 |
| Module Extender Boards (3) | DEC | | W900 |

*Tektronix type 454 oscilloscope is adequate for most test procedures: type 465 or equivalent, may be required for some measurements.

# CHAPTER 2

# TROUBLESHOOTING AIDS

## 2.1 ERROR SERVICING AND STATUS REGISTERS

Error logging stores selected machine registers in the Constant Scratch Pad (CSP) for certain types of errors. The types of errors which will cause logging can be divided into two groups, Jam Errors and Service Errors.

The Jam Errors group is a result of a machine jam. The PDP-11/60 will jam, (i.e., the machine is forced to jump to a particular microcode routine), for a number of Unibus errors and other reasons internal to the processor. The following conditions will cause the current instruction to be aborted and cause logging:

1. Microbreak
2. Memory Parity Error
3. Cache Parity Error*
4. Slave Sync Timeout
5. Illegal Internal Address
6. Red Zone
7. Memory Management Abort
8. Odd Address
9. WCS Parity Error

The two remaining jam conditions, Power Up and Internal Address, will not cause an instruction to be aborted; nor will they cause logging to occur.

The Service Error group will cause logging to occur, but will not cause the current instruction to be aborted. The two Service Error conditions are:

1. Yellow Zone
2. Cache Parity Error*

*If the Parity Error Abort (CPE-bit-7) bit is set, in the Cache Control Register, then this error condition falls into the Jam Errors group and causes the current instruction to abort. If it's not set, then this error condition falls into the Service Errors group; and will not cause the current instruction to abort.

Error Servicing and Status registers which are logged are listed in Table 2-1, and described, along with their bit maps, in the following paragraphs.

**Table 2-1 Logging Registers**

| Location in CSP Register Name | Read/Write Code | Error Log Contents |
|---|---|---|
| CSP(Ø)<br>LOG JAM | 1ØØ/3ØØ | Jam register of status |
| CSP(1)<br>LOG SERVICE | 1Ø1/3Ø1 | Service register of status |
| CSP(2)<br>LOG PBA | 1Ø2/3Ø2 | Physical Bus Address Register (Bits 17, 16 located in Log Service Register) |
| CSP(3)<br>LOG CUA | 1Ø3/3Ø3 | Current Microword Address |
| CSP(4)<br>LOG FLAG/INTERRUPT | 1Ø4/3Ø4 | Flag Request Register Address (Status of Last Vector Serviced) |
| CSP(5)<br>LOG WHAMI | 1Ø5/3Ø5 | Logging instructions, internal flags Processor option status |

| | | |
|---|---|---|
| CSP(6)<br>LOG CACHE DATA | 106/306 | Logs Cache data being addressed at time of jam |
| CSP(7)<br>TAG/HIT | 107/307 | Logs Tag Field, Hit LOG Register and Valid Bit Status at time of jam |
| CSP(11)<br>LOG DS | 111/311 | Data stored; logs whatever was in the D register at the time of the jam |

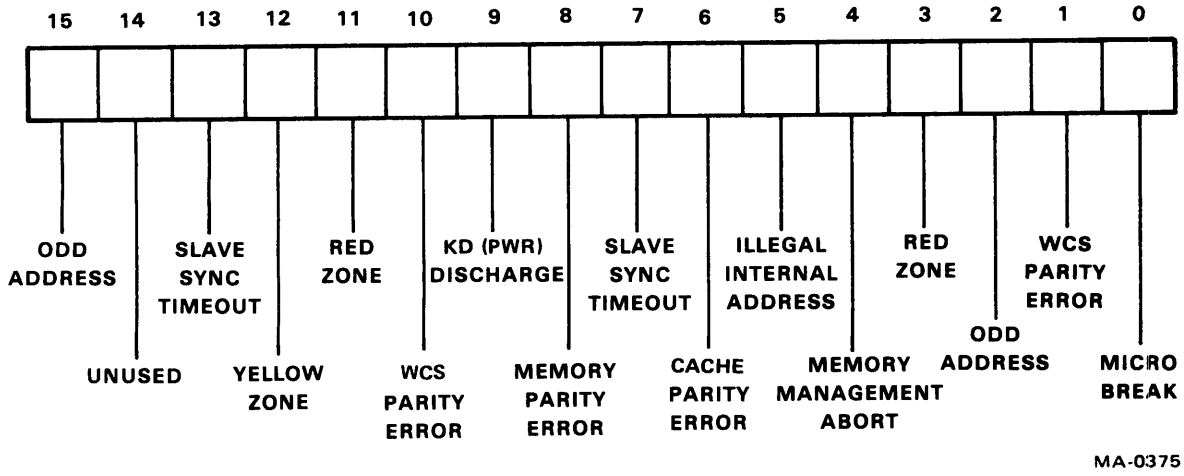



Figure 2-1 Log Jam Register Bit Map

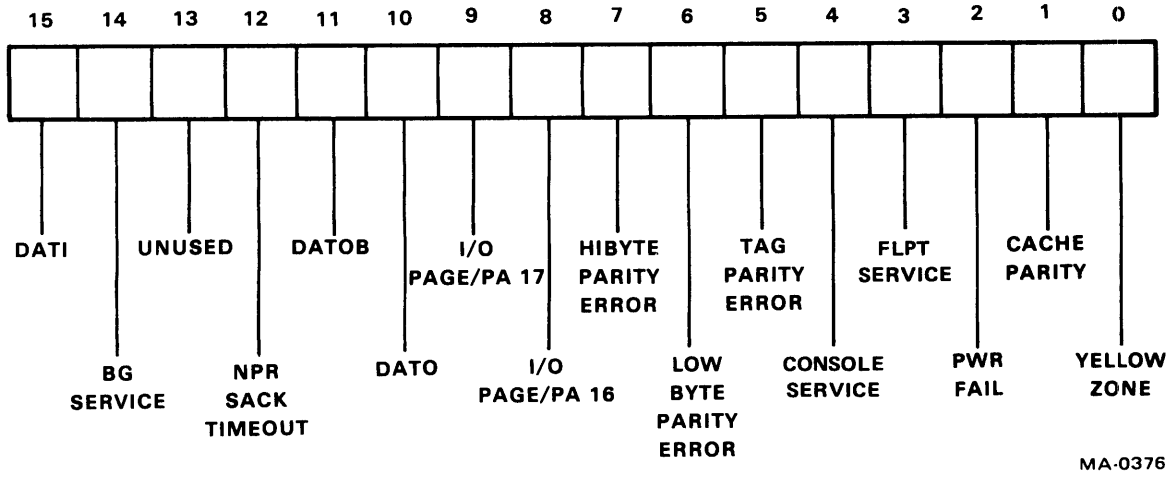



Figure 2-2 Log Service Register Bit Map

### 2.1.1 PDP-11/60 Error Logging Registers

The following descriptions apply to registers which are logged. These 16 registers, contained in the "C" Scratch Pad, are dual purpose registers. Their first purpose is that of storing special Floating Point Constants; their second purpose, which is our main concern, is for Logging Unibus or processor error information. Those bits which are not self-explanatory are described below their bit maps.

**2.1.1.1 LOG JAM (CSP0)** - The Log Jam register is used to reflect the type of Unibus or processor error that caused the PDP-11/60 to jam. Figure 2-1 shows the bit positions indicative of the jam conditions that may occur in the PDP-11/60.

Bit 9, KD(PWR) DISCHARGE, is looked at by the processor to see if the battery backup (used on systems with MOS memory) was good upon power up. (A "1" means that the battery is good.)

Bit 5, ILLEGAL INTERNAL ADDRESS, is set whenever a floating point instruction tries to address an internal address.

**2.1.1.2 LOG SERVICE (CSP1)** - Figure 2-2 shows the Service Error bit indicators.

**2.1.1.3 LOG PBA (CSP2)** - This register contains the Physical Bus Address at the time of the abort. The upper two bits of the PBA are found in the Log Service register. This register may not

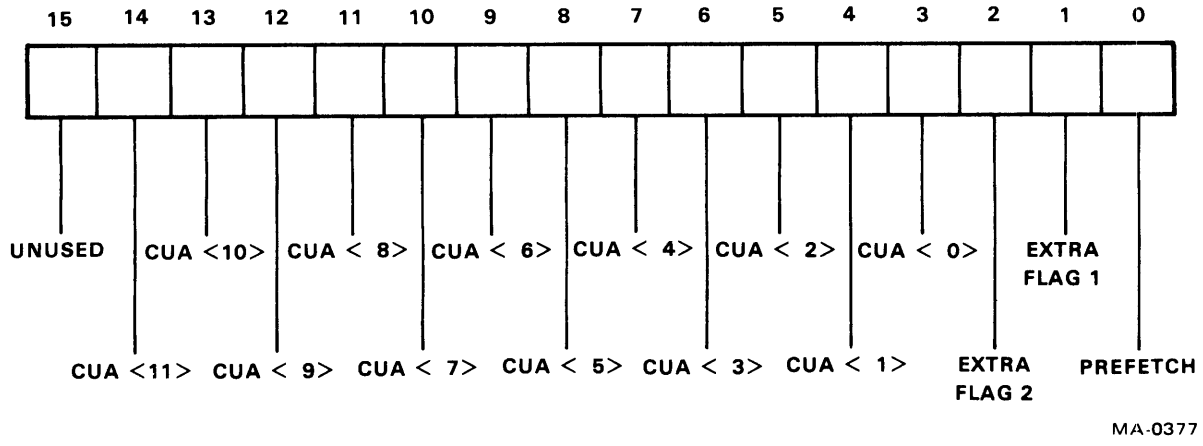



Figure 2-3 Log Current Micro-Address Register Bit Map

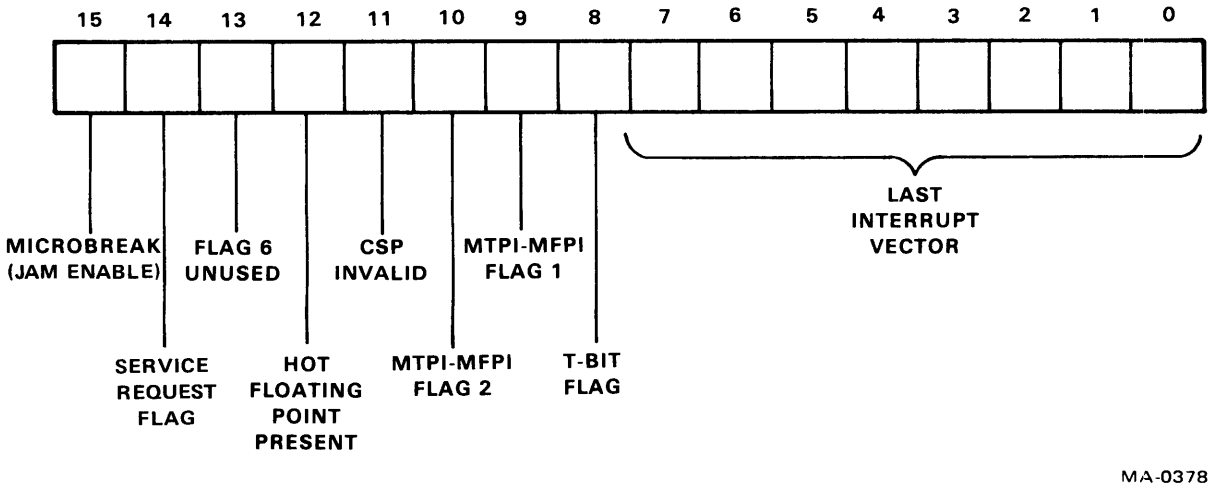



Figure 2-4 Log Flag/Interrupt Register Bit Map

reflect the error PBA if logging was caused by the Service Error group. Since the instruction is not aborted in this case, the PBA could change.

**2.1.1.4 LOG CUA (CSP3)** - This register contains the address of the microword which caused the jam. If a Service Error caused logging, this register will not contain meaningful data. Figure 2-3 shows the Log Current Micro-Address register's bit map.

Bit Ø (PREFETCH) when set, signifies the processor was executing a Fetch Overlap at the time of the log. Bit 1 (EXTRA FLAG 1) is used to control the signal: FLPT PROC H, which specifies which floating point processor is available. (A"1" means that FP11-E is present.)

Bit 2 (EXTRA FLAG 2) can be used by a microprogrammer as a branching bit.

**2.1.1.5 LOG FLAG/INTERRUPT (CSP4)** - Figure 2-4 shows where this register logs the contents of the Flag register and the Interrupt Vector at the time of the jam.

Bit 8 (T-BIT FLAG) when set, T-bit Traps are disabled.

Bits 9 and 1Ø (MTPI/MFPI FLAGS) - The Move to Previous Instruction and Move from Previous Instruction flags perform two functions:

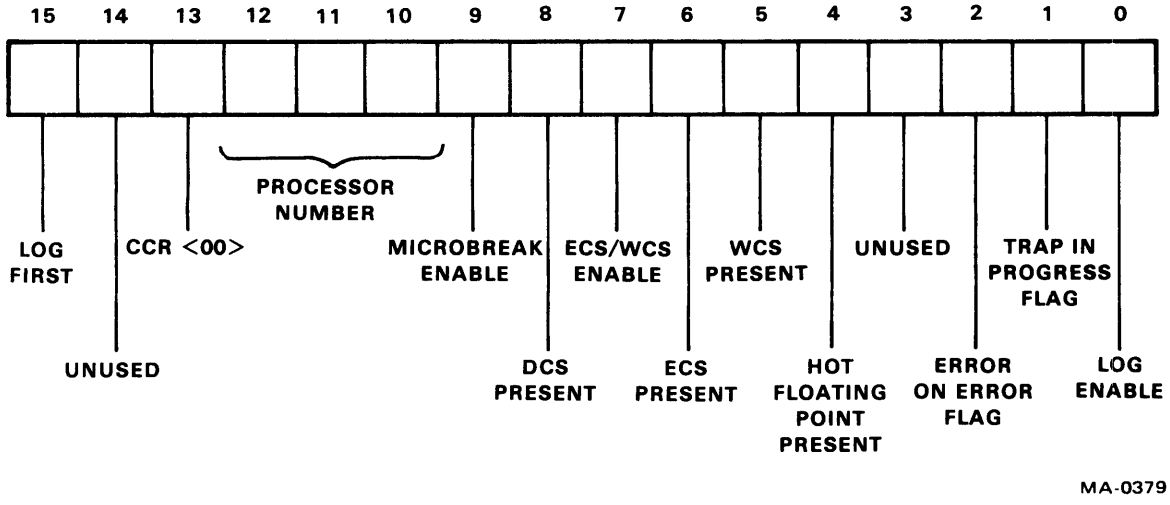



Figure 2-5 Log WHAMI Register Bit Map

1. MTPI-MFPI

| | FLAG 2 (bit-10) | FLAG 1 (bit-9) |
|---|---|---|
| MFPI | 1 | 0 |
| MTPI | 0 | 1 |

2. Control of the Double Precision Mode (FD) bit.

| | FLAG 2 (bit 10) | FLAG 1 (bit 9) |
|---|---|---|
| Pass FD Directly | 0 | 0 |
| Invert FD Bit | 1 | 0 |
| Force FD to 0 | 1 | 1 |

Bit 11 (CSP invalid), is set when the Floating Point constants, stored in the CSP, are invalid.

**2.1.1.6 LOG WHAMI (CSP5)** - The LOG WHAMI register contains administration information regarding the options that are in the machine, logging instructions and some internal flags. Figure 2-5 represents the LOG WHAMI registers bit map. Bit 15 (LOG FIRST) indicates, if set, only the first system/processor error will be logged, if Log Enable (bit 0) is also set.

Bit 13 (CCR<00>) logs the status of bit-0 from the Cache Control Register.

Bits 12, 11 and 10 (PROCESSOR NUMBER) are used to specify, to the microcode, which processor, in a multiprocessor environment, it is servicing. This is a hook for multiprocessing.

Bit 9 (MICROBREAK ENABLE), when set, it allows the machine to trap on a micromatch. For a more complete description see the micro-break register (Chapter 2.1.4).

Bit 7 (ECS/WCS ENABLE) is set by a microprogrammer to enable the WCS or ECS.

Bit 2 (ERROR ON ERROR FLAG) is set when an error trap causes another error; it will stay set until cleared by the MED code, see the section on the MED Instruction (Chapter 2.2).

Bit 0 (LOG ENABLE), if set, specifies that logging is unconditional; if it is clear logging occurs depending on the state of WHAMI <15>. Bit-0 is cleared the first time through the logging flow.

**2.1.1.7 LOG CACHE DATA (CSP6)** - This register logs the cache data that was being addressed at the time of the jam. If logging was caused by a Service Error then this data will not necessarily be relevant because the PBA may have changed since the time of the error.

**2.1.1.8 LOG TAG/HIT (CSP7)** - Figure 2-6 shows the LOG TAG/HIT

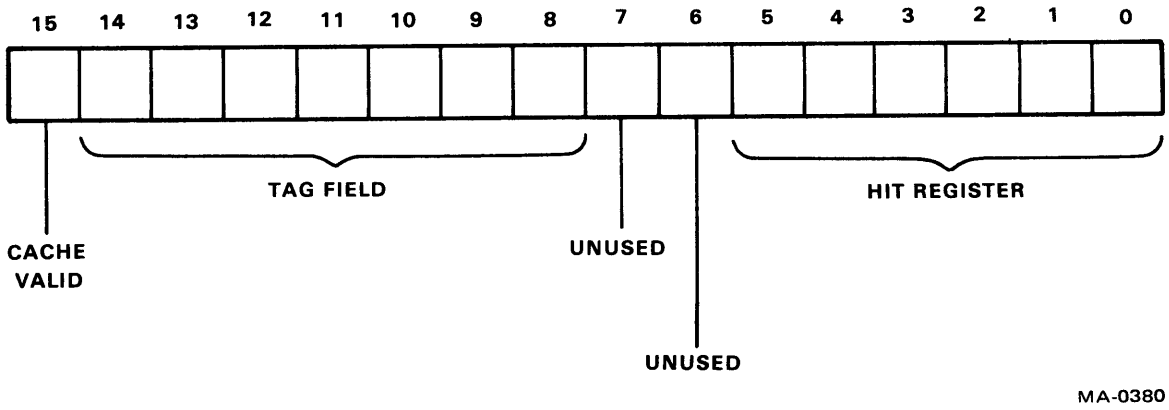



Figure 2-6 Log Tag/Hit Register Bit Map

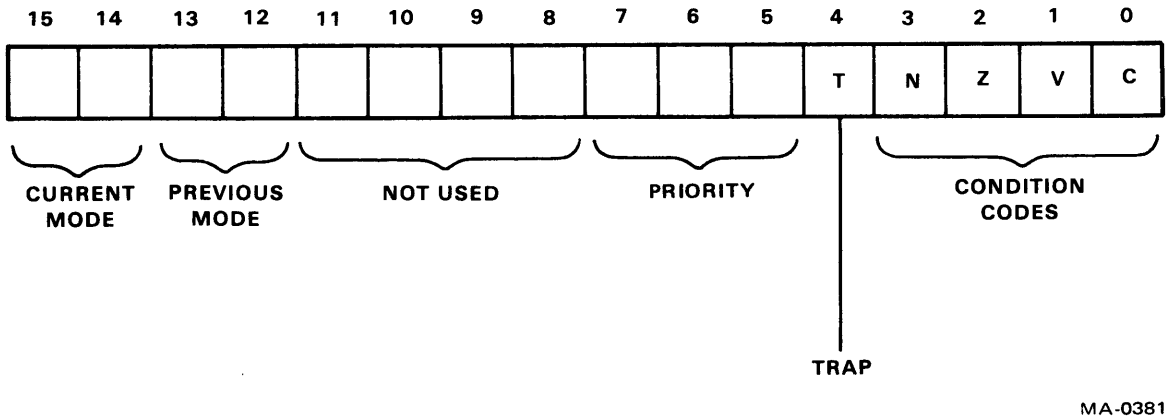



Figure 2-7 Processor Status Word Bit Map

register's bit map. This register logs the Tag Field and the contents of the Hit register at the time of a jam. The Tag field may not be relevant in a Service Error instance.

