# VII. 32-BIT MICROPROCESSOR PODS

---

WARNINGS

**BEFORE PLUGGING IN THE ML4400 TO AN AC POWER SOURCE, VERIFY THAT THE CORRECT VOLTAGE (115 OR 230) HAS BEEN SELECTED VIA THE VOLTAGE SELECTOR SWITCH AT THE UPPER RIGHT OF THE REAR PANEL.**

**DO NOT INSERT OR REMOVE ANY CAPTURE MODULE, AND DO NOT CONNECT OR DISCONNECT ANY LOGIC POD OR MICROPROCESSOR POD FROM THE ML4400, WITHOUT FIRST POWERING DOWN THE ML4400 VIA THE POWER SWITCH AT THE LOWER RIGHT OF THE REAR PANEL. FAILURE TO DO SO MAY DAMAGE THE UNIT.**

---

## A. GENERAL (Applicable to all 32-bit microprocessor pods)

### 1. CONNECTIONS AND ADAPTOR POD

See Sections II-E and II-F, "Standard Capture Module and Microprocessor Pods" and "Microprocessor Pod and Microprocessor", regarding connections.

### 2. DISASSEMBLY

See Section IV-D.4, "Disassembly Display Mode", as well as various parts of this section about the particular Microprocessor Pod being used.

### 3. OPERATION

For operating details for each of Arium's 32-Bit Microprocessor Pods, see Section VII-B and following. (There is one section per Pod.)

## 4. CONFIGURATION AND SETUP

When the ML4400 is configured with a Microprocessor Pod instead of a Logic Pod, there are several differences in its Setup screens, as discussed below.

### a. Format Display Screen (via Format key)

The ML4400 will automatically configure itself at powerup for synchronous microprocessor cycle data collection. The default State format has three or four fields defined:

EXT, a binary field for status lines
STS, a second binary field, for status lines
ADR, address bus
DAT, data bus

The Data Inputs are labeled on the Format Display screen as probe numbers. For specific probe assignments for a particular 32-bit Microprocessor Pod, see the manual section for that Pod (Section VII-B and following, below).

### b. Clock Setup Screen (via CLOCK key)

The external clock signal is derived as required from the signals on each microprocessor. (External clock qualification is not used with Microprocessor Pods.)

An internal clock mode is also available with some pods, which permits sampling of microprocessor address, data and status lines at a rate independent of the microprocessor clocking signals.

### c. Trigger Words Setup Screen (via TRIGGER key)

When a Microprocessor Pod is used with the ML4400, two additional softkeys are available on the Trigger Words Setup screen:

The ROLL STATUS softkey speeds up editing the Status fields by rolling through common choices for Status values (fetch, read, write, etc.). Status mnemonics are displayed to the right of the Status field.

The STATUS BIT DEF'S softkey opens the Status Bit help window, which defines each bit in the Status field. Depressing the softkey a second time closes the window. For specific status values for a particular 32-bit Microprocessor Pod, see the manual section for that Pod (Sections VII-B and following).

### d. Status, State, and State Search Screens

In these screens, status mnemonics are displayed to the right of each Status field.

## B. 32M-682 MICROPROCESSOR POD (Supports Motorola 68020)

### 1. GENERAL

The probe assembly of the 32M-682 Microprocessor Pod inserts directly into the socket of a 68020 114-pin grid array (PGA), and includes a PGA socket to contain the target's 68020 chip. The pod collects data at microprocessor clock speeds of up to 33 MHz, or at internal clock speeds of up to 20 MHz, and formats the data for logic analysis and full program disassembly (including all exceptions and bus cycles) by the ML4400.

The Pod has two operating modes (Disassembly and Probe), and produces disassembled code in Disassembly mode only (using an external clock). (The Pod modes are designated onscreen by 68020DIS and 68020PRB.) Special hardware in the Pod samples the data once per microprocessor bus cycle (external clock mode) or once per ML4400 internal clock.

The Disassembly display details all microprocessor data cycles ("unscrambled" and put back together with their corresponding instructions), provides labels for readability, and marks all instructions as either "(usr)" for User or "(sup)" for Supervisor, depending upon how they are fetched. The State display shows data as it occurs on the address and data busses. Data may also be viewed on the Timing display.

