II
THEORY OVERVIEW

The material in this section provides an overview description of the Central Processor Unit (CPU).

2.1 CPU HARDWARE OVERVIEW

Figure 2-1 is a major block diagram of the CPU, showing all of the major data transfer paths among the CPU major elements.

2.1.1 CPU Clock

The CPU clock generates the timing signals necessary for proper operation of the CPU. These clock signals are used to: (1) trigger all CPU registers and all test condition visible control flops, (2) stabilize registers after being loaded, and (3) signify that address, data, control, and parity are valid, allowing initiation of a Megabus cycle. The timing signals distributed throughout the CPU also provide four clock cycles that differ only in the duration of the cycle.

The clock is stalled early if needed Megabus data have not arrived and additional activity is to be performed in a cycle. It is stalled late to synchronize with a Megabus cycle if no activity is to be performed in the cycle.

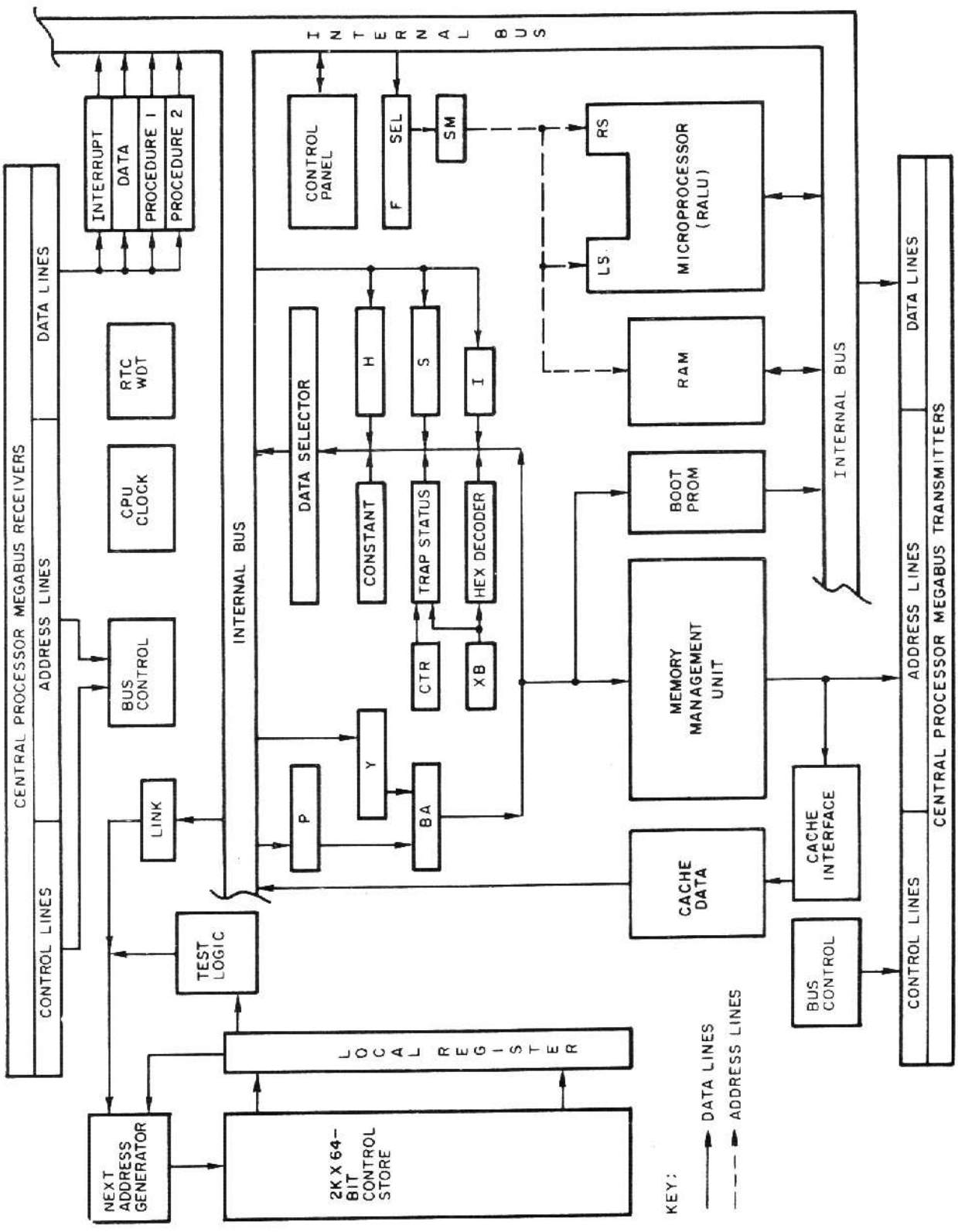


Figure 2-1 CPU Major Block Diagram

2.1.2 Control Store

Control store is comprised of a maximum of 2,048 by 64-bit locations (firmware words). These firmware words control almost all hardware operations within the CPU. Refer to subsection 3.1 for a detailed description of the control store (firmware) word.

All firmware words, as they are extracted from control store, are temporarily stored in a 72-bit register called the Control Store Local Register (CR).

2.1.3 Next Address Generation (NAG) Logic

The CPU uses one of the following three methods to generate the next firmware address:

- Method 1: Test and Branch
- Method 2: Major Branch
- Method 3: Subroutine Return.

Each of the above methods is a conditional branch based on a test condition selected by bits 42 through 47 of the control store word.

Method 1: This method is used when the next address(es) can be explicitly specified in bits 53 through 63 of the control store word.

Method 2: This method is used when branching to another firmware routine. The next address is conditionally obtained from the branch logic (refer to subsection 2.1.3.2) which generates numerous preassigned addresses.

Method 3: This method is used to conditionally return the firmware to the next control store location after execution of a firmware subroutine. The return address is obtained from the LINK register (refer to subsection 2.1.3.3), and must be stored in this register prior to entry to the specified subroutine.

2.1.3.1 Test Logic

The test logic selects 1 of 64 possible test conditions to participate in generating the next firmware address. Depending on whether the tested condition is true or false, the Test Condition True function is generated to control the NAG logic.

2.1.3.2 Branch Logic

The branch logic makes numerous preassigned addresses available for Major Branch operations. The address selected is determined by a decode of the instruction register contents, the control store outputs, and other control flops.