**LOG DS (CSP11)**

This register logs whatever was in the D register at the time of a jam. The D register is a register that holds data to be passed to memory; it also holds intermediate data as well as operands at different times during execution of an instruction.

### 2.1.2 PDP-11/60 Error and Status Registers

The following paragraphs contain descriptions of registers containing information on the status of the PDP-11/60. These registers have Unibus addresses and are used for control and diagnostic purposes.

**2.1.2.1 PROCESSOR STATUS WORD (777776)** - Figure 2-7 shows the bit map of the Processor Status Word (PSW) register. This register contains information on the current status of the PDP-11/60. It includes the current processor priority, current and previous modes, condition codes describing the results of the last instruction, and an indicator for detecting the execution of an instruction to be trapped during program debugging.

Bits 12-15 (PREVIOUS and CURRENT MODE)

These bits indicate the relocation and protection mode of the Memory Management unit at the present time (current) and prior to

the last mode (Trap or Interrupt) change (previous). For details on Memory Management operation refer to the PDP-11/60 Processor Handbook, Document Number EB06498.

Bit 5-7 (PRIORITY)

The central processor operates at any one of eight levels of priority, 0-7. The current priority is maintained in the Processor Status Word register bits 5-7.

Bit 4 (TRAP)

The trap bit (T) can be set or cleared under program control (vectoring instructions only). When set, a processor trap will occur through location 14 on completion of an instruction and a new PSW will be loaded. This bit is especially useful for debugging programs as it provides an efficient method of installing breakpoints.

Bits 0-3 (CONDITION CODES)

The Condition codes contain information on the result of the last CPU operation. The bits are set as follows:

0(C) = 1, if the operation resulted in a carry from the MSB.

1(V) = 1, if the operation resulted in an arithmetic overflow.

2(Z) = 1, if the result was zero.

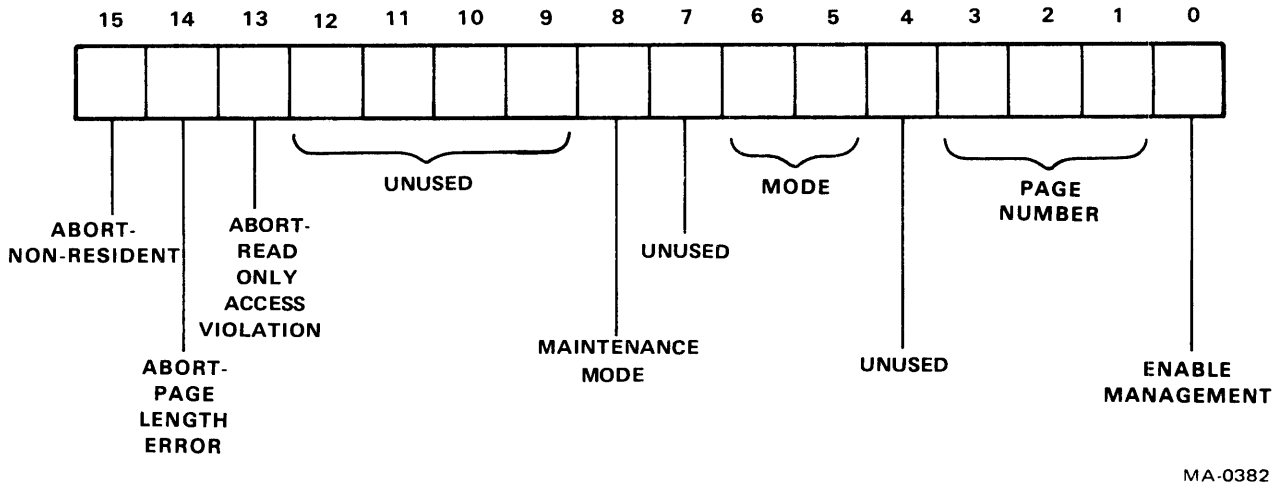



Figure 2-8 Memory Management Register 0 Bit Map

3(N) = 1, if the result was negative.

**2.1.2.2 MEMORY MANAGEMENT STATUS REGISTERS** - Aborts generated by the protection hardware are vectored through kernal virtual location 250. Status Registers #0 and #2 are used to determine why the abort occurred.

**MEMORY MANAGEMENT REGISTER 0 (MMR0) 777572**

MMR0 contains abort error flags, memory management enable, plus other essential information required by an operating system to recover from an abort or service a memory management trap. The MMR0 bit map is shown in Figure 2-8.

Bits 13-15

These bits are the Abort Flags. They may be considered to be a "priority queue" in that flags to the right are less significant and should be ignored. For example, a Non-Resident Abort service routine would ignore Page Length and Access control flags. A Page Length Abort service routine would ignore an Access control fault.

**NOTE**

**Bits 15, 14, or 13, when set (Abort conditions) cause the logic to freeze the contents of MMR0 bits 1 to 6 and status register MMR2. This is done to facilitate recovery from the abort.**

Bit 15 is set when an attempt is made to access a page with an access control field key equal to Ø or 4. Bit 14 is when an attempt is made to access a location in a page with a block number (virtual address bits 6-12) that is outside the area authorized by the Page Length Field of Page Descriptor Register for that page.

Bit 13 is set when an attempt is made to write in a "Read Only" page having an access key of 2.

Bit 8 specifies maintenance use of the Memory Management unit. For the instructions used in the initial diagnostic program, bit 8 is set so that only the final destination reference is relocated. It is useful to prove the capability of relocating addresses.

Bits 5 and 6 indicate the CPU mode (User or Kernel) associated with the page causing the abort. (Kernel = ØØ, User = 11.)

Bits 1-3 contain the page number of reference. Pages, like Blocks, are numbered from Ø upwards. The page number bit is used by the error recovery routine to identify the page being accessed if an abort occurs.

Bit Ø is the Enable bit. When it is set to 1, all addresses are relocated and protected by the Memory Management unit. When it is set to Ø, the Memory Management unit is disabled and addresses are neither relocated nor protected.

**MEMORY MANAGEMENT REGISTER 2 (MMR2) 777576**

MMR2 is loaded with the 16-bit Virtual Address at the beginning of each instruction fetch but is not updated if the instruction fetch fails. MMR2 is read only; a write attempt will not modify its contents. Upon an abort, the result of MMRØ bits 15, 14, or 13 being set, the logic will freeze the contents of MMR2 until the MMRØ abort flags are cleared.

**2.1.2.3 STACK LIMIT REGISTER (777774)** - The contents of the Stack Limit Register are compared to the stack address to determine if a violation has occurred (although memory references that do not alter memory are always allowed).

Bits 15 through 8 contain the stack limit information. These bits are cleared by System Reset, Console Start, or the Reset instruction. The lower 8 bits are not used. Bit 8 corresponds to a value of (400) or (256) . For a more detailed description of the Programmable Stack Limit, refer to the PDP-11/60 Processor Handbook.

**2.1.2.4 CPU ERROR REGISTER (777766)** - This register, (bit map shown in Figure 2-9), is a word accessible only. The CPU Error register was included in the PDP-11/60 so PDP-11/70 software could be used with minimal change.

**2.1.2.5 Bit Ø (ILLEGAL INTERNAL ADDRESS)** - This bit is set when a

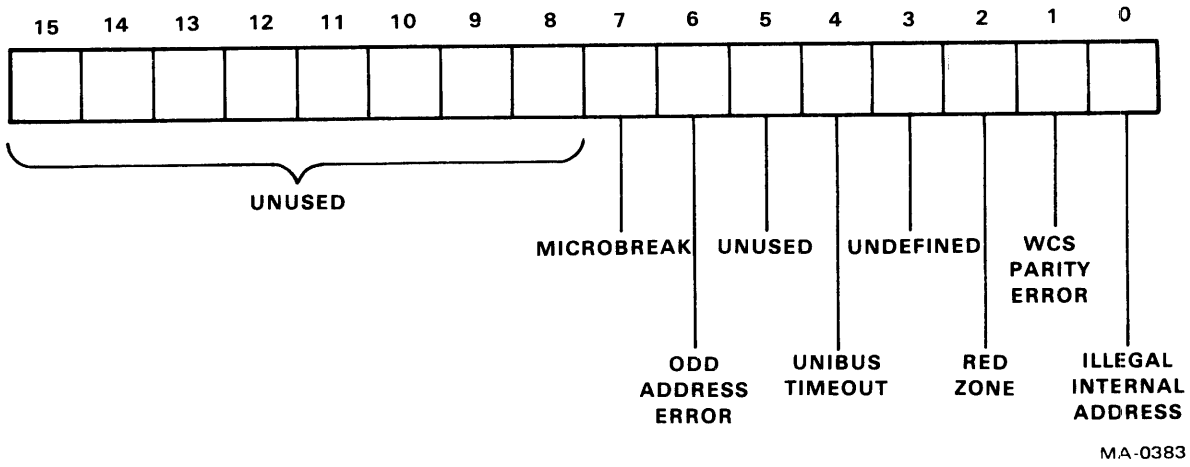



Figure 2-9 CPU Error Register Bit Map

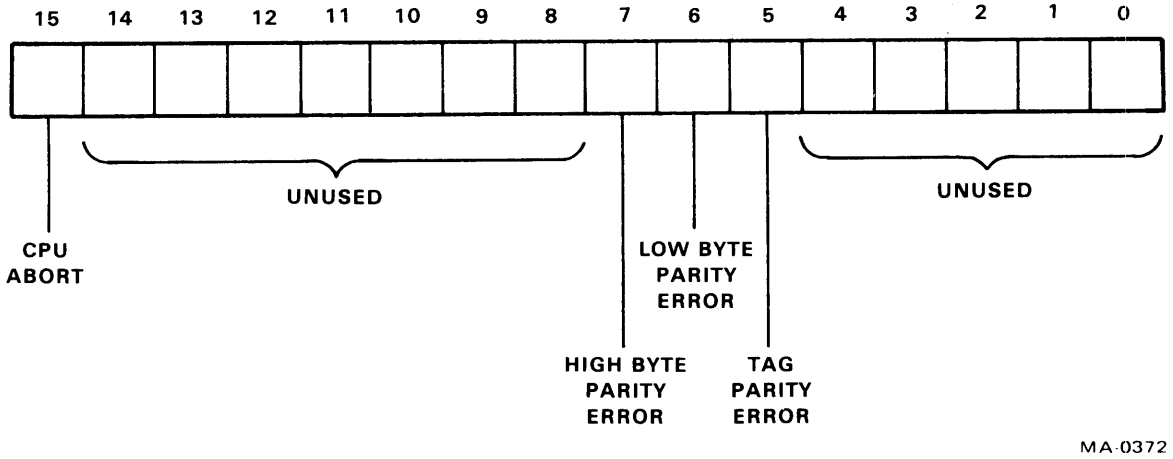



Figure 2-10 Memory Error Register Bit Map

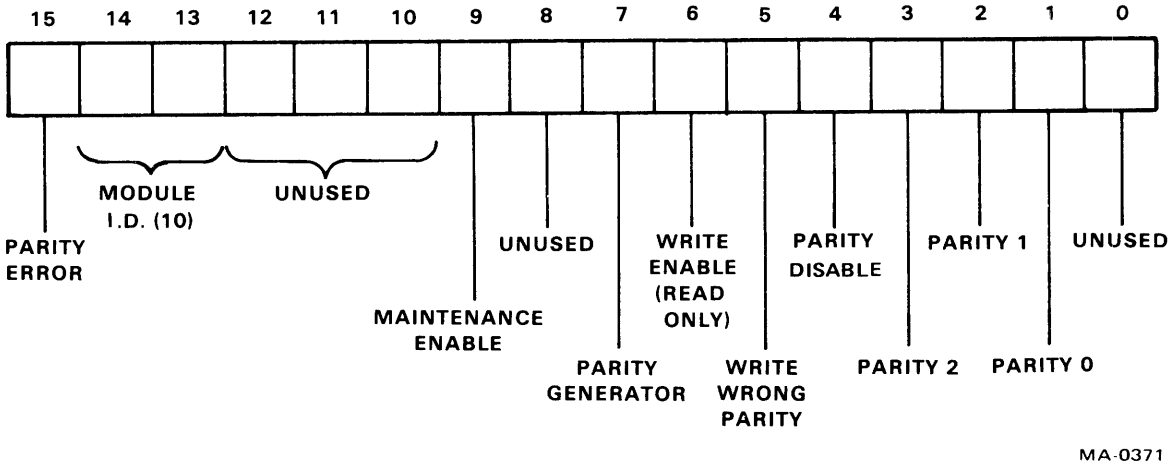



Figure 2-11 WCS Status Register Bit Map

Floating Point instruction tries to address an internal address. The remaining errors listed in the bit map are all self-explanatory.

**2.1.2.6 MEMORY ERROR REGISTER (777744)** - Figure 2-10 shows the bit map of the Memory Error Register. The parity error bits are correct only if there has been an abort, i.e., the CPE Abort bit of the Cache Control Register is set. If a cache parity error is handled by the service flow the CPU Abort bit will not be set and the parity bits will be forced to ones.

**2.1.2.7 CACHE HIT REGISTER (777752)** - This register indicates whether the six most recent references by the processor were hits or misses. A one (1) indicates a read hit; a zero (0) 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.

**2.1.2.8 SWITCH REGISTER (777570)** - The Switch Register is stored in one of the scratch pads. It is initialized to a zero upon power up. This register can be loaded from the console while a program is running. A DATO to this register will load the Display Register.

**2.1.2.9 WCS (Writeable Control Store) Status Register (777740)** - Figure 2-11 shows a bit map of the WCS Status Register. Refer to the PDP-11/60 Processor Handbook for a simplified description of

the PDP-11/60's microprogramming capabilities.

**2.1.2.10 MICROBREAK REGISTER (777770-Write Only)** - The Microbreak Register is a twelve bit register whose contents are compared with the next microaddress; when there is a match. The machine will respond in a number of different ways depending on how the machine was set up and what options are available. For set up conditions and a more detailed description of the Microbreak Register see Chapter 2.1.4.

### 2.1.3 PDP-11/60 Control Registers

The following paragraphs contain descriptions on registers used to control the accessing of data in Kernel and User modes, allowing system recovery from errors, memory protection, program relocation and description information.

**2.1.3.1 CACHE CONTROL REGISTER (777746)** - The PDP-11/60 has the capability to recover from cache parity errors, and to allow operation in a degraded mode if a section of the memory system is not operating properly. If data found in a location in cache does not have correct parity, a backing store reference can automatically occur to allow program execution to proceed. If a number of locations in cache fail, it is possible to turn off part or all of cache using the Force Miss bits, (see Figure 2-12 and the note after the paragraph describing the Force Miss Bits) of the Cache Control Register (CCR). Refer to the PDP-11/60 Processor Handbook for a more detailed description of the Cache Control Register.

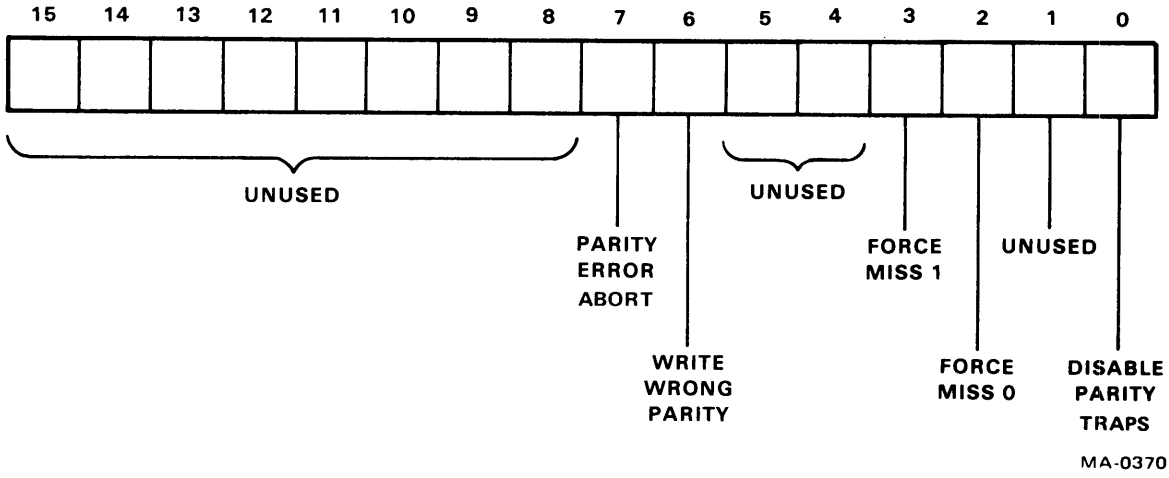



Figure 2-12 Cache Control Register Bit Map

Bit 7 (PARITY ERROR ABORT)

This bit is cleared on power up. It is set only during maintenance diagnostics and will cause an abort when a cache parity error occurs.

Bit 6 (WRITE WRONG PARITY)

This bit is cleared on power up. It is used during maintenance diagnostics and, if set, will write wrong parity in the tag, high

byte and low byte when cache is updated.

Bits 2 & 3 (FORCE MISS Ø & 1)
Setting these bits forces misses on reads to the cache and on attempts to invalidate the cache on NPR DATO* references. Bit 3 forces misses on words 512 to 1Ø23. Bit 2 forces misses on words Ø to 511. Setting both bits forces all cycles to main memory.

**NOTE**

**Disabling either half of cache should only be used as a diagnostic tool. The PDP-11/6Ø will not run properly with either half of cache disabled.**

Bit Ø (DISABLE TRAPS)
Set by the cache parity error handler when it is desired to disable traps occuring as a result of nonfatal cache errors.

**2.1.3.2 ACTIVE PAGE REGISTERS** - The Memory Management Unit provides two sets of eight Active Page Registers (APR). Each APR consists of a Page Address Register (PAR) and a Page Descriptor Register (PDR). These registers are always used as a pair and contain all the information required to locate and describe the current active pages for each mode of operation. One PAR/PDR set is used in Kernel mode and the other is used in User mode. The current mode bits (and in some cases, the previous mode bits) of the Processor Status Word (see Chapter 2.1.2.1) determine which

set will be referenced for each memory access. A program operating in one mode cannot use the PAR/PDR sets of the other mode to access memory. This is one of the protection features employed in the 11/60 for a time-shared multiprogramming system.

*Setting either bit will prevent invalidate on NPR DATO's for ALL cache locations.

A specific processor I/O address is assigned to each PAR and PDR of each set. Refer to Table 2-3 for the PAR/PDR address assignments. The following paragraphs indicate the ranges of these addresses for both Kernel and User modes, along with breakdowns of the registers bit maps.

PAR (Kernel, 772340-56)

PAR (User, 777640-56)

**Table 2-3 PAR/PDR Address Assignments**

| Kernel Active Page Registers | | | User Active Page Registers | | |
|---|---|---|---|---|---|
| **No.** | **PAR** | **PDR** | **No.** | **PAR** | **PDR** |
| 0 | 772340 | 772300 | 0 | 777640 | 777600 |
| 1 | 772342 | 772302 | 1 | 777642 | 777602 |
| 2 | 772344 | 772304 | 2 | 777644 | 777604 |
| 3 | 772346 | 772306 | 3 | 777646 | 777606 |
| 4 | 772350 | 772310 | 4 | 777650 | 777610 |

| | | | | | |
|---|---|---|---|---|---|
| 5 | 772352 | 772312 | 5 | 777652 | 777612 |
| 6 | 772354 | 772314 | 6 | 777654 | 777614 |
| 7 | 772356 | 772316 | 7 | 777656 | 777616 |

The Page Address Register, shown in Figure 2-13, contains the 12-bit Page Address Field (PAF) that specifies the base address of the page.

The PAR may be alternatively thought of as a relocation constant, or as a base register containing a base address. Either interpretation indicates the basic function of the PAR in the relocation scheme.

PDR (Kernel, 772300-16)

PDR (User, 777600-16)

The Page Descriptor Register, shown in Figure 2-14, contains information relative to page expansion, page length, and access control. The following paragraphs give brief descriptions of the bit functions of the PDR; for more details, refer to the 11/60 Processor Handbook.