If the microprocessor ceases operation (e.g., because of a HALT condition or a lack of DSACK or BERR, acknowledge during a bus cycle), several preceding bus cycles will remain stored in the pod. These cycles cannot be recorded or triggered upon until the microprocessor resumes operation.

The Pod may be run with either external or internal clocking. With an external clock, the Pod captures one 80-bit set of data for each bus cycle in the target system; the target microprocessor clock may be up to 33 MHz (11-MHz normal bus cycle rate). Use of the internal clock allows sampling of the 80 channels at a rate (of up to 20 MHz) independent of the target clock; this feature can be used to analyze hardware timing problems.

### 2. HARDWARE

#### a. Signal Assignments

Eighty signals are saved in each cycle of the Trace Buffer:

32 Data lines
32 Address lines
16 Discrete lines

The data is saved in the Trace Buffer noninverted. The probes are displayed in four groups (Address, Data, Status and External). Figures IV-A and IV-B, below, relate probe numbers to signals and describe the signals. (Note that Probe Numbers 79-76 carry External signals in the Probe mode, but Status signals in the Disassembly mode.)

Figure VII-A

SIGNAL ASSIGNMENTS FOR 32M-682 POD
(Asterisk indicates active low)

| Probe No. | 68020 Signal | | |
|---|---|---|---|
| | Probe Mode | Both Modes | Disassembly Mode |
| **External (EXT) -- (Field of 8 binary digits)** | | | |
| 39 | | Probe 4 | |
| 38 | | BGACK* | |
| 37 | | CDIS* | |
| 36 | | HALT* | |
| 35 | | BERR* | |
| 34 | | AVEC* | |
| 33 | | DSACK1* | |
| 32 | | DSACK0* | |
| **Status (STS) -- (Field of 8 binary digits)** | | | |
| 79 | Probe 3 | | CHIT |
| 78 | Probe 2 | | OCS* |
| 77 | Probe 1 | | SIZ1 |
| 76 | Probe 0 | | SIZ0 |
| 75 | | R/W* | |
| 74 | | FC2 | |
| 73 | | FC1 | |
| 72 | | FC0 | |
| **Address (ADR) -- (Field of 8 hexadecimal digits)** | | | |
| 71-68 | | A31-A28 | |
| 67-64 | | A27-A24 | |
| 63-60 | | A23-A20 | |
| 59-56 | | A19-A16 | |
| 55-52 | | A15-A12 | |
| 51-48 | | A11-A08 | |
| 47-44 | | A07-A04 | |
| 43-40 | | A03-A00 | |
| **Data (DAT) -- (Field of 8 hexadecimal digits)** | | | |
| 31-28 | | D31-D28 \ Data | |
| 27-24 | | D27-D24 / Byte 0 | |
| 23-20 | | D23-D20 \ Data | |
| 19-16 | | D19-D16 / Byte 1 | |
| 15-12 | | D15-D12 \ Data | |
| 11- 8 | | D11-D08 / Byte 2 | |
| 7- 4 | | D07-D04 \ Data | |
| 3- 0 | | D03-D00 / Byte 3 | |

Figure VII-B

68020 EXTERNAL AND STATUS SIGNALS
FOR 32M-682 POD

(Asterisk indicates active low)

Probe 79: CHIT, when high, indicates that the 68020 aborted a bus cycle due to an instruction cache hit. In this case, the address and status bits are meaningful, but data is not. (Note: Occasionally the microprocessor will abort a cycle, even with cache disabled; ignore these.)

Probe 78: OCS*, when low, indicates that the 68020 bus cycle is the first cycle of an operand transfer or instruction prefetch.

Probe 77-76: SIZ1 and SIZ0 are the coded bits by which the 68020 informs the external hardware as to how many bytes of data it needs (according to the table in Section _?_ (Figure _?_)) in the corresponding cycle.

Probe 75: R/W* is the READ/WRITE* signal from the processor; it is high during Read cycles and low during Write cycles.

Probes 74-72: FC2, FC1, and FC0 are the processor function codes which indicate which of several address spaces is accessed by a given cycle. (These "spaces" are listed Section VII-B.2.b, below, "Status Codes.")

Probe 38: BGACK* is the Bus Grant Acknowledge input to the processor. When low, cycles are performed by a device other than the processor.