2.1.3.3 LINK Register (XL)

The XL register is an 8-bit register that is loaded from the internal bus and is available to the NAG logic.

2.1.4 Microprocessor

The microprocessor performs most of the arithmetic, logical, and shift operations required by the Level 6 system, including storage of operands for subsequent use by the firmware and over half the software visible registers. Included within this element are 17 storage registers, plus an arithmetic logic unit, that are used to implement the above functions. Of the 17 storage registers, 14 are software visible. The other three registers are visible only to the firmware, and are used as work registers for temporary storage of operands during firmware operations.

2.1.5 Microprocessor Addressing Logic

The Left Select (LS), Right Select (RS), and Selector Modifier (SM) logic areas comprise the microprocessor addressing logic. Although they are not physically part of the microprocessor, these logic areas perform the required register selection. The LS logic also provides addressing for the Random Access Memory (refer to subsection 2.1.9.1).

2.1.6 Internal Bus

The internal bus receives data from any one of several sources and makes this data available to destinations throughout the CPU.

Elements that function as internal bus sources include:

- Microprocessor output
- Sixteen 20-bit registers (RAM)
- Megabus buffer registers
- Other control registers (refer to subsection 2.1.7)
- Constant-generation facilities.

Elements that may serve as destinations for the internal bus include:

- Microprocessor input
- 16 RAM registers
- Memory address register and program counter
- Instruction register
- Other control registers (refer to subsection 2.1.7)
- Other control logic (e.g., Test logic).

In general, the internal bus receives inputs from a single source and optionally delivers copies to one destination in each of the categories listed above. Internal bus data are also available to the Megabus.

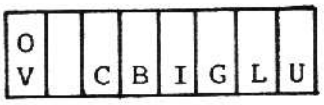
2.1.7 CPU Registers

The CPU registers, except those contained within the microprocessor, are described in the following subsections. Refer to subsections 4.4.2 and 4.4.3 for a description of the microprocessor registers.

2.1.7.1 Indicator Register (I)

The I register is an 8-bit software-visible register that contains various single bit indicators. The register format is as follows:

I Register



The indicators contained in this register can be grouped as follows:

- Arithmetic indicators
  - OV (overflow indicator)
  - C (carry bit)
- Bit indicator
  - (bit test indicator)
- I/O indicator
  - I (input/output indicator)
- Comparison indicators
  - G (greater than indicator)
  - L (less than indicator)
  - U (unequal signs indicator)

2.1.7.2 LINK Register (XL)

Refer to subsection 2.1.3.3 for a description of the XL register.