Bits 2 & 1 (ACCESS CONTROL FIELD)

This 2 bit field of the PDR describes the access rights to this particular page. The access codes, shown in Table 2-4, or "Keys" specify the manner in which a page may be accessed and whether or not a given access should result in an abort of the current operation.

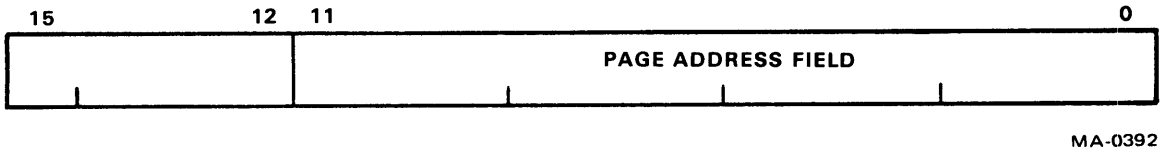



Figure 2-13 Page Address Register Bit Map

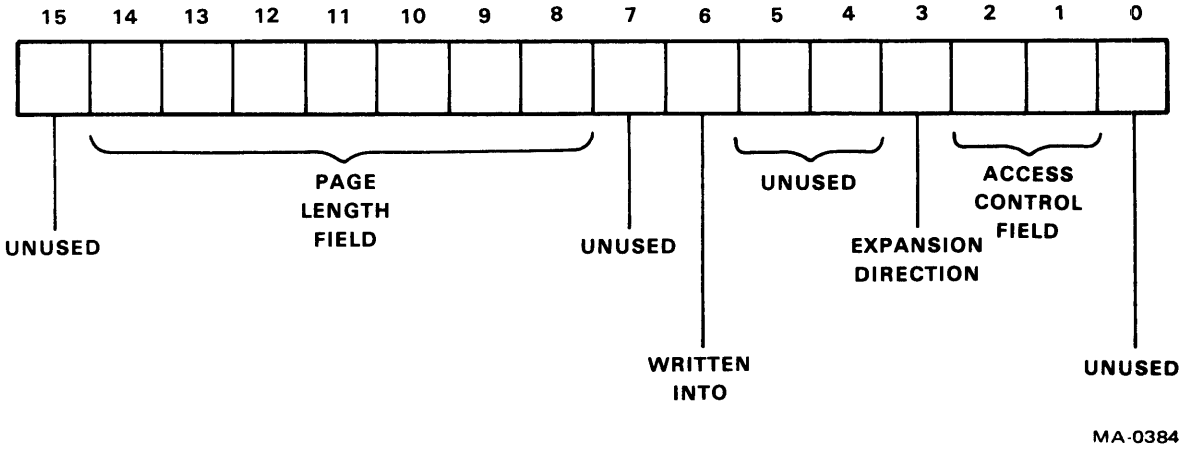



Figure 2-14 Page Descriptor Register Bit Map

**Table 2-4 Access Control Field Keys**

| AFC | Key | Description | Function |
|---|---|---|---|
| ØØ | Ø | Non-resident | Abort any attempt to access this non-resident page. |
| Ø1 | 2 | Resident read-only | Abort any attempt to write into this page. |
| 1Ø | 4 | (unused) | Abort all Accesses. |
| 11 | 6 | Resident read/write | Read or Write Allowed. No trap or abort occurs. |

Bit 3 (EXPANSION DIRECTION)

The ED indicates the authorized direction in which the page can expand. A logic Ø indicates the page can expand upward from relative Ø. A logic 1 indicates the page can expand downward toward relative Ø.

Bit 6 (WRITTEN INTO)

The W bit indicates whether the page has been written into since it was loaded into memory. A logic 1, in bit positon 6, indicates that the page has been written into.

Bits 14-8 (PAGE LENGTH FIELD)

The 7-bit PLF specifies the authorized length of the page, in 32 word blocks. The PLF holds block numbers from Ø to 177 ; allowing any page length from Ø to 128 blocks. The PLF is written in the PDR under program control.

### 2.1.4 Microaddress Match and Microbreak Register [K311]

The Microaddress Match and Microbreak Register provides a customer engineer with three different ways of examining any single microstate.

The heart of this feature is a twelve bit microaddress comparator. The comparator compares each microaddress to the contents of a microbreak register and indicates when they are equal. The microbreak register is loaded from the console.

Loading control is provided by UCON PROC WR and UCON Ø9. The three features are:

1. Dynamic Testing: Each time a microaddress match is obtained, a synch pulse is generated at BØ3M2. This pulse allows the dynamic scoping of an internal signal on any selected microword.

2. Static Testing: The processor clock can be stopped* in the second microstate following the previously loaded microword address when BØ3M2 is jumpered to EØ6K1 on the Timing Module. This feature allows microstate conditions to be accumulated at normal processor speed and then to be stopped just before they reach a suspected area of faulty microcode. Once stopped, the processor can be single stepped for closer examination.

   To single microstep, flip the bottom switch on the Timing Module (M7876), so that it points downward. Then flip the top switch up, then down, and the processor will advance one microword. To allow the processor to free run again, flip the bottom switch up, and microstep twice (flip the top switch up and down twice). This action will cause the clock to free run again.

**NOTE**

**In Single Micro Step, the clock is**

**stopped immediately after the loading of the next microword, (and before the last microword was used).**

3. Register Dump: (u break Jam): when bit 8 in the Flag register, (Refer to Flows box #546) is set upon a micro-match the machine will log the various registers which are normally logged for a Log Jam or Log Service. When this is completed the machine will halt. Flag bit 8 can be set via console or the MED instruction.

4. If Flag bit 8 or WHAMI bit 9 are set the machine will trap to four upon a micro-match. Logging occurs in this instance also. WHAMI bit 9 can be set via console or the MED instruction.

## 2.2 MED INSTRUCTION AND ITS USE

MED is an acronym for Maintenance, Examine and Deposit. It allows the user to read and write many of the internal registers. This instruction is primarily a diagnostic tool.

The MED instruction is a two word instruction; the first word being the opcode for the MED instruction (076600), the second word being the MED function code for the specific operation desired. The MED instruction will only work in Kernel mode, an attempt to execute a MED instruction in User mode will cause a trap to 10.

To use the MED instruction the MED opcode should precede the MED function code in memory. If the MED function code specifies a read, the data will be stored in RØ; if it specifies a write the data in RØ will be written into the desired register.

Reading and writing internal registers can also be performed directly from the console via the MAINT key. The read/write function codes for registers used for diagnostic purposes are listed in Table 2-1. The following procedures describe how to read and write internal registers from the console:

READ Internal Register Procedure

1. Load the Temporary Switch Register with the read function code of the register that is desired to be read.

2. Press the (L)SWR keypad switch while holding in the CNTRL key. This transfers the contents of the Temporary Switch Register to the Console Switch Register.

3. Press the MAINT key while holding in the CNTRL key. The console display will display the contents of the register specified by the function code in step 1.

WRITE Internal Register Procedure

1. Load the Temporary Switch Register with the write func-

tion code of the register that it is desired to write.

2. Press the (L)SWR key while holding in the CNTRL key. This transfers the contents of the Temporary Switch Register to the Console Switch Register.

3. Load the Temporary Switch Register with the data to be written by pressing the applicable numeric keys.

4. Press the MAINT key while holding in the CNTRL key. The console display will display the data that has been written into the specified register.

## 2.3 TOGGLE-IN ROUTINES

A few ideas that should aid in debugging failures in the PDP-11/60 processor are presented here. The troubleshooter of course should attempt to run the standard PDP-11/60 diagnostics before attempting to use the methods that are outlined here. If these will not load or run properly, then the user should try to load and run them with the cache and/or the bootstrap loader & terminator modules disabled; that is:

1. Cache Sweep Bypass

    a. Installing a jumper from E02P2 (Bus DIN, bit 14) to ground. This jumper must be taken off after starting to prevent damage to "BUS DIN" (TRISTATE). The microcode will go into a safety loop until the jumper is taken off. The Slide switch must be in the HALT position.

    b. Setting bits 3 & 2 (Force Miss 1 & 0) of the Cache Control Register (UBA-777746) forces misses on reads to the cache, and a backing store reference automatically occurs.

    c. When single stepping on power-up.

2. Bootstrap/Terminator Bypass

a. The toggle in routines, shown here, to boot the RK06 and RK05 disk drives will bypass the M9301-YH Bootstrap/Terminator Module's diagnostics; therefore the microswitches that determine the actions of the PDP-11/60 upon Power-up and Bootstrap operations do not have to be alterred. Refer to the PDP-11/60 Installation and Operation Manual for a detailed description of the M9301-YH microswitch settings.

### 2.3.1 RK05 Manual Boot

If the troubleshooter cannot boot the RK05 in the conventional manner, and the console is operational, toggle in the following routine:

| Address | Data |
|---|---|
| 001000 | 012737 |
| 001002 | 000005 |
| 001004 | 777404 |
| 001006 | 000001 |

1. Load address 001000.
2. Press START while holding in the CNTRL key.
3. Wait 5 seconds.
4. Press the HALT key.
5. Load address 000000.
6. Press START while holding the CNTRL key.

### 2.3.2 RK06 Manual Boot

If the troubleshooter cannot boot the RK06 in the conventional manner and the console is operational, toggle in the following routine:

| Address | Data |
|---|---|
| 001000 | 012737 |
| 001002 | 000003 |
| 001004 | 777440 |
| 001006 | 132737 |
| 001010 | 000200 |
| 001012 | 177440 |
| 001014 | 001774 |
| 001016 | 012737 |
| 001020 | 000021 |
| 001022 | 177440 |
| 001024 | 000001 |

1. Load address 001000.
2. Press START while holding in the CNTRL key.
3. Wait 1 second.
4. Press the HALT key.
5. Load address 000000.
6. Press START while holding in the CNTRL key.

### 2.3.3 Branch Self

This routine is an easy check of CPU. It exercises approximately 100 microwords and some CPU logic. If the console is operational, toggle in the following routine:

| Address | Data |
|---|---|
| 000200 | 000240 |
| 000202 | 000776 |
| 000204 | 000000 |

1. Load Address 000200.
2. Press START while holding in the CNTRL key.

This routine can be observed through the display by single stepping the instruction. This is accomplished by pressing the HALT/SI key switch (stopping the instruction), and the pressing of the HALT/SI key switch again will initiate a single instruction step, with the program counter pointing at one of the two instruction addresses. This operation, as the section heading (Branch Self) implies, will occur as often as the HALT/SI key switch is pressed.

### 2.3.4 Console Echo Routine

This routine is a simple check of the DL11-W Module and its input/output lines. If the console is operational, toggle in the following routine:

| Address | Data | Description |
|---|---|---|
| 000200 | 105737 | |
| 000202 | 177560 | wait for keyboard flag |
| 000204 | 100375 | |
| 000206 | 113737 | |
| 000210 | 177562 | put keyboard data buffer to |
| 000212 | 177566 | printer data buffer |
| 000214 | 105737 | |
| 000216 | 177564 | wait for printer done |
| 000220 | 100375 | |
| 000222 | 000766 | branch ADDR |

1. Load address 000200.
2. Press START key switch while holding in the CNTRL key.
3. Type any key on the LA36 terminal's keyboard.

If character key was pressed, it should print out. If function key was pressed, the stated function should have been performed.

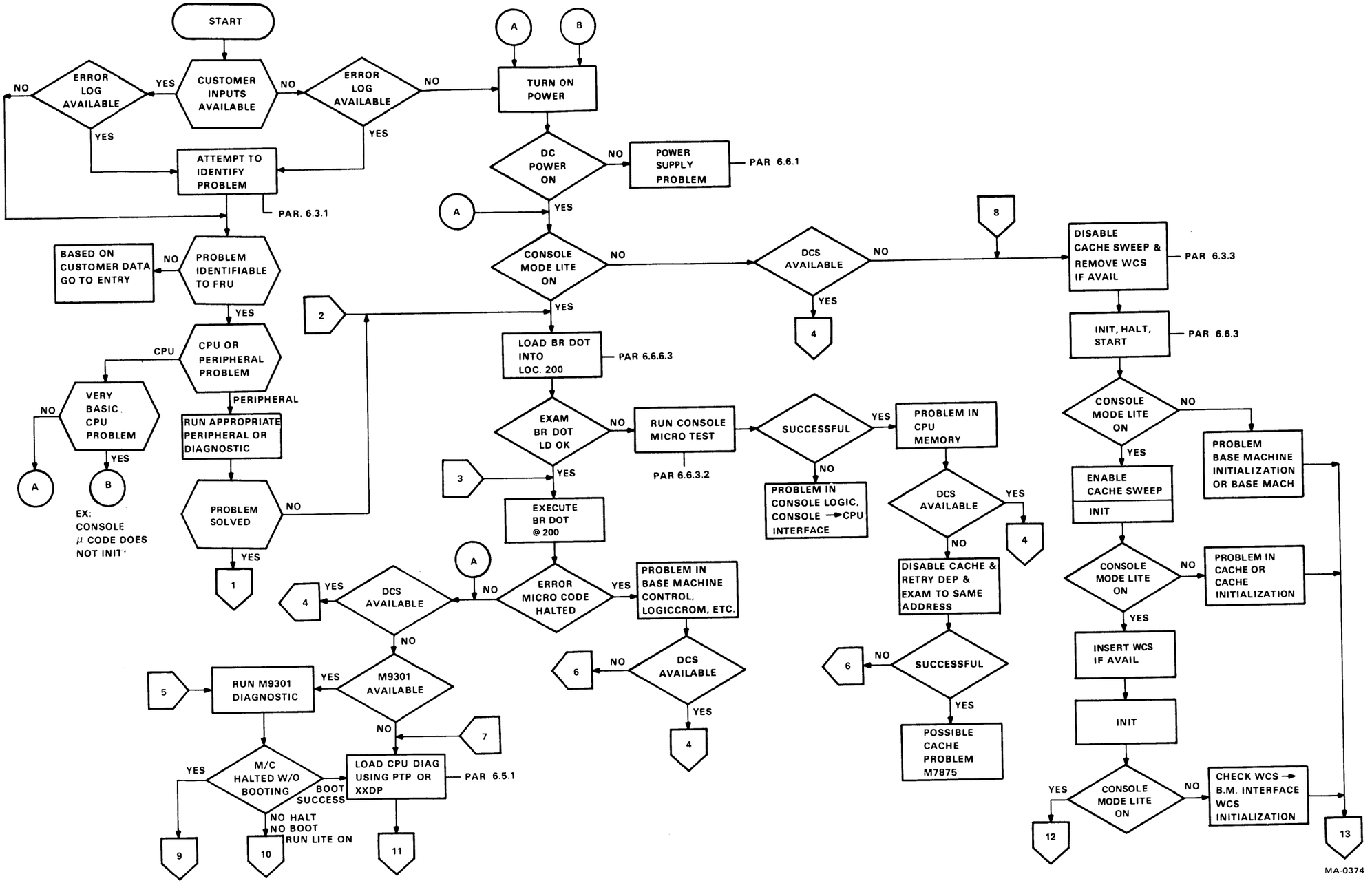



Figure 3-1 Guidelines For Troubleshooting (Sheet 1 of 3)

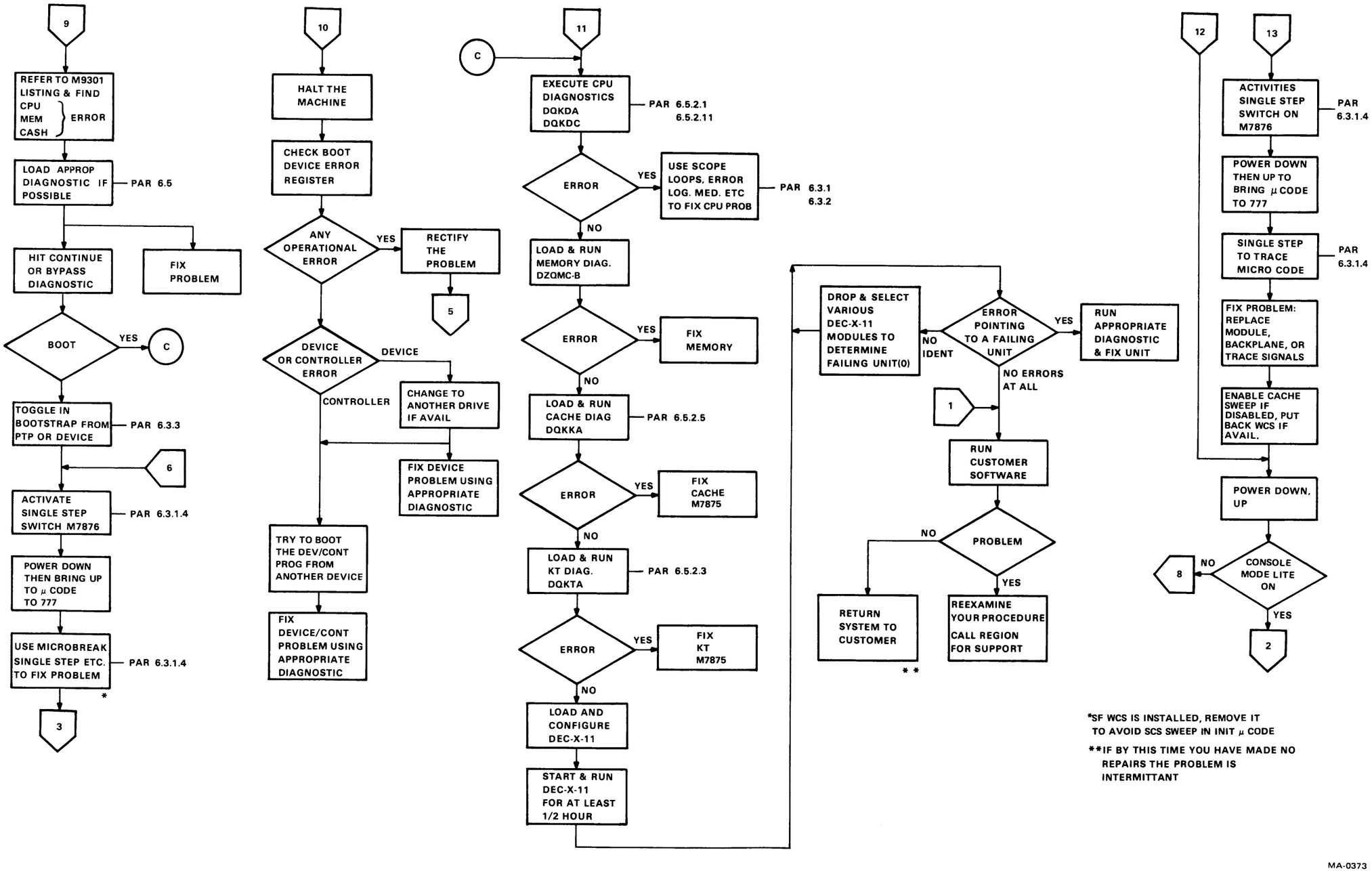



Figure 3-1 Guidelines For Troubleshooting (Sheet 2 of 3)