Probe 37: CDIS -- This bit is high if the hardware (the target system or the ML4400) is forcing the 68020 instruction cache to be disabled. The cache may also be forced off in software.

Probe 36: HALT* -- See Probe 35, BERR*.

Probes 36-35: BERR* and HALT* signals are taken directly off the processor. When the BERR* signal is asserted without HALT*, the processor takes a bus error exception; when BERR* and HALT* are both asserted, the processor will retry the cycle. In the Filtered mode, cycles with both BERR* and HALT* asserted (cycles to be retried) are removed, as the cycle will recur.

Probe 34: AVEC*, when low, indicates that autovectoring is enabled for interrupt acknowledge cycles.

Probe 33-32: DSACK1* and DSACK0* are supplied by the target system, and signify the width of the external data bus for the corresponding bus cycle according to the table in Section VII-B.2.b, below). They also perform the data transfer acknowledge function (when either is low).

b. Status Codes

The EXT (External) and STS (Status) fields (on the default Trigger screen and State Display screen) contain status information about the bus cycle. The three least significant bits (Bits 74-72) of the STS field are the function codes FC2, FC1 and FC0 of the processor, which define the "space" of the cycle as shown:

| FC2 | FC1 | FC0 | |
|---|---|---|---|
| 0 | 0 | 1 | User Data space |
| 0 | 1 | 0 | User Program space |
| 1 | 0 | 1 | Supervisor Data space |
| 1 | 1 | 0 | Supervisor Program space |
| 1 | 1 | 1 | CPU space; used for Interrupt Acknowledge and Breakpoint cycles |

(Other combinations are unassigned and cannot be accessed in the 68000; the 68010 can access them via the MOVES instruction.)

Bits 33 and 32 (DSACK1* and DSACK0*) are coded as follows:

| DSACK1* | DSACK0* | |
|---|---|---|
| 1 | 1 | No response; insert wait states |
| 1 | 0 | 1-byte bus response |
| 0 | 1 | 2-byte (word) bus response |
| 0 | 0 | 4-byte (long-word) bus response |

Bits 77 and 76 (SIZ1 and SIZ0) are coded as follows:

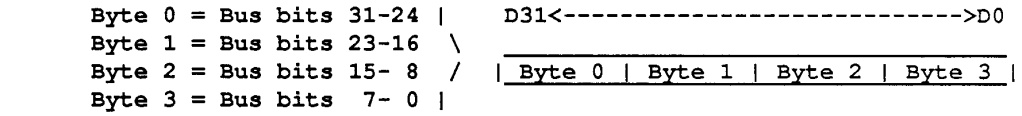
| SIZ1 | SIZ0 | |
|---|---|---|
| 0 | 1 | 1-byte data request |
| 1 | 0 | 2-byte (word) data request |
| 1 | 1 | 3-byte data request |
| 0 | 0 | 4-byte (long-word) data request |

The DSACK and size (SIZ1 and SIZ2) bits, in combination with A1 and A0, can be used to determine which of the data bytes is significant for the corresponding bus cycle. Figure VII-C shows which bytes are valid for each possible case.

Figure VII-C

68020 DATA BUS SIZE CODES AND BYTE VALIDITY

Data bus -- "First" byte location byte numbering:




| Size Req. (Bytes) | DSACK Resp. (Bytes) | Addr. Offset (Bytes) | No.of Bytes Valid | "1st" Byte Loc. | Size Req. (Bytes) | DSACK Resp. (Bytes) | Addr. Offset (Bytes) | No.of Bytes Valid | "1st" Byte Loc. |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 4 | 0 | 4 | 0 | 2 | 4 | 0 | 2 | 0 |
| 4 | 4 | 1 | 3 | 1 | 2 | 4 | 1 | 2 | 1 |
| 4 | 4 | 2 | 2 | 2 | 2 | 4 | 2 | 2 | 2 |
| 4 | 4 | 3 | 1 | 3 | 2 | 4 | 3 | 1 | 3 |
| 4 | 2 | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 0 |
| 4 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 |
| 4 | 2 | 2 | 2 | 0 | 2 | 2 | 2 | 2 | 0 |
| 4 | 2 | 3 | 1 | 1 | 2 | 2 | 3 | 1 | 1 |
| 4 | 1 | 0 | 1 | 0 | 2 | 1 | 0 | 1 | 0 |
| 4 | 1 | 1 | 1 | 0 | 2 | 1 | 1 | 1 | 0 |
| 4 | 1 | 2 | 1 | 0 | 2 | 1 | 2 | 1 | 0 |
| 4 | 1 | 3 | 1 | 0 | 2 | 1 | 3 | 1 | 0 |
| 3 | 4 | 0 | 3 | 0 | 1 | 4 | 0 | 1 | 0 |
| 3 | 4 | 1 | 3 | 1 | 1 | 4 | 1 | 1 | 1 |
| 3 | 4 | 2 | 2 | 2 | 1 | 4 | 2 | 1 | 2 |
| 3 | 4 | 3 | 1 | 3 | 1 | 4 | 3 | 1 | 3 |
| 3 | 2 | 0 | 2 | 0 | 1 | 2 | 0 | 1 | 0 |
| 3 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 |
| 3 | 2 | 2 | 2 | 0 | 1 | 2 | 2 | 1 | 0 |
| 3 | 2 | 3 | 1 | 1 | 1 | 2 | 3 | 1 | 1 |
| 3 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 3 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 3 | 1 | 2 | 1 | 0 | 1 | 1 | 2 | 1 | 0 |
| 3 | 1 | 3 | 1 | 0 | 1 | 1 | 3 | 1 | 0 |

The address offset in the above tables is coded as follows:

| A1 | A0 | Add. Offset |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 3 |

Figure VII-D

68020 STATUS CODE EXAMPLES
FOR 32M-682 POD

| | STS | EXT | A1 | A0 | Data Bits Used |
|---|---|---|---|---|---|
| Word write (attempt), user data space, no cache hit, byte device response: | 0X100001 | X1X11X10 | X | X | 31-24 |
| Longword read (attempt), user program space, no cache hit, byte device response: | 0X001010 | X1X11X10 | X | X | 31-24 |
| Byte write (attempt), user data space, no cache hit, word device response: | 0X010001 | X1X11X01 | X | X | 23-16 |
| Longword read (attempt), supervisor program space, first cycle prefetch, no cache hit, address offset 0, word device response: | 00001110 | X1X11X01 | 0 | 0 | 31-16 |
| Longword read (attempt), user program space, first cycle prefetch, cache hit (? = only addresses/status are valid): | 10001010 | X1X????? | 0 | 0 | Inval. |
| Longword write (attempt), supervisor data space, address offset 1, no cache hit, long-word device response: | 0X000101 | X1X11X00 | 0 | 1 | 23-0 |
| Interrupt acknowledge (*int. level on A3-A1, other addresses high), no cache hit, word response: | 0X001111 | X1X11101 | X | 1 | 31-16 |
| Autovectored interrupt acknowledge, no cache hit (see above): | 0X001111 | X1X110XX | X | 1 | Unused |
| Bus error cycle (may occur on any of the above processor cycle types): | XXXXXXXX | X1X10XXX | X | X | Inval. |
| Retry cycle (may occur on any of the above processor cycle types): | XXXXXXXX | X1X00XXX | X | X | Inval. |
| Bus grant (non-processor/DMA) cycle (can look like any of the processor cycles described): | XXXXXXXX | X0XXXXXX | X | X | Varies |

c. Instruction Cache

The 68020 is capable of caching up to 64 longwords of instructions in an internal memory, and executing directly from this memory when appropriate (it has no cache for data). When the cache is enabled and the desired instruction is in the cache (cache "hit"), the 68020 performs a truncated two-clock external bus cycle (signified by an ECS signal without a corresponding AS).

During this cycle, the address and status bits (FC2-FC0, R/W*, SIZ1-SIZ0) reflect the addressing of the desired instruction, but the data bus is not allowed time to become valid. The Pod (and the disassembler) thus cannot "see" the instructions corresponding to cache hits.

It is therefore necessary that the cache be disabled when using disassembly or performing cycle-by-cycle interpretation of instruction execution. This may be accomplished in any of three ways:

(1) Do not enable the cache in the target program. The 68020 defaults to a disabled cache on reset, or it can be disabled by writing to the cache control (CACR) register.