2.1.7.3 Counter Register (CTR)

The CTR register is a 4-bit counter that indicates the number of procedure words consumed in the processing of the current instruction. Its value is reported in the trap status Z-word; otherwise, it is not software-visible.

2.1.7.4 Select Register (SEL)

The SEL register is a 4-bit register (not visible to software) that can be loaded from the internal bus. SEL can also be decremented and tested for zero by the firmware, and is thus useful, for example, to maintain the number of loops for a Multiply or Divide operation.

2.1.7.5 Byte Indexing Register (XB)

The XB register is a 4-bit shift register (not visible to software) that supplies trap context information regarding indexing of bit or byte operations. The output from this register is fed to both the internal bus and the hexadecimal decoder logic.

2.1.7.6 Instruction Register (F)

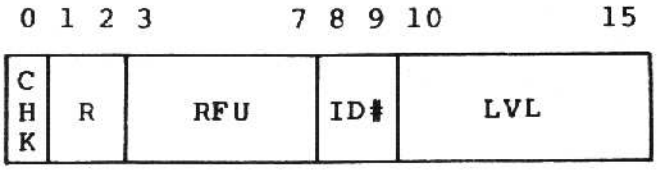
The F register is a 12-bit temporary storage register (not visible to software) that usually holds the most significant 12-bits of the instruction word as it is received from memory. Instruction words for the F register are received over the internal bus and loaded under firmware control.

2.1.7.7 H Register

The H register is a 16-bit register (not visible to software) that accepts data directly from the internal bus and makes these data available to the internal bus source selector for byte swap operations onto the internal bus (i.e., the least and most significant eight bits of the register are swapped as its contents are deposited onto the internal bus).

2.1.7.8 Status/Security Register (S)

The S register is a 16-bit software-visible register that contains the system status and CPU security keys. The format of this register is:



CHK - On if a system device did not pass its QLT

R - Ring Number

00 = Ring 3 } User
01 = Ring 2 }

10 = Ring 1 } Privileged
11 = Ring 0 }

RFU - Reserved for Future Use

ID# - CPU Identity Number (assigned during system configuration) provides the two least significant bits of the CPU channel number

LVL - Interrupt priority level 0 (highest) through 63 (lowest)

Instructions directed from a user ring at system resources (e.g., HLT, I/O, etc.) are not executed and cause a unique trap. Further, when the memory management unit is enabled, it scrutinizes each address to determine whether this user is permitted access to this location for the purpose intended (read, write, or execute).

All instructions directed from a privileged ring at system resources are permitted, subject to access rights checking.

2.1.7.9 P Register (Program Counter)

The P register is a 20-bit software-visible counter that is always incremented during instruction execution to point to the next procedure word.

2.1.7.10 Memory Address Register (Y)

The Y register is a 20-bit counter (not visible to software) that makes addresses available (via the address bus) to the Megabus, to the Memory Management Unit, and (via the internal bus) to the firmware.

2.1.7.11 Megabus Procedure Buffers (BP)

The Megabus procedure buffers, also known as the Procedure 1 (P1) and Procedure 2 (P2) registers, are 16-bit storage buffers

that receive procedure words requested from memory. Neither of these buffers is software visible.

2.1.7.12 Megabus Data Buffer (BD)

The Megabus data buffer is a 16-bit storage buffer (not visible to software) that receives non-procedural data from memory and/or I/O devices.

2.1.7.13 Megabus Interrupt Register (RUP)

The Megabus interrupt register is a 16-bit register (not visible to software) that receives the channel number and level number of an interrupting device. It stores this information from the time the CPU accepts the request until the CPU services the request.

2.1.7.14 Mode Registers

Refer to subsection 2.1.9.1 through 2.1.9.7 for a description of the mode registers.

2.1.7.15 M Collector Register

Refer to subsectgion 2.1.9.1.8 for a description of the M collector register.