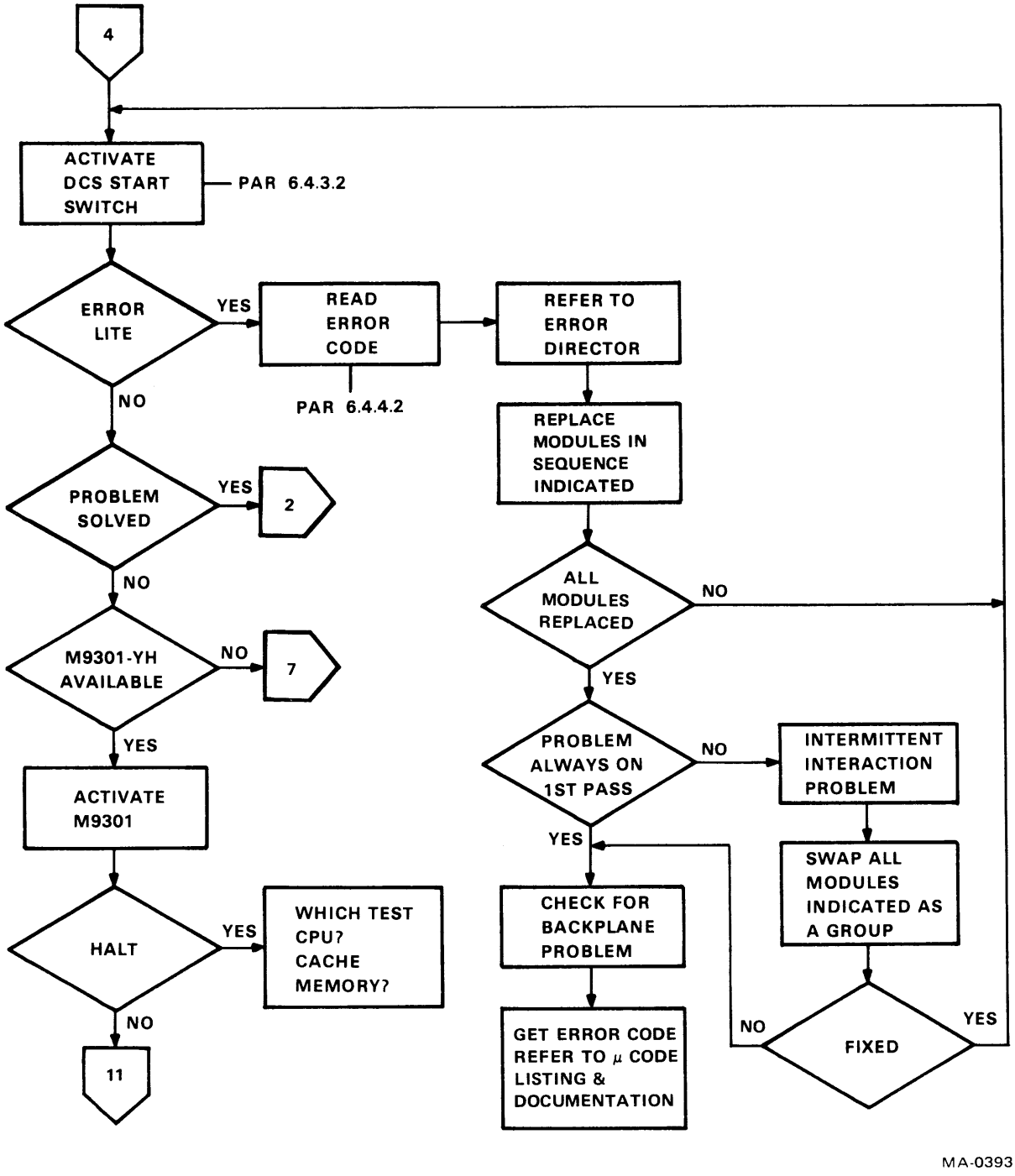



Figure 3-1 Guidelines For Troubleshooting (Sheet 3 of 3)

# CHAPTER 3

# COLD START TROUBLESHOOTING

## 3.1 INTRODUCTION

When a system is inoperative, the first things that should be checked and repaired if necessary include:

1. AC and dc power and cabling
2. Signal cabling
3. Console functions

The flowchart shown on Figure 3-1 suggests some steps that could be taken to bring a dead machine to a point at which the problem can be identified and fixed with a field replaceable unit, (FRU).

### 3.1.1 AC and DC Power

AC and dc power maintenance and troubleshooting are explained in the PDP-11/60 Cabinet and Power Supply Manual.

### 3.1.2 Cabling

Unibus and power (ac and dc) cabling is also described in the PDP-11/60 Cabinet and Power Supply Manual.

### 3.1.3 Console Operation

**3.1.3.1 Initialization to Console Loop --** Table 3-1 is a list of micro-addresses that the CPU will go through to get to the Console Loop from INIT (HALT and START switches). This list is for ROM level #50. To bypass Cache sweep refer to Chapter 2.3. The following notes will help you to determine where you are during the CPU's cycling to the Console Loop.

Table 3-1 PDP-11/60 Micro-Addresses from INIT to Console Loop

| | | | | |
|---|---|---|---|---|
| 777 | 3466 | 3557 | 1265 | 1520 |
| 1020*Note 1 | 3453 | 3524 | 1254 | 1522 |
| 1024 | 3467 | 3525 | 1255 | 1524 |
| 1025 | 3076 | 3547 | 1256 | 1526 |
| 1037 | 3424 | 3434 | 1257 | 1534 |
| 1121 | 3117 | 3327 | 1266 | 1536 |
| 1124 | 3511 | 3435 | 1270 | 1544 |
| 1126 | 3474 | 3440 | 1260 | 1546 |
| 1130 | 3116 | 3441 | 1262 | 1550 |
| 1132 | 3425 | 3442 | 1271 | 1551 |
| 3007 | 3137 | 3257 | 1272 | 1463 |
| 3615 | 3136 | 3256 | 1273 | 1552 |
| 3045 | 3037 | 3277 | 1274 | 1553 |
| 3653 | 3035 | 3526 | 1275 | 1554 |
| 3046 | 3157 | 3275 | 1003 | 1464 |
| 3013 | 3153 | 3313 | 1276 | 1465 |
| 3420 | 3426 | 3357 | 1300 | 1470 |
| 3412 | 3427 | 3530 | 1032 | 1472 |
| 3421 | 3177 | 3641*Note 3 | 1344 | 1471 |
| 3422 | 3175 | 3532 | 1345 | 1611 |
| 3017 | 3213 | 3643 | 1346 | 1473 |
| 3016 | 3430 | 3347 | 1350 | 1474 |
| 3423 | 3537 | 3443 | 1351 | 1007 |
| 3077 | 3475 | 3677 | 1555 | 1475*Note 4 |
| 3444 | 3504 | 1056 | 1556 | 1500 |

| | | | | |
|---|---|---|---|---|
| 3445 | 3505 | 1060 | 1557 | 1502 |
| 3446 | 3510 | 1061 | 1214 | 1504 |
| 3447 | 3512 | 1064 | 1460 | 1062 |
| 3450 | 3514 | 1065 | 1461 | |
| 3452 | 3515 | 1070 | 1462 | |
| 3460 | 3516 | 1477 | 1505 | |
| 3470-* | 3517 | 1655 | 1506 | |
| 3471-** | | | | |
| 3461 | 3071 | 1040 | 1507 | |
| 3462 | 3520 | 1000 | 1514 | |
| 3463 | 3522 | 1004 | 1515 | |
| 3464 | 3527 | 1664 | 1516 | |
| 3465 | 3431 | 1263 | 1517 | |
| | | 1264 | | |

*No op. C.S.

**DCS/WCS/ECS Installed

1. Holding in the HALT, START keys will cause Ø777 to be shown in the Micro-word Module's NUA lights. Releasing the keys, even though the CPU is in single cycle mode will cause the CPU to go to NUA = 1Ø24; this is because the maintenance mode latch is reset by the unit and it takes two microcycles before it will be set again to stop the clock.

2. Jumper E Ø2P2 must be off now to get out of the loop. (3615 to 1121).

3. Loop is 256 single cycles.

4. Console Loop.

Section 3 of the PDP-11/60 Installation and Operation Manual contains a detailed operational description of the KY11-P console.

**3.1.3.2 Console Microtest** -- This feature is permanently resident in the base machine ROM and allows checking and repair of the console, even with the base machine down. In particular it can be used to test and verify that console switches are generating correct codes, (see Table 3-2) and the display is generating correct characters. It is meant to be used only after it has been verified that initialization is working. This diagnostic will not operate if initialization is not functioning.

When invoked, this diagnostic continuously moves the contents of the Temporary Switch Register to the display. Whenever a key switch is pressed, the seven digit code generated by the key pad is read. This code is then linked together with a zero in bit <5> and left shifted into the Temporary Switch Register.

The diagnostic then goes back to continuously displaying the Temporary Switch Register. The keypad codes then have the appearance of drifting through the low order 16-bits of the display, as other key switches are pressed. The following procedure shows how to set up and initiate the console micro-test.

1. Turn the PDP-11/60 OFF, turning the rotary switch to the STD BY position.

2. Put clock in free run mode, (see Chapter 2.1.4).

3. Jumper Bus DIN<13>(E02M2) to ground.

4. Turn machine ON.

5. Remove jumper, (put on in step 3).

6. The console microdiagnostic is now running.

7. When through using the diagnostic, INIT the machine by pressing HALT, START key switches.

## Table 3-2 Key Switch Codes

| Octal Code | | |
|---|---|---|
| Display Position 2 | Display Position 1 | Key Switch Name |
| 0 | 0 | |
| 0 | 1 | Not Used |
| 0 | 2 | |
| 0 | 3 | |
| 0 | 4 | *DIAG |
| 0 | 5 | *BOOT |
| 0 | 6 | *START |
| 0 | 7 | *CONT |
| 1 | 0 | DEP |
| 1 | 1 | *MAINT |
| 1 | 2 | (L)ADRS |
| 1 | 3 | EXAM |
| 1 | 4 | (D)ADRS |
| 1 | 5 | (D)SWR |
| 1 | 6 | *(L)SWR |
| 1 | 7 | HALT/SI |
| 2 | 0 | 0 |
| 2 | 1 | 1 |
| 2 | 2 | 2 |

| | | |
|---|---|---|
| 2 | 3 | 3 |
| 2 | 4 | 4 |
| 2 | 5 | 5 |
| 2 | 6 | 6 |
| 2 | 7 | 7 |

*Code generated in conjunction with the CNTRL key.

**NOTE**

**Damage to hardware may result if jumper in Step 3 is attached while machine is running.**

The first thing the console microdiagnostic does is check to see that the jumper is removed from Bus DIN<13>.

Once the jumper has been removed the entire machine is initialized.

The microdiagnostic then checks to see if a keyswitch has been pressed. It accomplished this by testing service. Since the entire machine has been initialized the only thing that can be requesting service is the console. If a keyswitch has not been pressed then the microdiagnostic proceeds to load the display with the contents of the Temporary Switch Register.

# CHAPTER 4
# MICRODIAGNOSTICS (DCS), MACRODIAGNOSTICS (XXDP) - MAINTENANCE AIDS

## 4.1 MICRODIAGNOSTIC - DCS (M7871)

The 'DCS' (Diagnostic Control Store) Module is a diagnostic tool specifically designed for the PDP-11/60 [KD11-K] Central Processor. Functioning as an alternate 2048 word control store, microcoded tests are executed to detect and isolate errors within the internals of the processor control and datapath hardware. Error indication information is provided by the DCS module; coupled with an indexed FAULT DIRECTORY, errors are resolved to the module level, when possible. Additional information is also provided resolving the error to a specific functional logic block. Significant benefits of this micro diagnostic approach are seen to be:

- Memory/I-O Device/UNIBUS independence
- Direct hardware microcontrol and visibility
- Extremely fast load and execution times
- Excellent coverage and resolution

To use the DCS, the following hardware is required:

1. DCS (Diagnostic Control Store) Module M7871 (KU116-BB)
2. PDP-11/60 [KD11-K] Central Processor
3. DL11-W Line Clock (required)
4. First 4K memory bank (required minimum)

### 4.1.1 Documentation/Listings

The available documentation for the DCS module user comprises the following items:

- DCS User's Guide (portions of which are discussed in this Chapter).
- FAULT DIRECTORY Listing; for module replacement information
- DCS Microcode Listing; when IC level debug is necessary
- DCS Maintenance Manual, Print Set; for detailed information on DCS hardware operation
- KD11-K Processor Maintenance Manual, Print Set; for IC level debug, base machine hardware specific information

Specific MAINDEC component part numbers for the DCS documentation are as follows:

MD-11-DQKUB-*-D;

User's Guide, FAULT DIRECTORY (PAPER, 80 pages)

MD-11-DQKUB-*-LA;

DCS Microcode Listing (PAPER, 450 pages)

MD-11-DQKUB-*-FA;

User's Guide, FAULT DIRECTORY, DCS Microcode Listing (FICHE, 4 cards)

### 4.1.2 Loading Procedure

The DCS occupies slot 1 in the KD11-K processor backplane; thus if an ECS or UCS option is present, it must be removed. To load the DCS, the following sequence should be employed:

1. Power down the CPU

2. Remove the ECS/UCS option from processor slot #1, if present

3. Insert the DCS module into slot #1. Use caution while inserting the DCS module, as a slightly bowed board may require some gentle maneuvering to seat it in place.

4. Orient the DCS 'RUN/STOP' switch to 'STOP', and the 'NORMAL/VERIFY' switch to 'NORMAL'

5. Now power up the CPU

### 4.1.3 Starting Procedure

DCS execution can be initiated by two distinct methods:

**4.1.3.1 Startup Via the Operator's Console (KY11-P)** -- With the KD11-K CPU in "CONSOLE" ("HALTed") mode, simultaneously depress the KY11-P operator's console "CNTRL/DIAG" keys to start the DCS. (If no DCS/ECS/UCS happens to be present, there should be no effect.) Chapter 4.1.3.2 (below) interprets the display on the operator's console while the DCS is executing, and after it stops. If this method fails to start the DCS, proceed to the next paragraph.

**4.1.3.2 Startup Via the DCS Switches** -- If it is desired for some reason to bypass use of the KY11-P operator's console to initiate the DCS, an alternative means is provided. This method would be used, for example, in the event that starting from the operator's console was not possible (using "CNTRL/DIAG") due to some hardware malfunction. The procedure is:

1. On the DCS module, set the 'NORMAL/VERIFY' switch to 'NORMAL'

2. Now set the 'RUN/STOP' switch (on DCS module also) to the 'RUN' position, or flip it from 'RUN'->'STOP'->'RUN'

The DCS should now assume control of the KD11-K CPU, regardless of

the previous state of the CPU ("CONSOLE", "RUN", or whatever).

**NOTE**

**If the DCS 'RUN/STOP' switch was already in the 'RUN' position, then it is expected that the operator's console keys have no effect - the DCS is already enabled to execute, and is controlling the CPU. The DCS microcode does not monitor the console keypad for operator input.**

If neither of the above methods produces a 'RUNNING' indication of the DCS (as per Chapter 4.1.4.1 then proceed to the next paragraph.

**4.1.3.3 Using Console "INIT"** -- Set the DCS switches as detailed in Chapter 4.1.3.2 (i.e., 'NORMAL' and 'RUN'). Now generate a processor "INIT" signal by simultaneously depressing the "HALT" and "START" keys on the operator's console. This should now initialize the CPU logic and restart the DCS, producing a 'RUNNING' indication (see Chapter 4.1.4.1).

If no 'RUNNING' indication is now present, then a problem exists in the power supply, system clocks, DCS module, or ???.

**4.1.4 Operating Procedure**

**4.1.4.1 DCS Indications While Executing on the KD11-K Operator's Console** -- While the DCS is executing, the operator's console display should be approximately as follows:

"RUN" - ON Continuously

"PROC" - BLINKing

"USER" - BLINKing

"CONSOLE - OFF Continuously

"BATTERY" - <indeterminate>

The 6 digit octal display should read:

( 0 0 0 0 0 0 ) - In DCS pass 1:

(.0.0.0.0.0.0) - In DCS passes 2-64.

If any of the above conditions are not met, then either:

1. The console hardware is inoperable, or

2. The DCS is hung in an 'error/scope loop'
   (Not necessarily in that order)

**4.1.4.2 DCS Indications While Executing** -- On the DCS module, the two important indicators to watch at this time are the 'ERROR' and "EOP' LEDs:

| LEDs | | | |
|---|---|---|---|
| 'ERROR' | 'EOP' | State | Comment |
| OFF | OFF | Probable ERROR | DCS and/or processor HUNG |
| OFF | ON | * EOP * | Successful EOP |
| ON | OFF | * ERROR * | Genuine ERROR |
| ON | ON | Probable ERROR | DCS 'VERIFY MODE' indication |

Note the 'Probable ---' notation. Either 'EOP' or 'ERROR' on the DCS should be lit: both OFF or both ON indicate non-standard conditions that require further investigation. Neither 'ERROR' nor 'EOP' indicates that the DCS and/or processor are in a HUNG condition (e.g., clocks suppressed) See Chapter 4.1.9 (Error Handling). Both 'ERROR' and 'EOP' is the standard DCS 'VERIFY MODE' indication. Make sure the DCS 'NORMAL/VERIFY' switch is set to 'NORMAL' (if this is intended); and then restart the DCS. See Chapter 4.1.12 (Verify Mode) for further information.

### 4.1.5 DCS Execution Time

The execution time of the DCS will vary depending upon its mode of initiation.

**4.1.5.1 "CNTRL/DIAG" Start from Operator's Console** -- When started via "CNTRL/DIAG" from the operator's console, and assuming no errors, DCS will execute with console display as detailed above for approximately 6 seconds (64. passes at 100 milliseconds/pass,

about 350,000 microcycles/pass). After this time has elapsed, control should return to the KD11-K microcode, as if the CPU had just been powered up with the slide switch set to "HALT". Note that the machine will end up in console mode regardless of the actual slide switch settings (RUN/BOOT/HALT). Chapter 4.1.6 interprets the console and DCS displays after this delay time, for successful 'END-OF-PASS' and 'ERROR' indications.

**4.1.5.2 Start from DCS Switches** -- When started via setting 'RUN/STOP' to 'RUN' on the DCS module, the DCS will execute continuously, not returning to the KD11-K microcode until the 'RUN/STOP' switch is reset to the 'STOP' position. Throughout this time, the execution indications will be as previously detailed. When the switch is returned to 'STOP', the DCS may execute for a maximum of approximately 6 more seconds; and then proceed as described directly above, entering "CONSOLE" mode.

### 4.1.6 DCS 'END-OF-PASS' and 'ERROR' Indications

**4.1.6.1 On the DCS Module** -- Successful 'END-OF-PASS' and 'ERROR' conditions detected by the DCS are indicated most directly on the DCS module by the two LEDs labeled 'EOP' and 'ERROR'. 'EOP' is turned on at the end of each successful pass thru the DCS code - assuming 'ERROR' has not yet been turned on. In an error-free running situation, 'EOP' will be on; 'ERROR' off. Note that 'EOP' comes on at the end of the first pass (assuming no errors), and thus will appear to be on continuously.

In the event the DCS detects an error, the 'ERROR' LED will be turned on, and the 'EOP' LED turned off. The 'ERROR' LED will be latched in such a way that it cannot be turned off if the error disappears (i.e., a 'flaky' timing error), thus retaining the indication of an error. See Chapter 4.1.9 (Error Handling) for a full treatment of the various error conditions: Detection, Indication, Scope-looping.

**4.1.6.2 On the Operator's Console** -- Successful 'END-OF-PASS' and 'ERROR' conditions are indicated on the operator's console as follows:

1. A successful 64. passes and return to KD11-K processor microcode control, by a

   (.1.2.3.3.2.1)

   in the console display, and the KD11-K processor "HALTed" in "CONSOLE" mode.

2. An error in pass 1 will usually (but not always) be indicated by a

   ( Ø Ø Ø Ø Ø Ø )

   in the console display.

3. An error in pass 2 through -- will usually (but not always) be indicated by a

   (.0.0.0.0.0.0)

   in the console display.

Note the qualifications in the previous statements:

For a successful 'END-OF-PASS' and exit back to KD11-K microcode control, only one display is valid:

(.1.2.3.3.2.1)

For any other display except "(.1.2.3.3.2.1)", check the DCS module 'EOP' and 'ERROR' LEDs for the most reliable indication of the result of execution.

See Chapter 4.1.9 for a full explanation of error processing.

### 4.1.7 Hardware Restrictions

**4.1.7.1 Cache and Memory Management (KT)** -- The DCS executes with both Cache and KT disabled:

1. Cache, by setting both the "Force Miss" bits

2. KT, by clearing the "Enable" bit

The DCS checks the most basic path from the UNIBUS to/from internal data paths. Further macro diagnostic programs are available for Cache (MD-11-DQKKA-*) and Memory Management (MD-11-DQKTA-*) fault diagnosis.

**4.1.7.2 MOS Memory Battery Backup** -- The MOS memory battery backup (if present) must either be "Good", or else disabled. Otherwise, the micro diagnostic code WILL NOT execute without detecting an error.

Indications are: ERCD=5621/TNUA=7400, in TEST620C.