(2) Lower the cache disable (CDIS*) signal to the processor in the target system.

(3) Disable the cache from the ML4400, by setting "pod function" to "on" from the User Sequence screen. This signal is OR'ed with the target system's cache disable in the 68020 pod probe assembly.

d. Triggering Problems Due to Instruction and Data Alignment

(See Figure VII-E,"Trigger Setup Screen with 32M-682 Pod", below.)

Memories addressed by the 68020 processor may be 1, 2 or 4 bytes wide. Figure VII-C shows which data bytes are significant for all combinations of bus width, processor data size request, and address offset.

The disassembler uses a similar table to determine which bytes are significant and adapts accordingly. However, triggering schemes that assume a particular alignment of data or code will not work reliably.

Data items for 68020 programs have no alignment restrictions, other than those enforced by the user's software. A data item may start at any byte address, regardless of the size of the data item. This allows the possibility that a data item may be misaligned, even within a memory of the same width. A data item may have any of four alignments in a 4-byte-wide memory, or two alignments in a 2-byte-wide memory.

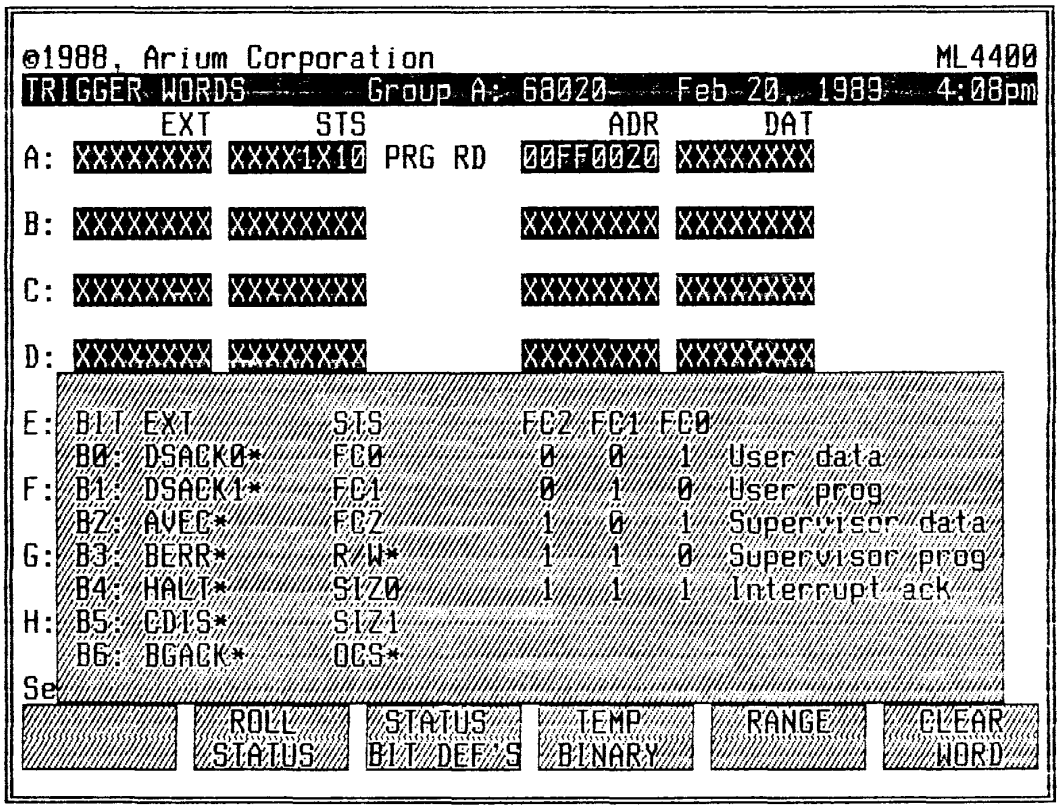
Fortunately, each data item is addressed by the first byte in the item, regardless of its position in the external bus. To simply detect an access to a data item, the trigger should be set at the item's starting address.

Instructions are aligned on even-byte (word) boundaries. Thus, a particular instruction word in a longword memory might be located in either half of the captured data field. However, prefetch cycles always access longword boundaries. To detect if an instruction starting at a particular address is prefetched, set the trigger address to a longword (low address bits = 00) boundary at or below one of the words in the instruction.