2.1.8 CPU Control Flops and Control Signals

The CPU provides 16 control flops and five control signals. These elements serve as hardware controls that can be manipulated as directed by the firmware, and include:

CPU Control Flops

- SIGN
- MISC
- SHIN1
- SHIN2
- ZERO
- WRAP
- NEWXR
- ACK
- YELLOW
- PARER
- EXTRAP
- INTBSY
- TICK
- LOAD
- TRAFFIC
- PANOK

CPU Control Signals

- EFFRING
- NONPROC
- NOCHEK
- SEGERR
- PROV

2.1.9 Miscellaneous CPU Hardware

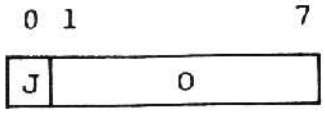
The following subsections describe those CPU elements not previously defined under one of the major CPU logic areas.

2.1.9.1 Random Access Memory (RAM)

The RAM consists of sixteen 20-bit locations. These locations contain data less frequently accessed than that stored in the microprocessor, including the seven software mode registers (M1 through M7). Addressing for the RAM is provided by the left select portion of the microprocessor addressing logic. Also included in this subsection is a description of the M collector register.

2.1.9.1.1 M1 Register

The M1 register is formatted to retain and provide the CPU trap masks as shown below:



J = Trace trap enable for jumps and branches.

- Zero = Trace Trap Disabled
- One = Trace Trap Enabled

01 through 07 = Overflow Trap - Enable controls for registers D1 through D7, respectively.

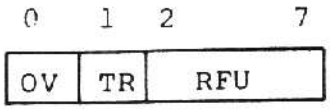
- Zero = Overflow Trap Disabled
- One = Overflow Trap Enabled

2.1.9.1.2 M2 Register

The M2 register is reserved for future use.

2.1.9.1.3 M3 Register

The M3 register is formatted as follows to retain the indicated CIP Trap Mask control (sent to the CIP for interpretaion).



OV = Overflow Trap Mask

- Zero = Trap Disabled
- One = Trap Enabled

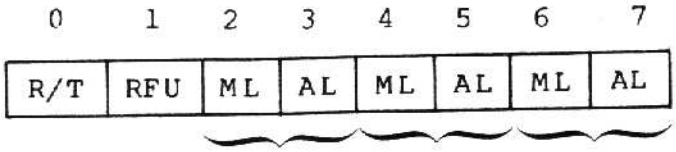
TR = Truncation Trap Mask

- Zero = Trap Disabled
- One = Trap Enabled

RFU = Reserved for Future Use (MBZ)

2.1.9.1.4 M4 Register

The M4 register is formatted as follows to retain the indicated SIP functions (sent to the SIP for interpretation).



R/T = Round/Truncate Mode

- Zero = Truncate
- One = Round

ML = Memory Operand Length in words

- Zero = 2 words (32 bits)
- One = 4 Words (64 bits)

AL = Accumulator Operand Length in Bits

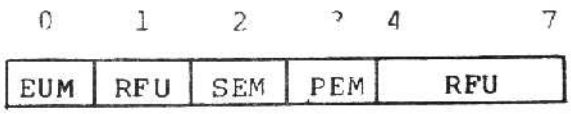
- Zero = 32-bit Operand
- One = 64-bit Operand

SA = Scientific Accumulator #1, #2 or #3.

RFU = Reserved for Future Use (MBZ).

2.1.9.1.5 M5 Register

The M5 register is formatted as follows to retain the indicated SIP functions (sent to the SIP for interpretation).



EUM = Exponent Underflow Trap Mask

- Zero = Trap Disabled
- One = Trap Enabled

SEM = Significance Error Trap Mask

- Zero = Trap Disabled
- One = Trap Enabled

PEM = Precision Error Trap Mask

- Zero = Trap Disable
- One = Trap Enabled

RFU = Reserved for Future Use (MBZ)

2.1.9.1.6 M6 Register

The M6 register is reserved for future use.

2.1.9.1.7 M7 Register

The M7 register is reserved for future use.

2.1.9.1.8 M Collector Register

Since the M register contents are stored in the RAM, they are not easily accessible to the test logic. The 8-bit M collector register is used to collect the pertinent bits that the CPU requies to make instantaneous decisions, and is formatted as follows:

```
M1(J) = Trace trap enable.
S1(D) = S1 memory operand length is quadruple-word.
S2(D) = S2 memory operand length is quadruple-word.
S3(D) = S3 memory operand length is quadruple-word.
      = Reserved for future use.
      = Reserved for future use.
      = The CIP is present and operational.
      = The SIP is present and operational.
```

2.1.9.2 Bootload PROM

The bootload PROM is comprised of 1,024, 16-bit locations that contain the standard bootload routines and internal test software for the CPU. It is automatically accessed in lieu of main memory whenever the Load mode flop on the control panel is On.