### 4.1.8 Software Restrictions

Return to Console/DCS End-of-Pass Processing

Console "HALT" mode is the ONLY exit provided from DCS back to base machine microcontrol. This action is due to:

1. The DCS, upon detecting an error, locks itself (and the processor) up in such a way that manual intervention by the operator is required for return to base machine microcontrol.

2. The DCS completely alters the internal microstate of the processor, destroying its previous contents, and leaving

"garbage" in its place. Thus the full base machine power-up "INIT" sequence is generated by DCS to "clean-up" the processor prior to returning control.

### 4.1.9 Error Handling

The concept of an 'ERROR' in DCS terms is very simple. It involves the use of the ENUA (Expected NUA), TNUA (Tracking NUA), and DCS COUNTER registers; all of which are local to the DCS module. The 11/60 processor itself has no control over the setting/clearing of 'ERROR'; in fact, it cannot directly determine whether 'ERROR' is set or clear.

The ENUA register (12 bits) is loaded from the EMIT field of the microword, under control of a DCS rom extension bit. It is set up at the beginning of a test to reflect the "EXPECTED" micro address after the test microbrancyh ("BUT") is executed.

The TNUA register (12 bits) is loaded continuously as the DCS microcode executes, TRACKING the progress of the microaddress field. This register contains the value of the "RECEIVED" micro address after the test microbranch is executed.

The DCS-CNTR is loaded with a value from (00)-(17) (octal), from the EMIT field of the microword, under DCS rom extension control. This register continuously counts up every microcycle. When the contents of this register is (17), the DCS hardware compares ENUA and TNUA, and does the following:

Set 'ERROR'="1" if DCS-CNTR=(17) and ENUA TNUA

else leave 'ERROR' unchanged from its previous value.

This is the manner by which DCS is able to set 'ERROR'. All DCS tests use this method.

Note also that the DCS hardware "locks up" the loading of the ENUA and TNUA registers after 'ERROR' is set, preserving their contents. Thus only the FIRST 'ERROR' will be recorded. There is no provision to detect subsequent errors until the previous ones are eliminated. See the DCS Maintenance Manual and Print Set for more detailed information.

### 4.1.10 FAULT DIRECTORY Format and Use

**4.1.10.1 Basic Structure** -- The FAULT DIRECTORY is essentially a tabular summary of all ERROR codes the DCS is able to generate - a total of 432 entries occupying .52 pages. Each individual ERROR code entry in the FAULT DIRECTORY contains a short description of the test, and the module replacement information pertaining to that test. For ease of reference, the ERROR codes have been organized into ascending numerical order, in the range 4000(8)-6777(8).

**4.1.10.2 Basic Use - With an Example** -- This section describes how to use the FAULT DIRECTORY after the DCS has been run, and has

indicated an error is present in the KD11-K processor.

Assume for the purpose of explanation that the DCS was started, and has returned the following values:

ERCD = 4616 (Error Code)

TNUA = 7405 (Tracking NUA)

with EOP=<OFF>, and ERROR=<ON>

1. Going to the FAULT DIRECTORY, we find the entry for ERROR code 4616 to be on page 9, entry number 73.

2. Some general information about the failing test is first obtained:

   a. 'Symbolic label' - A reference to the DCS microtest which failed, in this instance TEST 115 A2.

   b. 'Line number' - A reference to the line number in the DCS microcode listing where the failing test is located (here, line number 5993).

   c. 'ENUA' - the Expected NUA of this test, in this case 7412. Note that the obtained TNUA (7405) is not the same as the test's ENUA (7412); thus the ERROR.

d. The remainder of the line contains a short description of the function performed by this test; in this instance we note the test was diagnosing the ALU portion of the DATA-PATH module.

3. We now note that the TNUA we obtained was 7405. Scanning downward in the column of TNUA entries for this test, we find it listed as the fourth entry. More information, specific to this particular error, can now be obtained:

a. 'Module sequence' - These 3+ columns contain (scanning left to right) the top 3+ choices of processor modules to inspect/replace, in order to locate and correct the fault(s). The module choices are listed using "slot" notation (i.e., K#, where #=the slot), and a "confidence factor" to indicate the percent confidence that replacing this particular module will eliminate the fault(s). The best choice is the module called out in the first column ("#1"); then "#2" etc. Note that the percentages are rounded to the nearest 5%, and may therefore not always add up to exactly 100%.

b. To the right of the module choices is summarized the IC information obtained from the FAULT INSERTION effort of the DCS/KD11-K (signified by "FI"). IC information is referenced to a particular module by

the notation:

K4=E1Ø, E22, E24, E98; K3=E84

NOTE

**Callout of specific IC's on a module is --NOT-- intended to be an "Exhaustive-Only-These-Are-The-Ones" list. It is intended to provide reference to a specific functional area of a module, and give reference to those IC's which caused the failure during the fault insertion effort of hard STUCK-HIGH/-LOW and ADJACENT-PIN-SHORT type faults. Again, do not assume this list to be all inclusive of the possible choices for faulty IC's.**

Another type of entry is of the format:

K4Ø4=ALU/CARRY-LOOKAHEAD; or

Kmpp=functional-description-of-logic-block

which references a particular module (#m) and page (#pp) in the KD11-K Processor Print set. This notation is used when specific fault insertion data is not available for a test.

4. In the instance when the TNUA obtained does not match any of those provided under a given test/ERROR code entry, a wild-card character ("?") has been used to allow a match with any octal digit. Thus 740? matches 7400, 7401, .... 7407. These entries should be used for further information or when a specific TNUA is not present.

5. If there is no TNUA listed which matches the obtained TNUA, and also no wildcard entry is present; then the information about the functional nature of the test (from above), along with an intelligent interpretation of the obtained TNUA, will be required. The following table lists some TNUA's that might be obtained in such a case:

| **TNUA** | **CAUSE** |
|---|---|
| 4000 | DCS forced to its starting address |
| 4777\<br>4756 ><br>4747/ | an unexpected JAMUPP condition occurred |
| 7361 | in UNIBUS function tests, a JAMUPP did not occur when expected |
| 7400- | the "standard" BUT() target area for DCS |

7777 micro-tests

### 4.1.11 "SCOPE Loop' Facility

**4.1.11.1 General Information** -- The 'SCOPE loop' implementation provided by DCS is almost identical to that provided in the standard MAINDEC macro diagnostic program. What the 'SCOPE loop' does is to repeatedly execute the same sequence of diagnostic test code; this allows the technician to 'scope' appropriate logic signals in an effort to zero-in on the fault.

The DCS 'SCOPE loop' occurs ONLY and ALWAYS when 'ERROR' is set. There are no user options to change the size or range of the loop -- all these parameters have been fixed in microcode and hardware. The loops have been set up to be as tight, and as useful, as was possible. Most are in the range of 10-30 microwords, although some (three, in particular) are larger.

**4.1.11.2 Implementation and Use** -- A DCS extension rom control bit is used to enable the 'SCOPE loop' check at selected points in DCS code execution. These points are recognized by the following:

```
SCOPE123:
     <possible some other functions>
     NEXT,    BUTD[SCOPE],    !NO ERROR:    "TEST124" [+1. WORDS]
              J/TEST124       !   ERROR:    "LOOP123" [-5. WORDS]
```

The two comments "ERROR/NO-ERROR" tell the user where the DCS code will branch, depending upon the current state of 'ERROR'.

Usually, the 'NO-ERROR' condition falls through to the next word (e.g., +1. words). For the 'ERROR' case, the loop is ALWAYS backwards (i.e., up the page, toward the point where the error was detected). The "-number" notation gives a relative count of the number (approximately) of micro words backwards in the jump.

This facility can be used in two modes -- dynamic and static. Either mode must be entered via the use of the DCS 'RUN/STOP' switch set to the 'RUN' position, as this then enables the DCS code to execute continuously. The results are generally undefined if the switch is set to 'STOP', and the "CNTL DIAG" buttons were used to enter the loop.

Dynamic mode requires the use of an oscilloscope, logic analyzer, etc., and the determination of an appropriate logic signal on which to sync. The DCS microcode then automatically remains in this tight loop to allow observation of the suspected faulty signals, at processor cycle speed.

Static mode is entered in the same manner as dynamnic; but afterwards the "SINGLE-MICROSTEP/MAINTENANCE-CLOCK" feature of the 11/60 processor ((on K6 TIMING module, the two switches - see prints) is enabled. This allows the processor to be single micro-stepped, under user control. The additional debug features of the DCS can now be employed; the BUSDIN/DOUT display LEDs (16), and the (2) "free" LEDs. See the DCS Maintenance Manual for further details. Also available are the NUA (Next-U-Address) LEDs

on the 11/60 processor "UWORD' module (K2). Note that these 'point' at the NEXT microword to be executed, not the current.

### 4.1.12 DCS Verification/Self-Test

'VERIFY MODE' is a self-check mode designed to verify the operation of the DCS module and its associated error detection/ indication support logic.

**4.1.12.1 Requirements** -- This mode of operation requires that a known good PDP-11/60 system be used to test/verify a DCS module, so that errors detected by the DCS are due to the DCS module under test, and not due to the other system components. The set of PDP-11/60 processor macro diagnostics, or a known good DCS module, can be used to perform such a verification of the host system.

**4.1.12.2 Verification Method** -- The method (or algorithm) used to perform the DCS self-verification is as follows:

Hardware on the DCS module is conditioned to execute a single pass through the DCS microcode, via setting the DCS 'NORMAL/VERIFY' switch to the 'VERIFY' position. This also alters the 12 bit hardware counter on the DCS module from a 'Pass Counter' to the 'Verify Counter'.

At the start of a 'Verify Pass', this counter is preset to a specific value; predetermined so that when 'END-OF-PASS' is signaled by the DCS microcode, this counter will have a value of

octal (7777), or be at the point of overflow (carry out) enabled.

As the DCS executes in 'VERIFY MODE', this counter is incremented whenever:

1. A microword is executed from page 7, or

2. A microword is executed with the 'VERIFY' bit (a page 4-6 only DCS ROM extension bit) asserted. These 'VERIFY' bits have been scattered, more or less at random, throughout the DCS microcode. Thus this counter will be incremented at random intervals during a 'Verify Pass'.

The DCS code executes approximately 350,000 microwords per pass; thus the counter will overflow between 2-85 times (depending upon the number of 'VERIFY' bits and page 7 references encountered) before the 'END-OF-PASS' / 'Verify Counter' overflow match. Physically, the verify count is retained module 4096 (12 bits), with only the low order bits of the count used in the comparison.

A verification will be considered successful only if a verify counter overflow point exactly matches the microword which signals 'END-OF-PASS' (done only once) in the DCS microcode.

To run the DCS in verify mode, the following procedure is followed:

1. Install the DCS in the PDP-11/60 as detailed in Chapter 4.1.2

2. Set the DCS switches:

   'RUN/STOP' = 'STOP' and
   'NORMAL/VERIFY' = 'VERIFY'

3. Now set:

   'RUN/STOP' = 'RUN'

4. The DCS now executes a single 'Verify Pass'

5. At the end of the 'Verify Pass' the DCS enters a micro-code loop, in which:

   - An error is forced with specific
     'ENUA', 'TNUA', and 'ERROR code' values
   - 'END-OF-PASS' is repeatedly signaled
   - A 'Scope Loop' branch is executed

6. At this time examine the DCS module LEDs for comparison with their expected contents, as noted below.

7. To return control to the PDP-11/60 after a 'Verify Pass', position:

'RUN/STOP' = 'STOP' and

'NORMAL/VERIFY' = 'NORMAL'

And then generate a "CONSOLE INIT' ("START/HALT") on the operator's console

Only the status described below is acceptable to signal a successful DCS verification. Assuming a known good PDP-11/60 system, any deviation from the description (below) should be considered an indication of a fault in the DCS module under test.

After a 'Verify Pass', indications on the DCS module will be:

'TNUA' = (7522)

'ERROR' = (4255)

(Note the alternating ON/OFF pattern)

'ENUA' = (7523) was loaded to force an 'ERROR' indication

'EOP' LED - ON, Approx. 1/2 brilliance

'ERROR' LED - ON, continuously

This will be the only instance when both the 'EOP' and 'ERROR' LEDs should be on simultaneously.

Indications on the PDP-11/60 console should be:

"RUN" LED - OFF continuously

"PROC" LED - OFF continuously

"USER" LED - OFF continuously

"CONSOLE" LED - OFF/ dimly lit

"BATTERY" LED - <indeterminate>

Octal display = (212121), with the decimal points either on or off.

## 4.2 MACRODIAGNOSTICS (PDP-11/60 XXDP)

### 4.2.1 PDP-11/60 XXDP

XXDP is a catch-all name for a group of binary packages available for loading devices listed in Table 4-1. All XXDP packages require a console device (teletype, LA36, etc.), and one of the diagnostic package media. The previously stated requirements are for loading and running diagnostic programs already stored in one of the diagnostic package media. They are also sufficient for implementing permanent patches on programs when required. The XXDP monitor is loaded via the M9301-YH bootstrap module; refer to Chapter 2.3 for bypass routines. Complete documentation is contained in the XXDP User Manual, MAINDEC-11-DZQXA.

The XXDP packages have been designed for diagnostic purposes only.

The software used is not intended to be compatible with any other PDP-11 family software; any nondiagnostic uses of the software in other than the manner described in the XXDP user manual are not supported.

The XXDP packages are binary packages only. They provide the PDP-11 family diagnostic programs in the various media described. Documentation for each of the programs stored in an XXDP package must be obtained separately. However, said documentation must be obtained at the same time as the package, in order to ensure that the documents and programs are at the same rev. level.

**Table 4-1 PDP-11/60 Peripheral Devices Having XXDP Available**

| Device Code (Octal) | Model | Description |
|---|---|---|
| 1 | TM11/TU10 | Magnetic Tape |
| 2 | TC11/TU56 | DECTAPE |
| 3 | RK11/RK05 | DECPACK DISK Cartridge |
| 4 | RP11/RP03 | DISK |
| 5 | RK611/RK06 | DISK |
| 6 | RH11/TU16 | MAGTAPE |
| 7 | RH11/RP04 | DISK |
| 10 | RH11/RS04 | Fixed Head Disk |
| 11 | RX11/RX01 | DISKette |
| 12 | PC11 | High Speed Reader |

### 4.2.2 PDP-11/60 Diagnostics

The diagnostics described in the following paragraphs are available on all of the XXDP media except cassette. It is suggested that the diagnostics be run in the order in which they appear in the following paragraphs. Operating procedures and listings for all diagnostics are contained in the MAINDECs for each program.

**4.2.2.1 "DQKDA" - KD11-K Basic Logic Tests** -- "DQKDA" is a diagnostic program designed to detect, report, and identify logic faults in the KD11-K central processing unit of the PDP-11/60 system. It consists of 504(10) individual tests carefully designed and sequenced to detect and attempt to identify logic faults at a minimum hardware/software level. These tests are partitioned into four major sections as listed below:

1. Basic CPU Tests (BCPT)
2. Basic Instruction Tests (BIT)
3. Comprehensive Instruction Tests (CIT)
4. Combine Instruction Exerciser

**4.2.2.2 "DQKDB" - 11/60 Trap Tests** -- This program checks that on all trap operations register 6 is decremented the correct amount, that the correct PC is saved on the stack, that the old condition codes and priority are placed on the stack and that the new status and condition codes are correct. Both the "TRAP" and "EMT" trap instructions are tested to see that all combinations will trap. Verification of the "RTT" instruction (000003) which is used for

software debug routines: ODT,DDT, is done. Also, the trace bit is checked to see if it causes a trap. Yellow and red zone violations are checked. The RTI and RTT instructions are checked for correct stack operations.

**4.2.2.3 "DQKTA" - PDP-11/60 Memory Management Diagnostic** -- This program will test all of the memory management logic, including the stack limit register logic, and enable the field service representative to isolate the detected failures to a replaceable module. It is assumed that the CPU has been tested, or is known to be functioning correctly, and that the program is started from address 200. The 11/60 series cache is turned off for the first pass of the program and is turned back on for the second and subsequent passes. This will provide the earliest detection of memory management related errors and enable looping on the error involving minimum logic. This program may also expose faults that are on the interface between memory management and other sections of the computer.

**4.2.2.4 "DZQMC" - 0-124K Memory Exerciser**

**4.2.2.5 "DQKAA" - 11/60 Cache Diagnostic** -- The 11/60 Cache Diagnostic is comprised of a series of tests which were designed to check the cache's data paths on the Cache/KT board and its control logic on the Bus Control module. The tests are arranged in a logical order such that they build on one another. That is, the currently running test will depend on logic exercised by

previous tests. Basic cache operations are exercised first followed by address and data functions. Those tests requiring extensive amounts of cache functioning are done near the end of the program. This testing procedure should provide a very effective degree of fault isolation.

**4.2.2.6 "DQFPA" - 11/60 Floating Point Unit, Basic Instruction Tests** -- This program is the basic functional test for the PDP-11/60 floating point processor. Functionality tests of all status registers and accumulators are performed to verify their operation (e.g., rippling bit tests, alternating bit tests, unique reference tests). All address modes (source, destination, floating source, floating destination) are tested for correct operand reference, and verification of side affects. Finally, the basic no-operand and single operand instructions are tested to insure their full functionality in all PDP-11/60 FPU modes. Both "HOT" (FP11-E option) and "WARM" (PDP-11/60 microcode) floating point units can be selected for testing.

**4.2.2.7 "DQFPB" - 11/60 Floating Point Advanced Instruction Tests** -- This program extends the testing of instruction functionality to the remainder of the PDP-11/60 floating point instruction set not covered in the basic instruction tests. Full testing in all PDP-11/60 FRU modes of all the multiple operand arithmetic, comparison, and integer to float conversion instructions is performed. Both "HOT" (FP11-E option) and "WARM" (PDP-11/60 microcode) floating point units can be selected for

testing.

**4.2.2.8 "DQFPC" - 11/60 Floating Point Unit Instruction Exerciser** -- This program contains the PDP-11/60 floating point processor instruction exerciser. This exerciser program tests the complete floating point instruction set in an exerciser environment, using various combinations of instructions in common software configurations as the basis for the tests. Every conceivable floating point error condition is developed, and the response checked for correctness. In addition, interaction between floating point instruction execution, interrupts, and base machine interrupts (using the DL11-W line and/or KW11-P programmable clocks) is also tested to insure correct processing by both the base machine and floating point microcode. Both "HOT" (FP11-E option) and "WARM" (PDP-11/60 microcode) floating point units can be selected for testing.

**4.2.2.9 "DQFPD" - 11/60 Floating Point ADD/SUB/MUL/DIV Exerciser** -- This program is an exerciser for the PDP-11/60 floating point add, subtract, multiply, and divide instructions. Random number patterns are used as the operands, and the hardware generated results are checked against results obtained from floating point software routines to insure correctness. The PDP-11/60 is operated in double and single floating mode, round and truncate mode, and with underflow and overflow conditions enabled and disabled. The program will run for 400(8) "subpasses" before giving an "end of pass" indication, so that a sufficient

number of random patterns are obtained for use as operands. Also at this time, optional status information on the types of random operands selected can be printed on the console. Both "HOT" (FP11-E option) and "WARM" (PDP-11/60 microcode) floating point units can be selected for testing.

**4.2.2.10 "DZDLD" - DL11-W Diagnostic**

**4.2.2.11 "DQKDC" - 11/60 Series CPU Exerciser** -- This program is designed to be a comprehensive check of the PDP-11/60 series CPU cluster. The program executes each instruction in all address modes and includes tests for traps, interrupts, floating point, memory management, memory, the Unibus and Massbus. If not deselected, the program relocates the test code throughout memory (0-124K). Also, if not deselected, the program will relocate using available disks (RP03, RK05, RP04, RS03/4).

Since worst case testing occurs with all switches down, precautions must be taken to ensure the protection of user disks.

**4.2.2.12 "DZKAQ" - 11/60 Power Fail Test** -- This test checks out the Power Fail System.

**4.2.2.13 "DZM9A" - 11/60 Bootstrap/Terminator Diagnostic** -- This diagnostic program is intended to verify the rom contents of the rom bootstrap modules. The program computes and checks a cyclic redundancy character and a longitudinal parity character for the

contents of the rom storage available in an M9301 or M9400 module.