More complicated triggering, such as detecting a certain pattern of data or instruction in the program flow, requires setting a trigger for each of the possible alignments for that data and bus size. When possible, force the target software to align data items to boundaries of the same size. This maximizes execution speed (at some cost in memory size), and restricts misalignment to that of a smaller item within a larger field, a simpler case to handle.

Figure VII-E

TRIGGER SETUP SCREEN WITH 32M-682 POD



e. Probe Color Codes for External Input Lines

| Probe No. | Lead No. | Input Wire | Twisted with | Tip Color |
|---|---|---|---|---|
| 76 | 0 | Beige | Brown (inactive) | Black |
| 77 | 1 | Red | Beige (inactive) | Brown |
| 78 | 2 | Orange | Beige (inactive) | Red |
| 79 | 3 | Yellow | Beige (inactive) | Orange |
| 39 | 4 | Green | Beige (inactive) | Yellow |

## 3. DISASSEMBLY MODE

### a. General

Disassembly mode is a pod function mode (68020DIS) specifically configured to allow use of the Disassembly screen. However, any of the three screen display types (timing, state, and disassembly) may be used with this mode; these are described in Section VII-B. 5, "Data Display", below.

### b. Hardware Considerations

In disassembly mode (68020DIS), "Probe" bits 76-79 are connected to signals CHIT, OCS*, SIZ1 and SIZ0; the only actual physical probe line available to the ML4400 is Probe 4 (Bit 39). These four signals are required for the Disassembly display screen available in this mode.

## 4. PROBE MODE

In probe mode (68020PRB), "Probe" bits 76-79 are connected to the external probes numbered 0-3, which the user can connect to any signals of interest. However, signals CHIT, OCS*, SIZ1 and SIZ0 are no longer available to the ML4400. Because these signals are required for disassembly, the Disassembly display screen cannot be used in probe mode; Use the State and Timing screens only.

## 5. DATA DISPLAY

### a. General

The formatting and display of data is described generally for the ML4400 in Section IV, "Operation." Further details of data display when using 32-bit microprocessor pods, including typical State and Disassembly display screens, keyboard functions, and display formats, follow.

### b. Timing Display

The Timing display screen may be used with either pod mode (68020DIS or 68020PRB). Instructions and data are displayed in the same order in which they occurred on the bus.

c. State Display

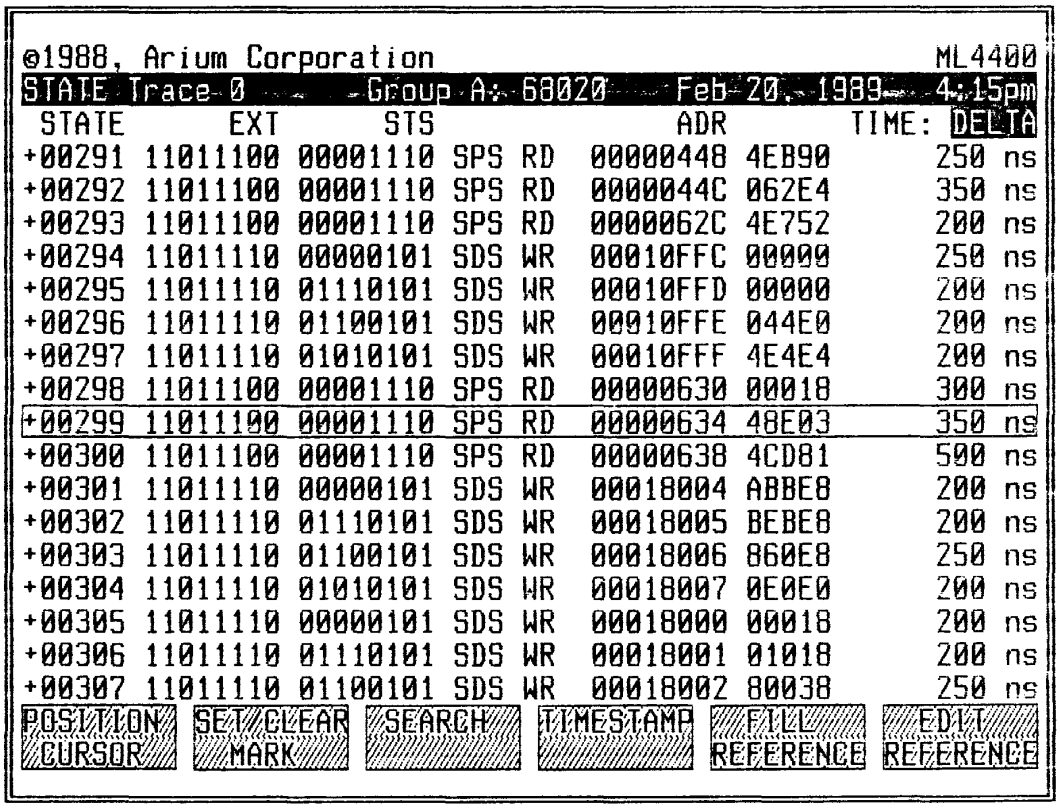
The State display screen may also be used with either pod mode. Figure VII-F shows the state display. Instructions and data are displayed in the same order in which they occurred on the bus.