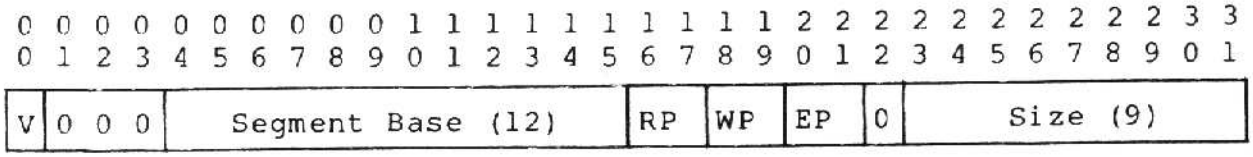
2.1.9.3 Address Bus

The address bus is a 20-bit wide bus that makes addresses for I/O and memory read or write cycles available to both the internal bus and the Megabus. It receives inputs from either the program counter or the memory address register; generally, the program counter is used for procedure references and the memory address register for all other references.

2.1.9.4 Memory Management Unit

When the Memory Management Unit (MMU) is installed, all addresses that reside in the internal processor registers (Y, B2, etc.) are reinterpreted before taking part in a memory reference (either through the Megabus or cache). Internal processor addresses are called virtual addresses; addresses after reinterpretation by the MMU are called physical addresses. Two steps are required to convert a virtual address into a physical address: (1) perform the virtual to physical mapping, and (2) determine, based on current processor states, whether this memory reference is permitted. The segmentation mechanism has been chosen to implement these requirements. Segments are sections of virtual memory space. Each segment is defined by a pattern of 32 bits stored in the MMU hardware.

2.1.9.4.1 Segment Descriptors



where:

V = Valid (this segment is currently valid)

Segment Base = Physical starting address of segment (in units of 256 words)

RP = Read Permisison (rings from which this segment is now readable)

WP = Write Permission (rings from which this segment is now writable)

EP = Execute Permission (rings from which this segment is now executable)

0 = Must be zero

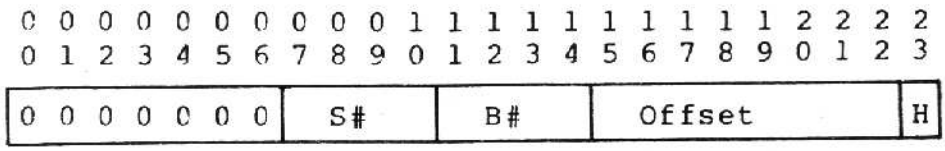
Size = Size of segment (in units of 256 words).

NOTE

A segment must begin on a 256-word boundary. The size field in the segment descriptor is in units of 256 words called a block.

2.1.9.4.2 Short Address Form (SAF) Virtual Address

When a virtual address is presented to the MMU, it is divided into three parts: a segment number, a block number, and an offset.



where:

S# = Segment number.

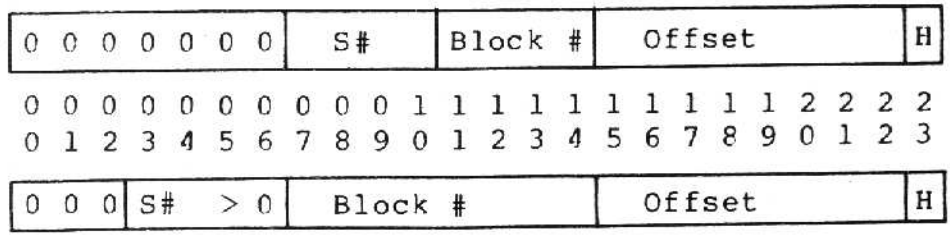
B# = Block number (maximum of 16 blocks of 256 words each).

Offset = These eight bits are never mapped.

H = Half-word address bit (never mapped).

2.1.9.4.3 Long Address Form (LAF) Virtual Addresses

LAF virtual addresses are also divided into three parts:



where:

S# = Segment number.

Block # = Maximum of 256 blocks of 256 words each (i.e., 64K).

Offset = These eight bits are never mapped.

H = Half-word address bit (never mapped).