A separate routine included allows the user to type the contents of the rom storage on the teletype as an aid to debugging.

**4.2.2.14 "DZKUA" - Unibus System Exerciser**

**4.2.2.15 "DZKUB" - Unibus Exerciser Module Diagnostic**

**4.2.2.16 "DZMML" - MS11-K MOS Memory Tests**

**4.2.2.17 "DQM9A" - PDP-11/60 ROM Bootstrap/Test Program (Listing for the M9301-YH Module)**

**4.2.2.18 "DQKUA" - 11/60 WCS Diagnostic** -- The WCS diagnostic program is intended to be used for checking the writable control store option of the PDP-11/60 processor. It has been designed to verify that the WCS operates correctly and attempts to diagnose faults that may occur therein.

The WCS is a PDP-11/60 option that provides the user with an ability to do his own micro-programming. The micro-code can be loaded and stored in the WCS and executed upon specific instructions from the base machine.

**4.2.2.19 "DQKUB" - KD11-K Micro-diagnostics** -- Refer to Chapter 4.1.

### 4.2.3 Peripheral Diagnostics

Diagnostic programs for peripherals and I/O devices in the system are listed and described in their associated maintenance manuals.

| ITEM | DWG. NO./PART NO. | DESCRIPTION |
|---|---|---|
| A | MD-7419055-0-0 | BRACKET, CONSOLE MOUNTING |
| B, C, D | 9006073-03 | SCR, PH TRUSS HD #10-32 x .50 |
| E, F | 9006565 | NUT, KEPS #10-32 |
| G, H | 9006664 | WASHER, FLAT |
| J, K, M | 9006635 | WASHER, INT. TOOTH LOCK #10 |
| N | MD-7415771-07 | FILLER PANEL |
| P | MD-7419056-0-0 | PLATE, CONSOLE MOUNTING |
| R | 9009922-00 | SPACER |
| S | 9009923-00 | NUT, CLIP FLOATING 8-32 |

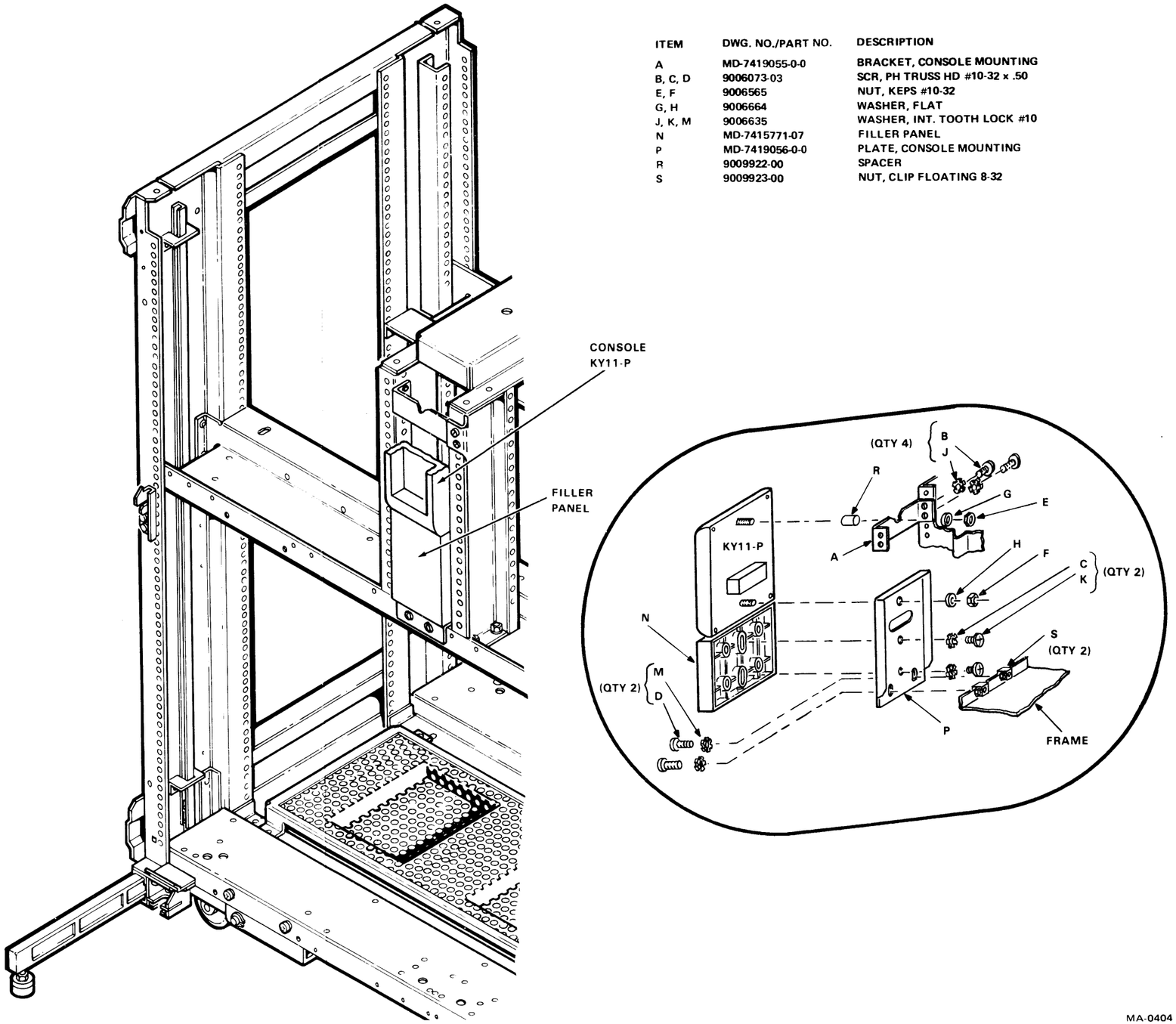



Figure 5-1 KY11-P Console Location Exploded View

# CHAPTER 5

# REMOVAL AND REPLACEMENT PROCEDURES

## 5.1 INTRODUCTION

No special procedures are required to disassemble and reassemble most of the components and assemblies in the PDP-11/60's Corporate Cabinet or processor mounting box. This Section outlines the procedures for removing and replacing modules and the steps required to disassemble the console. Refer to the PDP-11/60 Cabinet and Power Supply Manual for detailed procedures on removing and replacing the BA11-P boxes and H7420 power supplies.

### 5.1.1 Module Removal and Replacement

The multilayer modules used in the PDP-11/60 are equipped with lock/release handles, and each slot in the backplane has card edge and center guides that allow the modules to be installed easily. The card guides ensure that the modules are not removed or inserted at an angle so great that module or connector slots are damaged. Even though these features are provided, always install and remove the modules carefully.

### 5.1.2 Console Disassembly

Refer to Figure 5-1. The following steps are required to disassemble the console:

1. Turn power OFF at the circuit breakers.

2. Remove front panel.

3. Remove the two screws (Item D) that secure the Console Mounting Plate.

4. Lift the console up and away from frame.

5. Unplug the console harness connector from the KY11-P console filter board connector J1.

To remove the console's printed circuit board from the console frame, unscrew the four screws on the rear of the frame.

## 5.2 EQUIPMENT CONFIGURATION STATUS

The following paragraphs describe the Module Utilization List (MUL) sticker, the Engineering Change Order (ECO) status sticker, and the Mechanical Status sticker. The MUL sticker lists the equipment complement system serial number, etc.; the ECO status sticker provides information about the current ECO status of wire-wrap devices; module revision status (Mechanical Status) shows the current ECO status of the module.

### 5.2.1 Mechanical Status Sticker

The Mechanical Status sticker, see Figure 5-2, is located on the handle side of each BA11-P mounting box. A letter designates the

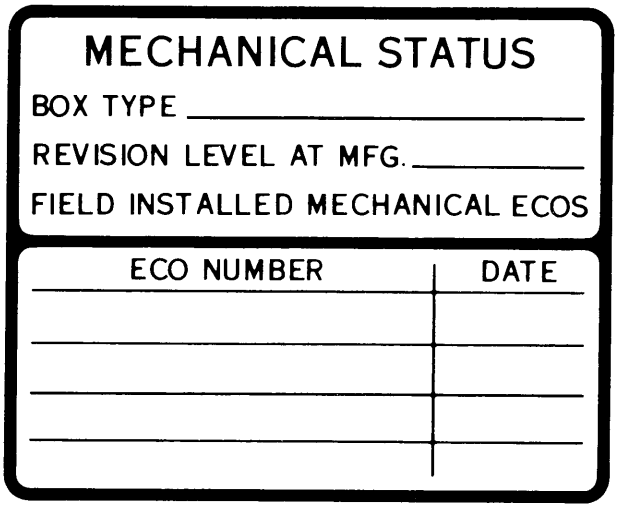



Figure 5-2 Mechanical Status Sticker

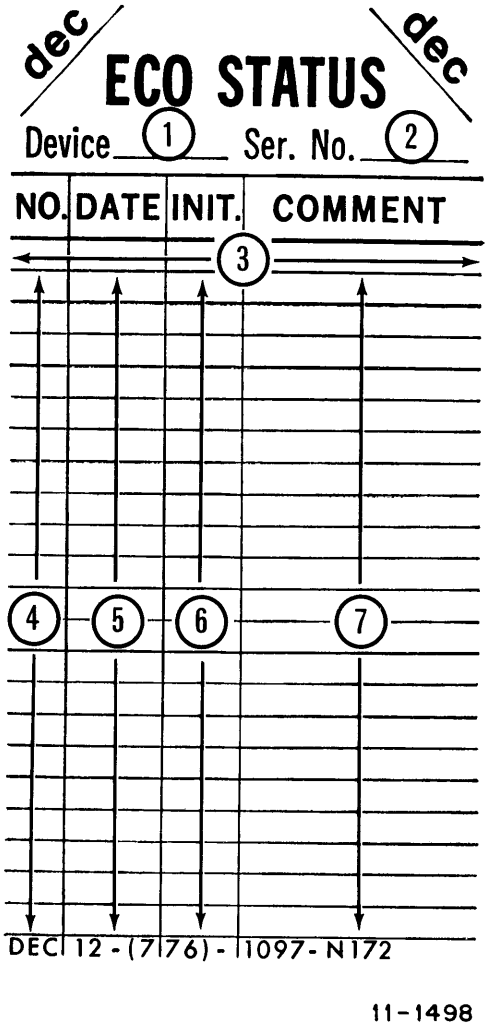



Figure 5-3 ECO Sticker

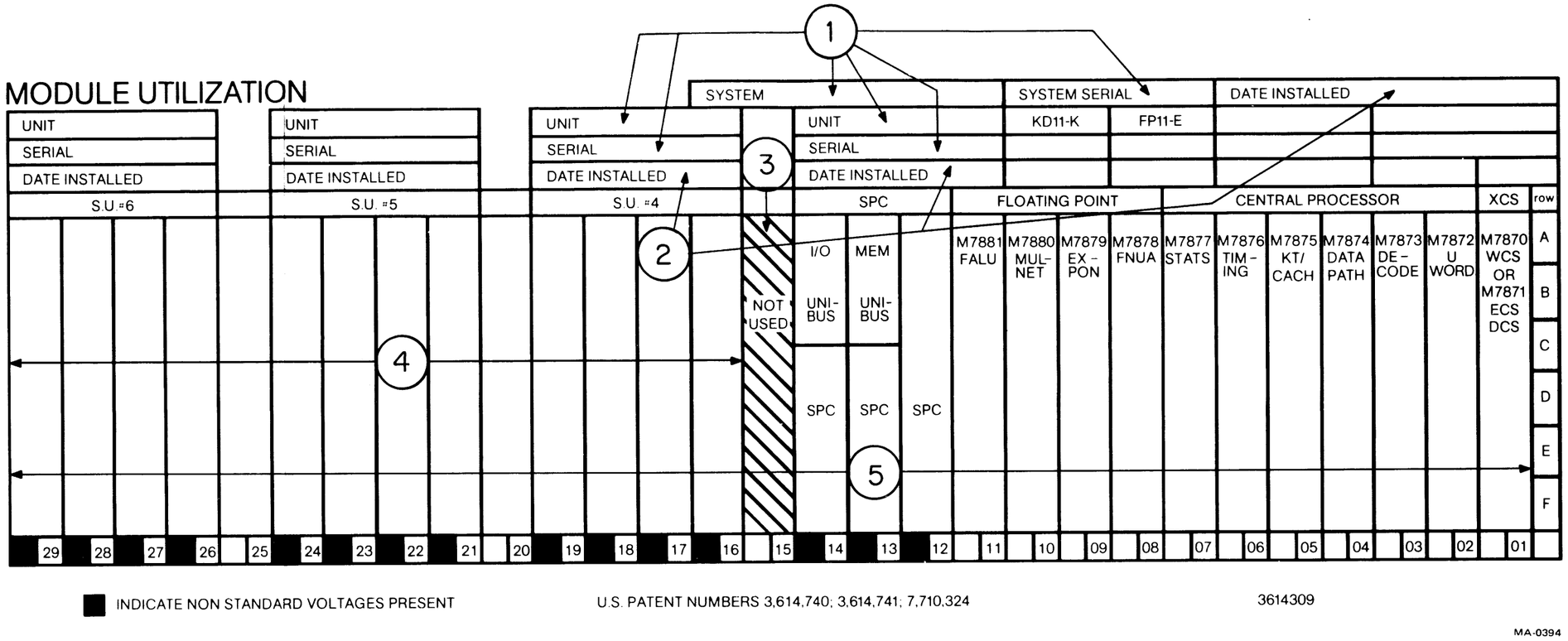



Figure 5-4 MUL Sticker (Base BA11-P)

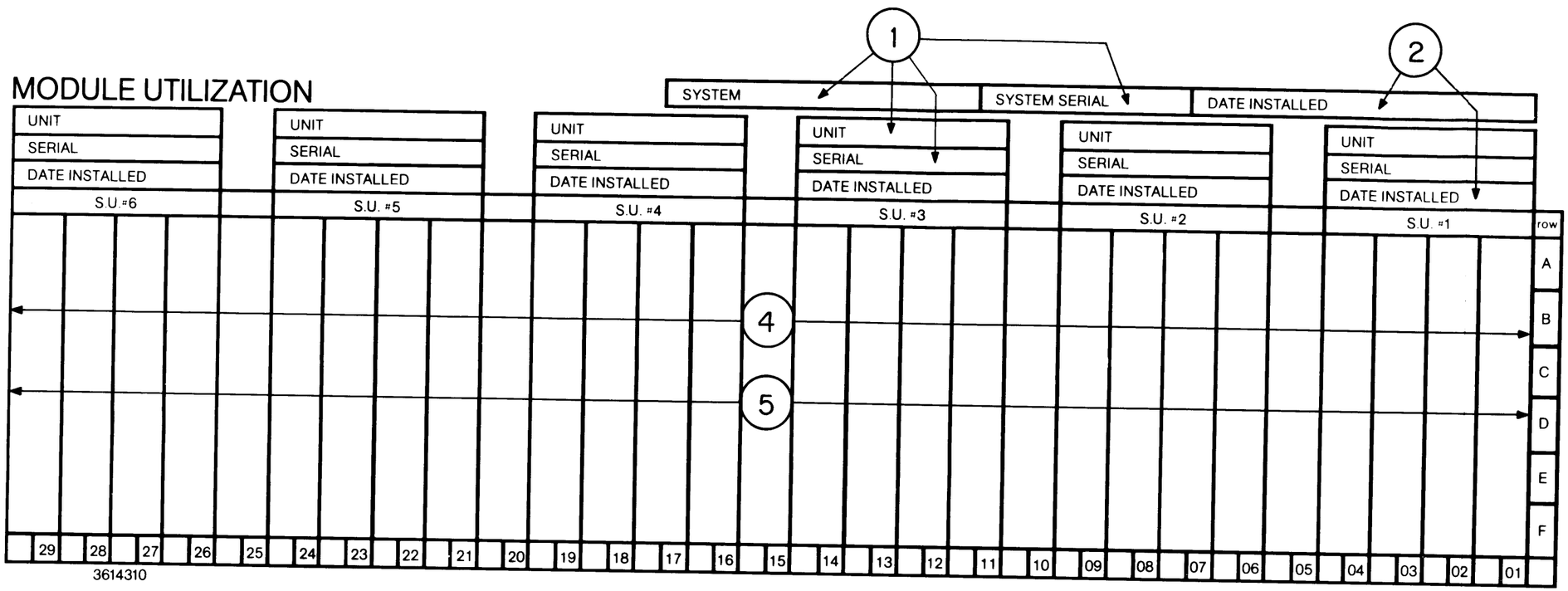



Figure 5-5 MUL Sticker (Expander BA11-P)

revision level at manufacturing. Field installed ECO's should be entered as performed.

### 5.2.2 ECO Status Sticker

The ECO Status Sticker, shown in Figure 5-3, is located is located on the rear center frame member of the PDP-11/60's corporate cabinet. This sticker is filled out for installation of ECOs to wire-wrap devices such as the KD11-K or the various system units. Table 5-1 describes how the various columns are to be filled out and the department responsible for filling out these items.

### 5.2.3 Module ECOs

Each module is stamped with the alphabet (except for G, I, and O) to record various circuit schematic revisions to a module. When a module is shipped from the factory, the actual revision letter from production is stamped on the handle. When ECOs that revise modules are installed in the field, scratch off the appropriate letters from the module. For example, if an ECO corresponding to revision F of the module was installed and an ECO corresponding to revision E of the module was not installed, the letter F would be scratched out and the letter E would remain intact.

### 5.2.4 Module Utilization List (MUL) Sticker

These stickers, shown in Figures 5-4/5-5, are located on the top of the BA11-P boxes; providing a quick, convenient tabulization of the various equipment located within the system.

Table 5-1 Completing the ECO Status Sticker

| Item No. | Responsibility | Description |
|---|---|---|
| 1 | Production | Option designation code for applicable device. |
| 2 | Production | Device serial number. |
| 3 | Production | Original wire wrap revision letter (e.g., ORIGINAL REV. B.) |
| 4 | Production/ Field Service* | Numerical portion of ECO/FCO number. |
| 5 | Production/ Field Service* | Installation date of ECO/FCO. |
| 6 | Production/ Field Service* | Initials of person installing ECO/FCO. |
| 7 | Production/ Field Service* | Necessary comments about ECO/FCO or its installation, (e.g., only part 2 installed). |

*If an option is installed in the factory, production has responsibility for filling out an ECO sticker. If the option is an add-on in the field, field service will fill out items 4 through 7.

NOTE: ECO Status Sticker is located on the inside of the module door of the mounting boxes.

Additional information such as serial number, comments, technical tips, and installation of partial ECOs is also shown on the sticker. Table 5-2 describes the manner in which the sticker is to be filled out and indicates the department responsible for filling out the various items.

## Table 5-2 Completing MUL Sticker

| Item No. | Responsibility | Description |
|---|---|---|
| 1 | Production | System Serial Number, Unit Serial Number |
| 2 | Field Service | Acceptance Date of installation at customer site |
| 3 | None | Not Used |
| 4 | Field Service | Comment area. Note ECO/FCOs installed, partial ECO/FCOs, miscellaneous information about module or slot. |
| 5 | Production/ Field Service* | Enter device type as installed. |

*To be filled out by production if option or device is factory installed. If option or device is an add-on in the field, field service will complete these items.

## 5.3 DESCRIPTIONS, REMOVAL AND REPLACEMENT OF ICs

### 5.3.1 IC Descriptions

The following Integrated Circuits (ICs) are described in this Section:

93516 (74161) Synchronous 4-Bit Binary Counter

74189 64-Bit Random Access Memory with 3-State Outputs

74194 4-Bit Shift Register

93516 (74161) Synchronous 4-Bit Binary Counter (Figure 5-6)

**5.3.1.1 74189 64-BIT RANDOM ACCESS MEMORY (Figure 5-7)** - These 64 bit active-element memories are monolithic Schottky-clamped transistor-transistor logic (TTL) arrays organized as 16 words of four-bits each. They are fully decoded and feature a chip-enable input to simplify decoding required to achieve the desired system organization. the memories feature p-n-p input transistors that reduce the low-level input current requirement to a maximum of -Ø.25 milliamperes, only one-eighth that of a Series 54S-74S standard load factor. The chip-enable circuitry is implemented with minimal delay times to compensate for added system decoding.