Note the extra mnemonic field describing the type of cycle being displayed ("UDS RD", etc.); this information is derived from the FC2-FC0 and R/W* bits of the STS field.

Note also that, when TimeStamp display is enabled, a portion of the data field is obscured; disable TimeStamp to see all of the data field.

Figure VII-F

STATE DISPLAY SCREEN
WITH 32M-682 POD



d. Disassembly Display (See Figure VII-F)

(1) General

The Disassembly display screen should be used only when the Pod is in disassembly mode (68020DIS), and when using external clock. This combination provides the disassembler with all the information it requires, one sample per bus cycle. Figure VII-G shows the Disassembly display screen.

When the disassembler is "thinking", disassembly may be aborted at any time by depressing the STOP key.

(2) Program Flow Ordering of Display

Because of prefetching and the loose linkage between the bus control and execution units within the 68020, bus activity does not necessarily occur in the same sequence as program flow would dictate. For this reason, the disassembler rearranges the time order of bus events to correctly show the program flow. In particular, data cycles are shown immediately following the instructions which caused them.

(3) Instruction Display

As with 8-bit microprocessor pod disassembly, each instruction displayed contains a state number corresponding to the instruction position relative to the trigger event, the value of the Program Counter when the instruction was fetched (PC), and the mnemonic and operands for the instruction itself. Also, with the 68020 pod, a "(sup)" or "(usr)" at the far right of the screen indicates whether the instruction was fetched from Supervisor space or from User space, respectively.

(a) Synchronization

The disassembly software must "sync up" to the cycles in the Trace Buffer. The disassembly software finds the first program space read, assumes it is an opcode fetch, and checks the following cycles to see if it made a correct assumption. If the following cycles don't match the expected pattern of instruction fetches, data cycles and prefetches for that particular opcode value, then the software assumes it found a subsequent byte of an instruction or a program space operand read. It ignores this cycle, finds the next program space read, and repeats the process.

The cycles before synchronization occurs are ignored, as are any incomplete ones at the end of the trace. For any other trace cycles which it cannot interpret, the disassembler will display the message "UNABLE TO DISASSEMBLE nnn CYCLES".

(b) Unused Prefetches

The disassembler determines which bus cycles (prefetches) are actually executed and displays them. It ignores unused prefetches; if it is unable to determine whether a particular instruction was actually executed, a question mark will be prepended to the mnemonic on the display.

(4) Data Cycle Display

Data cycles are displayed on lines following the instruction that caused them, and may be turned on or off via the DATA CYCLES softkey.

Each data cycle contains an address, an indication of whether data was read from (>) or written to (<) the address, and the data itself. Each address consists of eight hexadecimal characters (32 bits), while data may be a byte (2 characters), a word (4 characters), or a long word (8 characters). There may also be a register name or symbolic code representing the source or destination of the data.