The three-state output combines the convenience of an open-collector with the speed of a totem-pole output; it can be bus

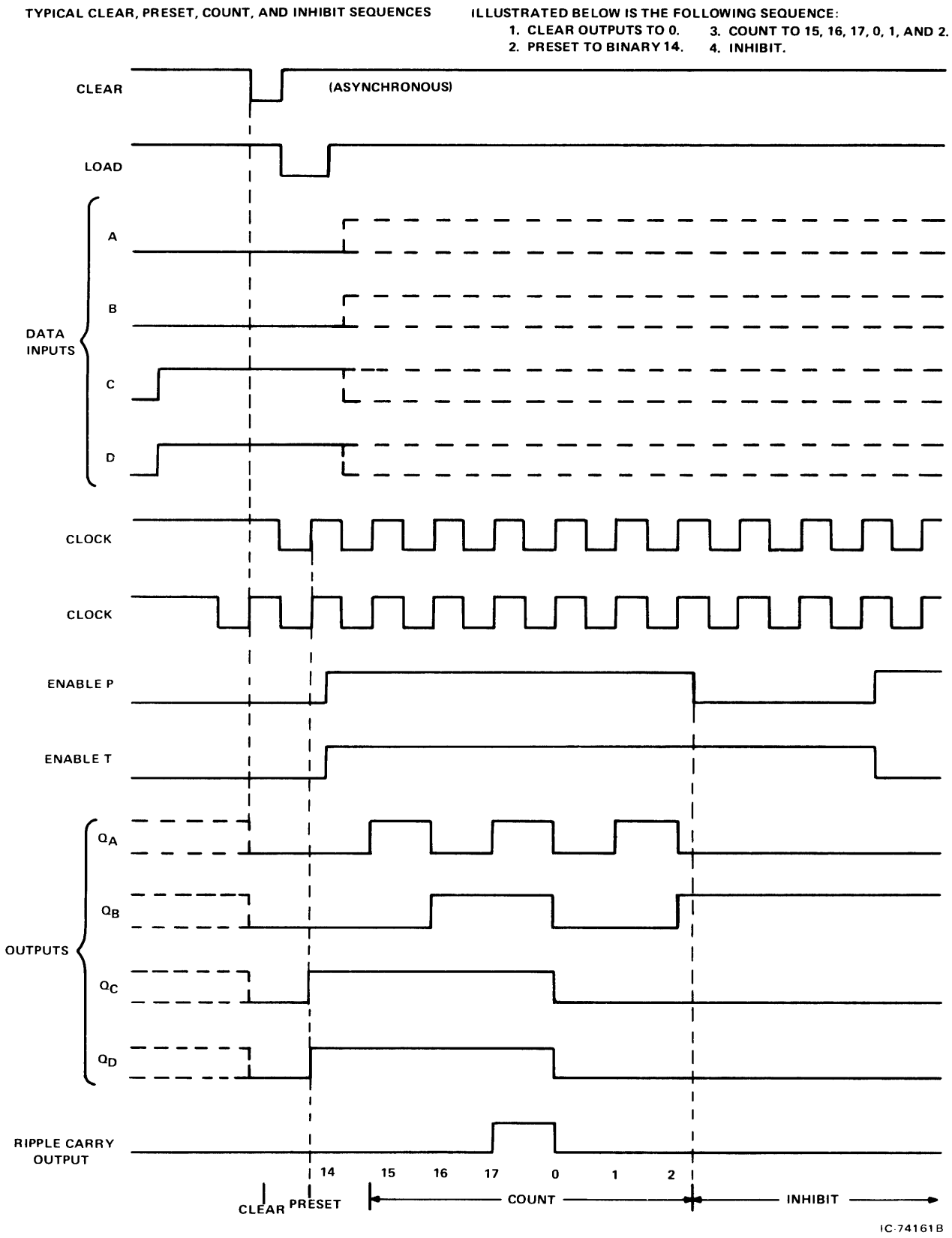



Figure 5-6 74161 Synchronous 4-Bit Counter

connected to other similar outputs, yet it retains the fast-rise-time characteristic of the TTL totem-pole output.

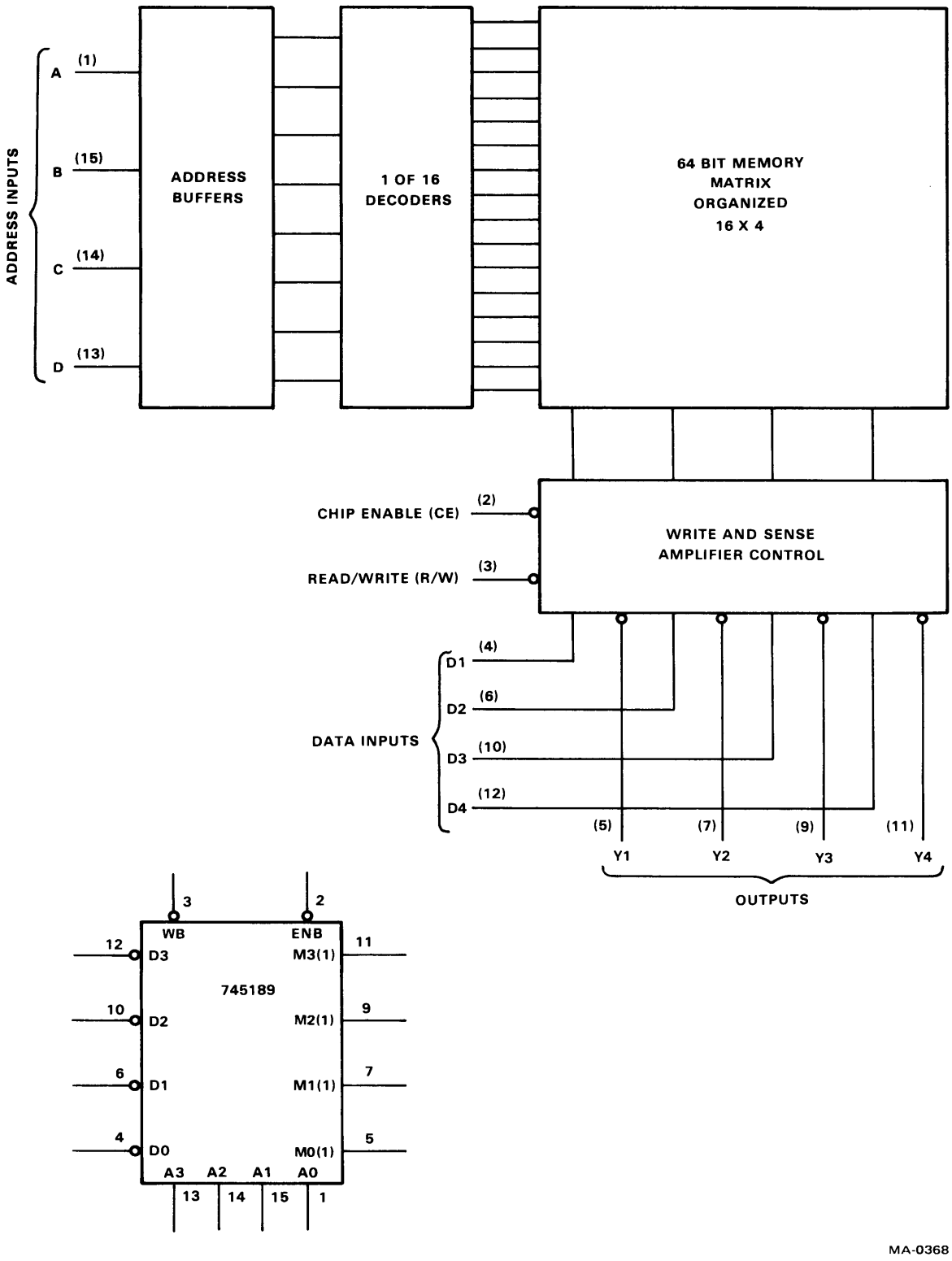


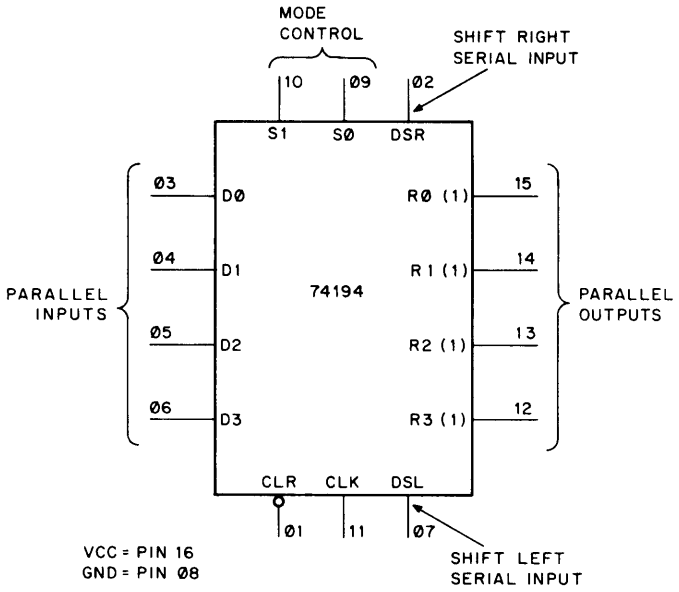

Figure 5-7 74189 64-Bit Random Access Memory

#### 5.3.1.2 74S194 4-BIT SHIFT REGISTER (Figure 5-8)

The 74S194 is a parallel load, parallel output, shift register with left shift and right shift capability. Clocking is accomplished by positive-edge triggering. In addition, the IC contains an inhibit function and direct overriding clear input.

## 74194 4-BIT SHIFT REGISTER

The 74194 is a parallel load, parallel output, shift register with left shift and right shift capability. Clocking is accomplished by positive-edge triggering. In addition, the IC contains an inhibit function and direct overriding clear input.




| | MODE CONTROL | |
|---|---|---|
| | S1 | S0 |
| PARALLEL LOAD | H | H |
| SHIFT RIGHT (IN THE DIRECTION R0 TOWARD R3) | L | H |
| SHIFT LEFT (IN THE DIRECTION R3 TOWARD R0) | H | L |
| INHIBIT CLOCK (DO NOTHING) | L | L |

Figure 5-8 74S194 4-Bit Shift Register

### 5.3.2 Removal and Replacement of ICs

The PDP-11/60 modules are multilayer etched circuit boards. The four layers consist of two (power and ground) internal planes, and two etched circuit external layers (Figure 5-9). The inner power and ground planes form a decoupling capacitor between the power and ground planes, providing shielding between the etched circuit layers and reducing the possibility of noise and/or cross-talk. One advantage in using this type of module construction is that the need to route power and ground signals to each individual component and IC via etching on the two outer etched boards is eliminated, allowing a much greater component density on each board.

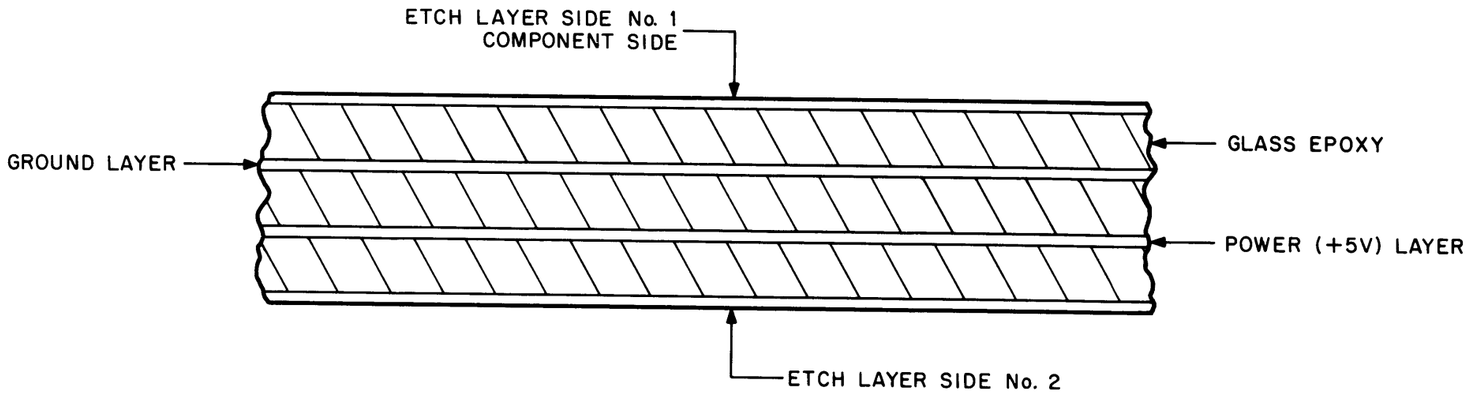



Figure 5-9 Cross Section of Multilayer Board

**5.3.2.1 Location of ICs** -- On the handle end of the board, the physical location of the last IC in each row is E-numbered to aid in locating ICs during maintenance and troubleshooting.

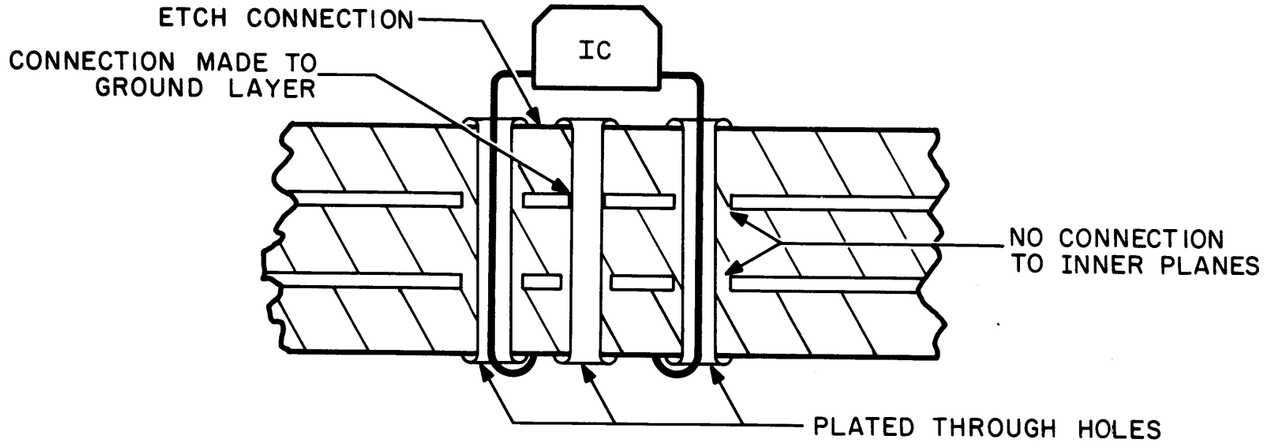
The first sheet of each module circuit schematic is a physical layout showing the location of the ICs and discrete components on that module.

When possible, some IC locations on most boards are not used; these spare locations, provided on a space available basis, ensure that if future ECOs (engineering change orders) involving additions are required, they can be implemented more easily.

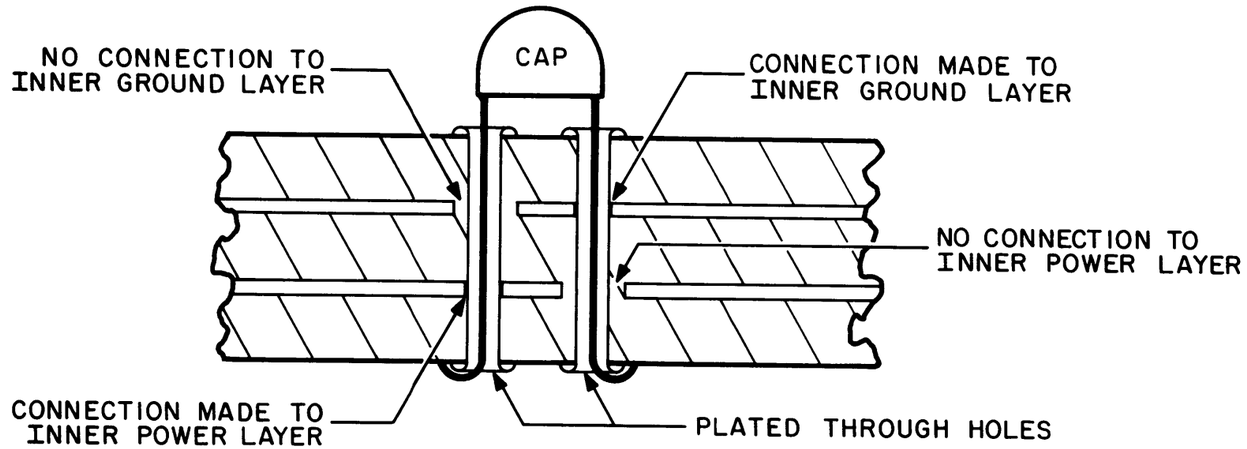
When spare locations are provided on the module, each spare location has a number just like one of the ICs. The spare locations must also be counted when locating ICs on the board. Thus, when an IC is added to a board (e.g., because of an ECO), the IC assumes the preassigned number of the location into which it is installed. Thus, the numbered IC locations at the handle end of the board, as well as all other IC locations, always remain the same.

**5.3.2.2 IC Connections** -- IC and component connections to the power and/or ground inner layers are normally made as shown in Figure 5-10A. The ICs and components are connected to the inner layers in this manner to allow the IC or component to be more easily replaced.

When a component is tied directly to an inner layer, as shown in Figure 5-10B, instead of connecting through an etch as shown in Figure 5-10A, it is difficult to remove the component because most of the heat from the soldering iron is absorbed by the inner layer, preventing the solder around the leads of the component or IC from melting. To minimize this difficulty, direct connections to the inner layer are made by a vein-type connection as shown in Figure 5-11. This type of connection reduces the connected area between the plated-through hole and the inner layer. This reduces the amount of heat transfer to the inner layer when heat is applied to the plated-through hole when melting solder and removing component leads, or removing excess solder once the lead has been removed.



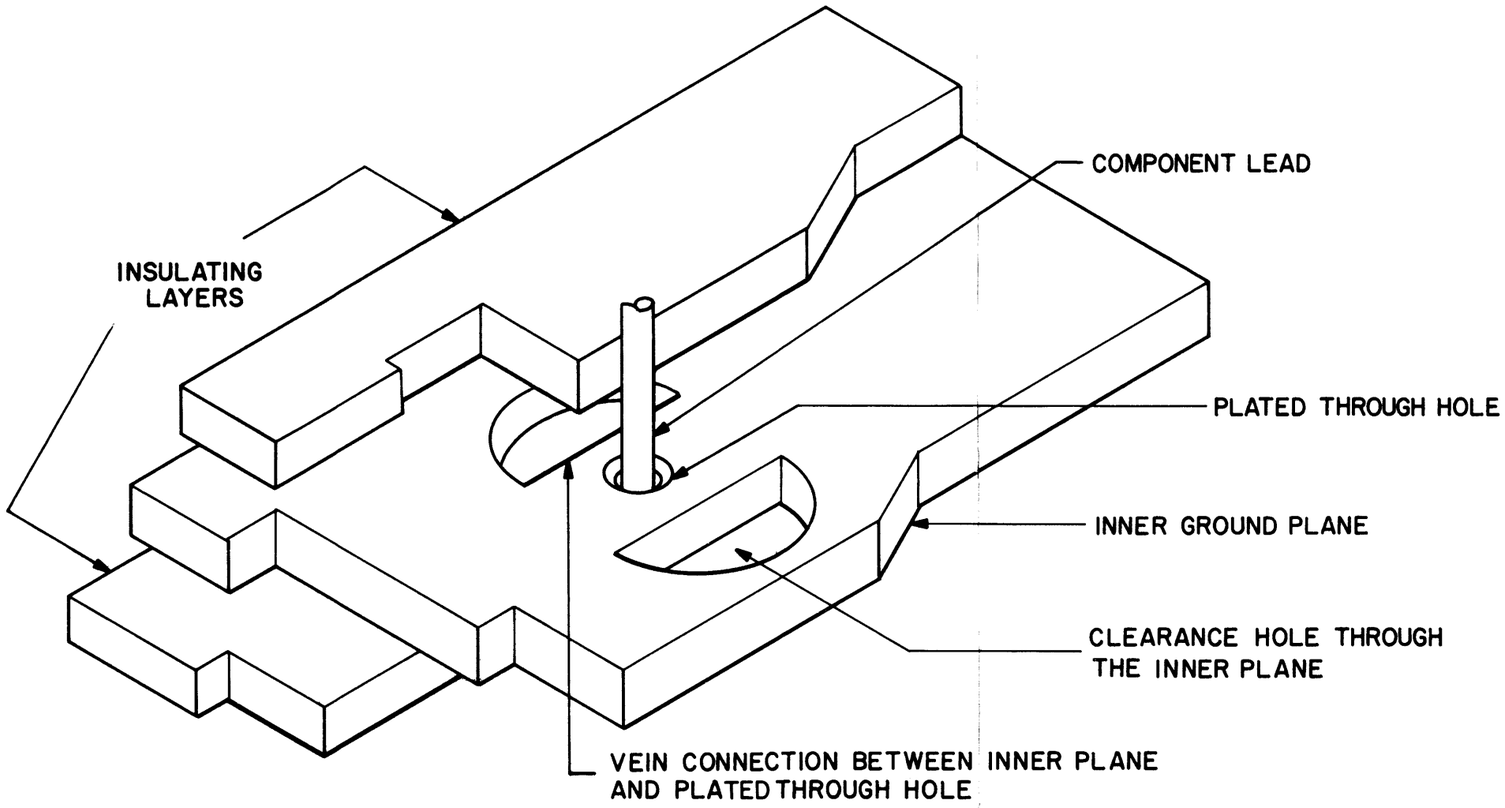

A. NORMAL COMPONENT CONNECTION TO INNER LAYER




B. COMPONENT CONNECTED DIRECTLY TO INNER LAYERS

11-0961

Figure 5-10 Component Connections to Inner Layer




II-2300

Figure 5-11 Top View of Component Connection Made Directly to Inner Layer

**5.3.2.3 IC and Component Removal and Replacement** -- Because the etch and plated-through hole eyelets are so small, extra care should be taken during the maintenance and repair of the multilayer modules, especially when soldering and unsoldering components. Certain tools (or their equivalent) are recommended for use during removal and replacement of ICs on the multilayer modules. The manufacturer and type of part number of each tool are indicated in the list below:

Small diagonal cutters, Utica No. 47-4

Soldering iron, Paragon No. 615

Pliers, Utica No. 23-4

**5.3.2.4 Removal and Replacement of Plastic Case ICs** -- To remove and replace a plastic case IC and to preclude damage to the multilayer board, the procedure described by Figures 5-12 through 5-18 should be strictly adhered to.

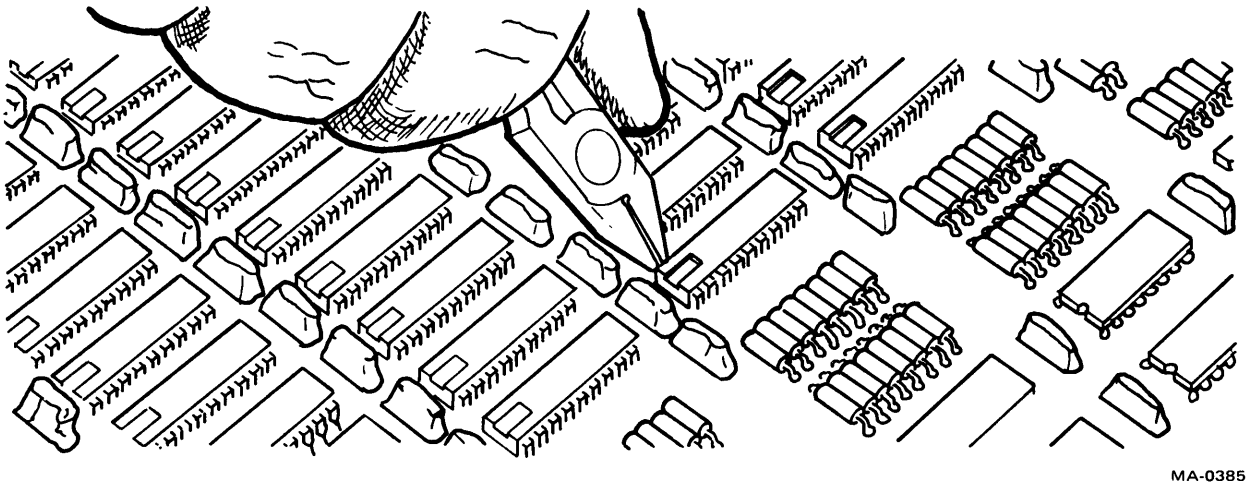



Figure 5-12 Removing a Defective IC From Module

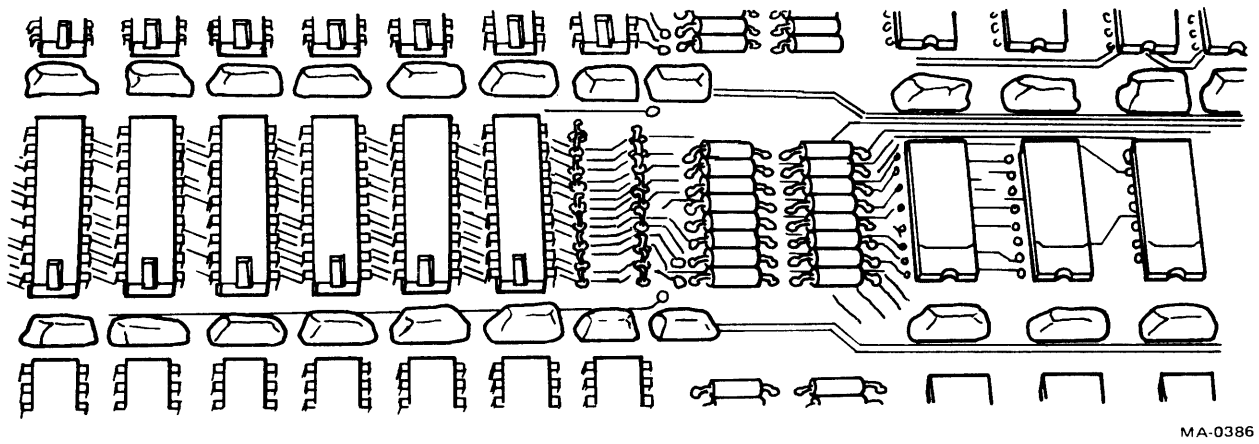



Figure 5-13 Defective IC Removed

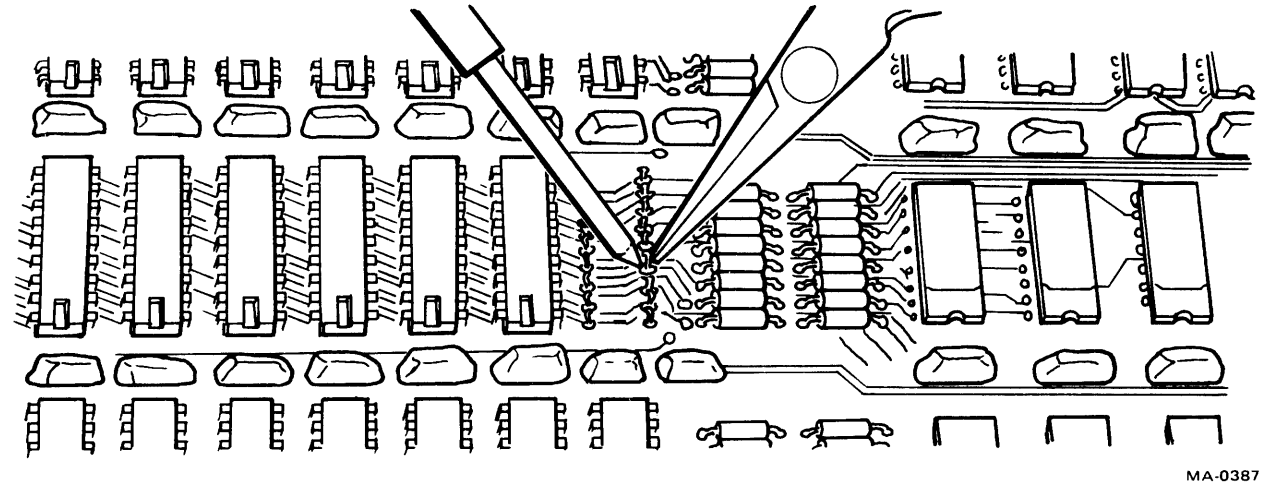



Figure 5-14 Removing IC Leads

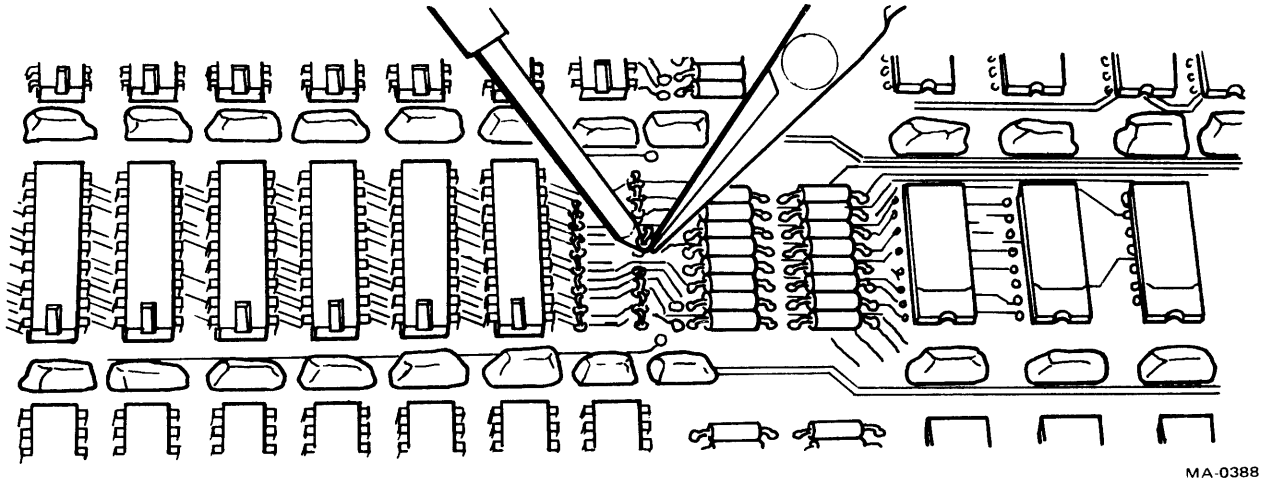



Figure 5-15 IC Lead Removed

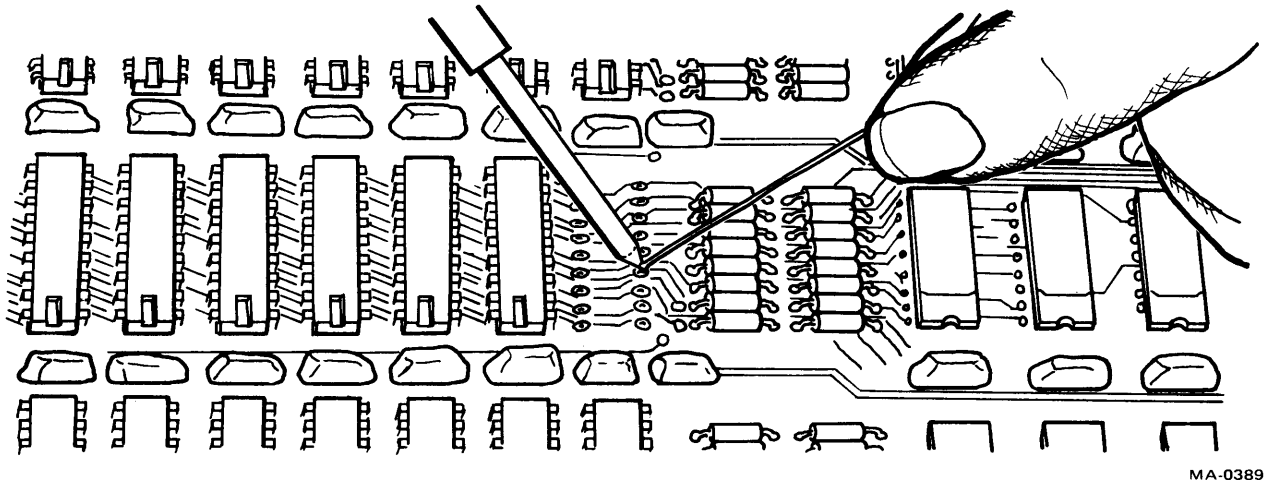



Figure 5-16 Applying Solder to Refill Eyelet

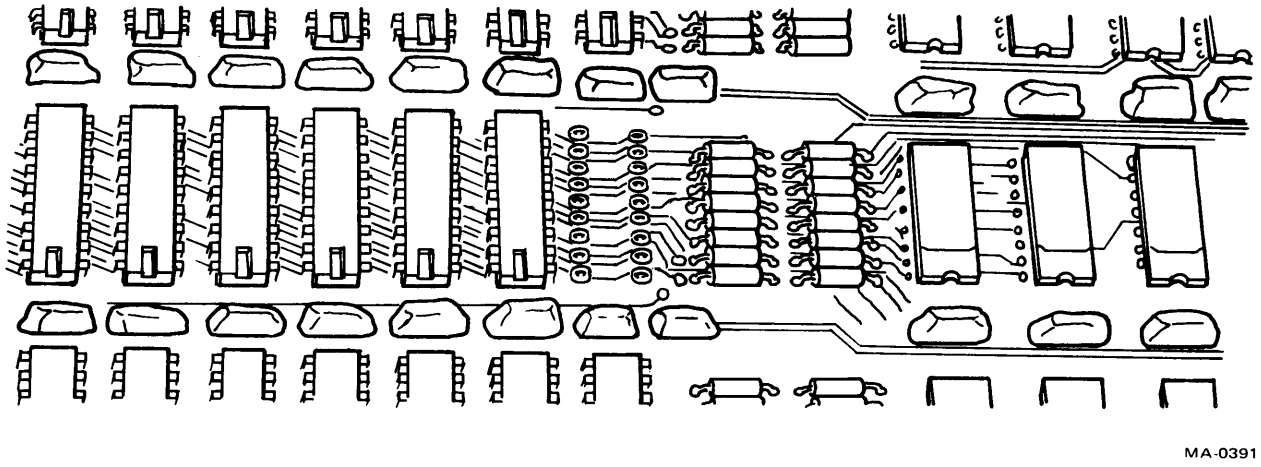



Figure 5-17 Removing Excess Solder From Eyelet

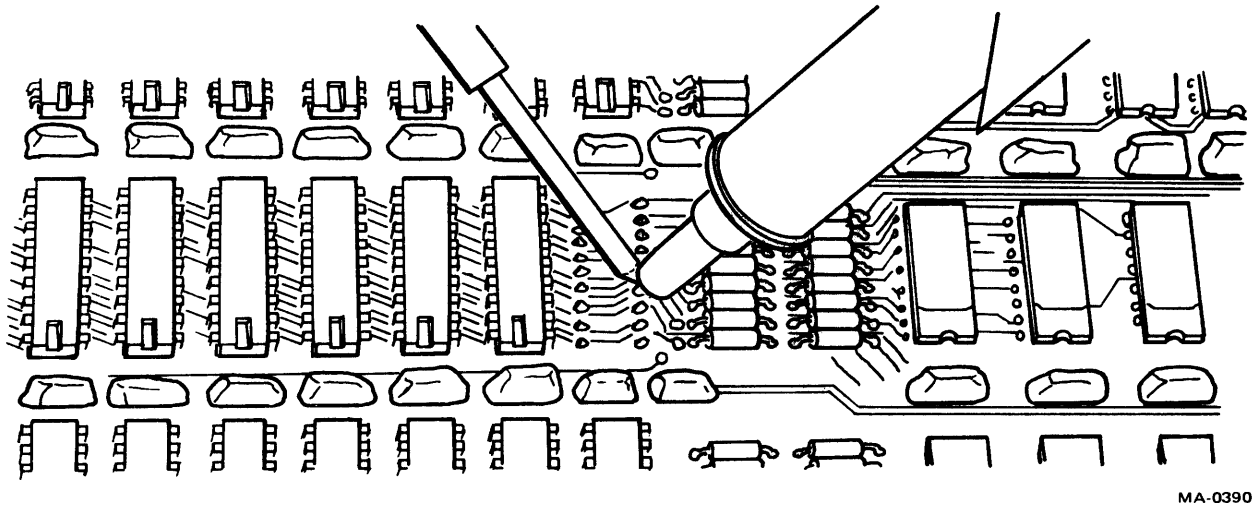



Figure 5-18 IC Location Ready For Insertion of New IC

Defective IC leads are clipped, (Figure 5-12) using small diagonal cutters (Utica, Part No. 47-4). Cut the leads as close to the body of the IC as possible to allow the remaining leads to be removed more easily.

IC location after the IC has been removed (Figure 5-13) with the IC leads still in the IC leads still in the board. Locate the leads of the IC just removed on the soldered (back) side of the board and cut all leads to avoid difficulty during their removal.

IC leads being removed from side 1 of the board, (Figure 5-14) Apply heat to the lead until the lead becomes loose. Then remove the lead by pinching with the soldering iron (Paragaon, Part No. 615) and pliers (Utica, Part No. 23-4).

**CAUTION**

**Leads that are connected to an inner layer require more heat because much of the heat is absorbed by the inner layer. It is helpful to add solder to the lead first causing more heat to be conducted to the solder in the eyelet around the lead.**

Lead directly after removal from the eyelet using the soldering iron and pliers, (Figure 5-15).

After all of the IC leads have been removed, (Figure 5-16) remove the excess solder remaining in the eyelets prior to inserting the new IC. This figure shows solder being applied to the eyelets after all the leads have been removed. The extra solder absorbs excess heat and keeps it from being applied directly to the etch of the plated-through holes.

Once the eyelets have been refilled with solder, as described in Figure 5-16, remove the solder using the soldering iron and solder extractor as shown in Figure 5-17. In this figure, the eyelet has no connection to the board inner layers; thus, the solder can be extracted from the same side of the module to which the heat is applied. However, in cases where direct connections to the inner layer are made, heat must be applied to one side of the module and the solder must be extracted from the opposite side due to the heat sinking properties of the inner layers. In this case, the module should be in a vertical position to allow access to both sides of the module simultaneously.

IC location after all the eyelets have been cleared of solder, (Figure 5-18).

1. Inspect the eyelets to ensure that no excess solder remains. If all the solder is not removed, refill the hole as described in Figure 5-16 and remove the solder again as described in Figure 5-17. Continue this procedure as required, until all of the eyelets are

cleared of excess solder.

2. Use a cleaning solvent and brush to clean the IC location of any excess solder flux.

3. Thoroughly inspect the IC location and surrounding area for solder splash and damage to etch lines and plated-through holes.

4. Ensure that none of the leads is bent, and insert the replacement IC in the holes. When inserting the replacement IC into place, avoid bending the leads on the opposite side of the module; this makes future removal of the IC easier, should it be necessary.

**CAUTION**

**If the leads must be bent to hold the IC in position for soldering, avoid bending the leads more than 45 degrees, using only one lead at each end and on opposite sides of the IC.**

5. Solder in the new IC from the opposite side of the module. Use enough solder to fill the holes and make a good connection. Avoid using an excess of solder to prevent overflow on the top side of the board, which could cause a short under the body of the IC.

6. Once all the solder connections are made, clean and inspect the area for any damage. Cut off IC leads close to the board. Take necessary corrective action for any defects that are found.

**CAUTION**

**After installing the ECO or replacing a faulty IC on a module, ensure that no short circuit exists between the power and ground planes of the module. Do this before replacing the module in the equipment.**