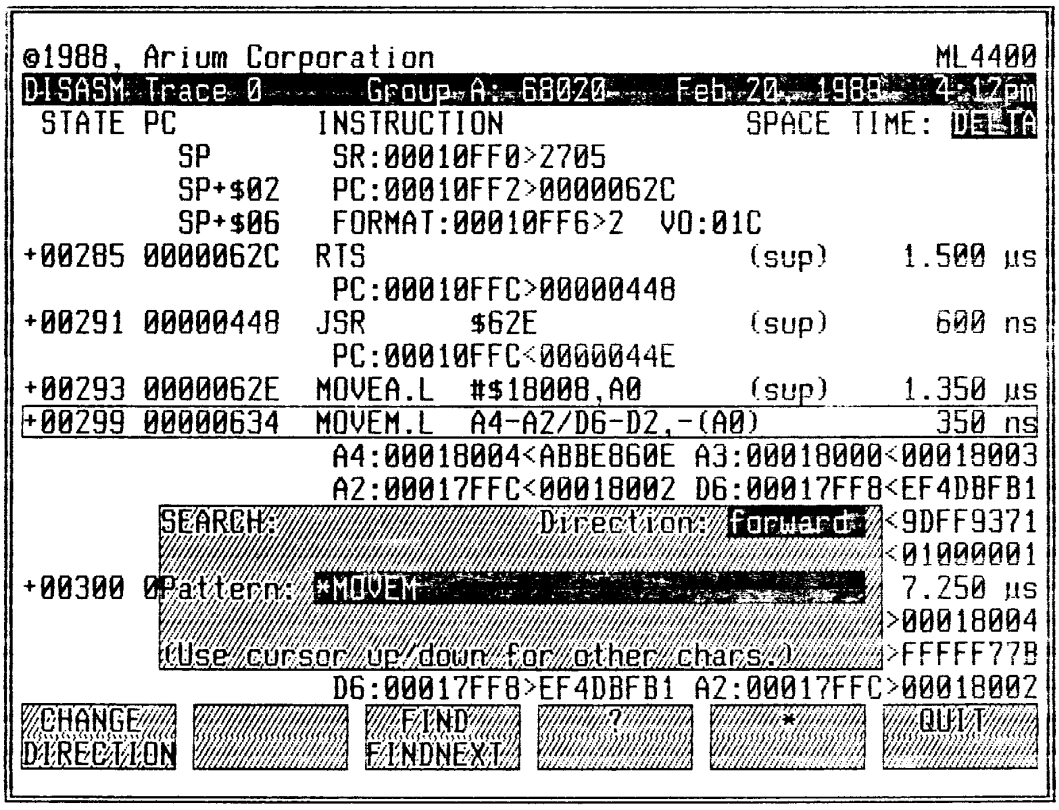
```
ADDRESS < DATA for WRITE cycles
ADDRESS > DATA for READ cycles
LABEL: data cycle for labeled data cycles
```

Even though the 68020 may perform word and longword accesses by doing two or more memory accesses, the ML4400 concatenates the data and shows it as as one data cycle on the Disassembly screen.

Data cycles corresponding to an exception are displayed in a stack frame format, sorted by order of increasing address and presented relative to the stack pointer. This may not correspond to the time order in which the data cycles occurred on the bus.

Figure VII-G

DISASSEMBLY DISPLAY SCREEN
WITH 32M-682 POD



(5) Specific Instruction Display Techniques

The first data cycle associated with all instructions using an indirect addressing mode is labeled "IMA:" (Indirect Memory Access).

MOVEM: The register list coded with the MOVEM instruction is reconstructed and displayed to the right of the instruction mnemonics.

BSR, JSR: The data cycle showing the value of the Program Counter pushed onto the stack is labeled "PC:".

RTD, RTS, RTR: The data cycle showing the value of the Program Counter popped from the stack is labeled "PC:".

CALLM, RTM: Data cycles representing the module stack frame follow the pattern given by the Motorola User's Manual Section D.1 (see Motorola's Figure D-3).

EXCEPTIONS: The first data cycle associated with an exception is the vector read and is labeled "VECTOR:". The following cycles match the pattern given in the Motorola User's Manual, Section 6.5.

RESET: Data cycles representing the vector reads of the Program Counter and Stack Pointer are labeled "PC" and "SP" accordingly.

INTERRUPTS: If the exception is an interrupt, then the type of interrupt (externally vectored, autovectored, spurious or uninitialized), interrupt priority level, and interrupt vector number are displayed and labelled as such.

EXCEPTION VECTOR: All exceptions label the exception vector, which is read as "VECTOR:".

(Blank)