2.1.9.4.4 Physical Address

The following steps take place when the MMU converts a virtual address into a physical address.

1. Use the segment number to fetch the segment descriptor.

2. Add the segment base (from the segment descriptor) to the block number (from the virtual address): the sum becomes bits 3 through 14 of the physical address.

3. Transmit the offset into bits 15 through 22 of the physical address.

4. Transmit the half-word bit 23 of the physical address.

2.1.9.4.5 Illegitimate Addresses

The following conditions cause a nonexistent system resource trap (TV15):

1. The validity bit in the segment descriptor is off.

2. The physical address is beyond the range of installed memory.

3. The block number in the virtual address exceeds the size field in the segment descriptor.

2.1.9.4.6 Access Rights

The MMU performs two types of checks each time it converts a virtual address into a physical address: (1) a comparison of the read/write/execute permission bits to the ring number in effect for this memory access, and (2) a determination as to whether this virtual address is legitimate (refer to subsection 2.1.9.4.5). Each of the three 2-bit fields (RP, WP, and EP) is coded to allow access to ring 0 only; 0 and 1; 0, 1, and 2; or all. The MMU (having knowledge of RP, WP, EP, the ring number, and the Megabus control lines that describe the intent of the processor regarding this memory cycle) may allow or disallow the memory reference. If the MMU disallows the memory reference, a protection violation results (TV14).

2.1.9.4.7 Activating the MMU

When Master Clear occurs, the MMU loads its segment descriptors so that no conversion takes place (trivial map). Thus, virtual and physical addresses are equal until the map is changed (by the ASD instruction or level change).

2.1.9.5 Cache Memory

The cache memory supplements main memory with a high-speed storage array for dedicated use by the CPU as a storage buffer for a copy of more frequently used CPU information. This effectively reduces access time, otherwise required by the CPU to fetch this information from main memory, and increases the overall speed and performance of the CPU. During communications

between the CPU and memory, the cache is effectively invisible to the system (i.e., if information requested by the CPU is not contained within the cache, a memory access will automatically be performed to obtain the required data).

2.1.9.6 Hexadecimal Decoder

The hexadecimal decoder consists of a 4-bit to 16-bit multiplexer that is used to generate a mask for bit and other operations. If the value of the 4-bit content of the XB register is $0 \leq N \leq 15$, then bit N of the decoder output is zero, and the other outputs are ones.

2.1.9.7 Subcommand Generator

The subcommand generator provides the control signals that permit firmware to manipulate the CPU hardware.

2.1.9.8 Constant Generators

The constant generators supply specific constants that are used by the CPU firmware.

2.1.9.9 Control Panels

Two types of control panels, full and basic, allow the operator to communicate with the CPU. The panels are available in one of the following hardware configurations.

- Bull Nose Configuration: The control panel is housed in a white molded housing, which projects from the CPU for panel visibility and operational ease.

- Industrial Configuration: The industrial configuration is mounted in a flat vertical dress panel, which is secured flush with the CPU outer exterior to allow the panel to be inside the cabinet.

- Remote Configuration: The remote configuration, using a repeater circuit board (refer to the Series 60 Level 6 Models 3X, 4X, and 5X Test and Verification Operators Guide, Order Number AW94), provides an extension of 10 cable feet between the CPU and the panel.

- Portable Configuration: The portable configuration is a self-contained full control panel, which is designed for exclusive use with the basic control panel.

2.1.9.9.1 Full Control Panel

The full control panel (see Figure 2-2) contains CPU controls, status indicators, and register displays, enabling the user to interrogate and analyze system performance. Table 2-1

describes the panel control and indicator functions. Table 2-2 lists the register selection codes. The logic design capabilities of the panel enable the user to arbitrarily:

- Stop the CPU
- Display and modify memory locations or CPU registers
- Single step through a program
- Load an operation or diagnostic program
- Manually store programs in memory
- Manually start programs.

The control panel also contains the facilities to accept diagnostic programs from an external tape cassette through the Honeywell Test and Verification Loader (TVL), which is designed specifically for this purpose. Panel coupling with this unit is accomplished by connecting the TVL signal cable to the ribbon connector at the lower right side of the control panel. It should be noted that the panel keypad array is automatically disabled while data is transferred from the cassette, to prevent erroneous alterations to its input data. For additional TVL technical and operational information and cassette specifications, refer to the Test and Verification Loader Manual (Order Number FL97).

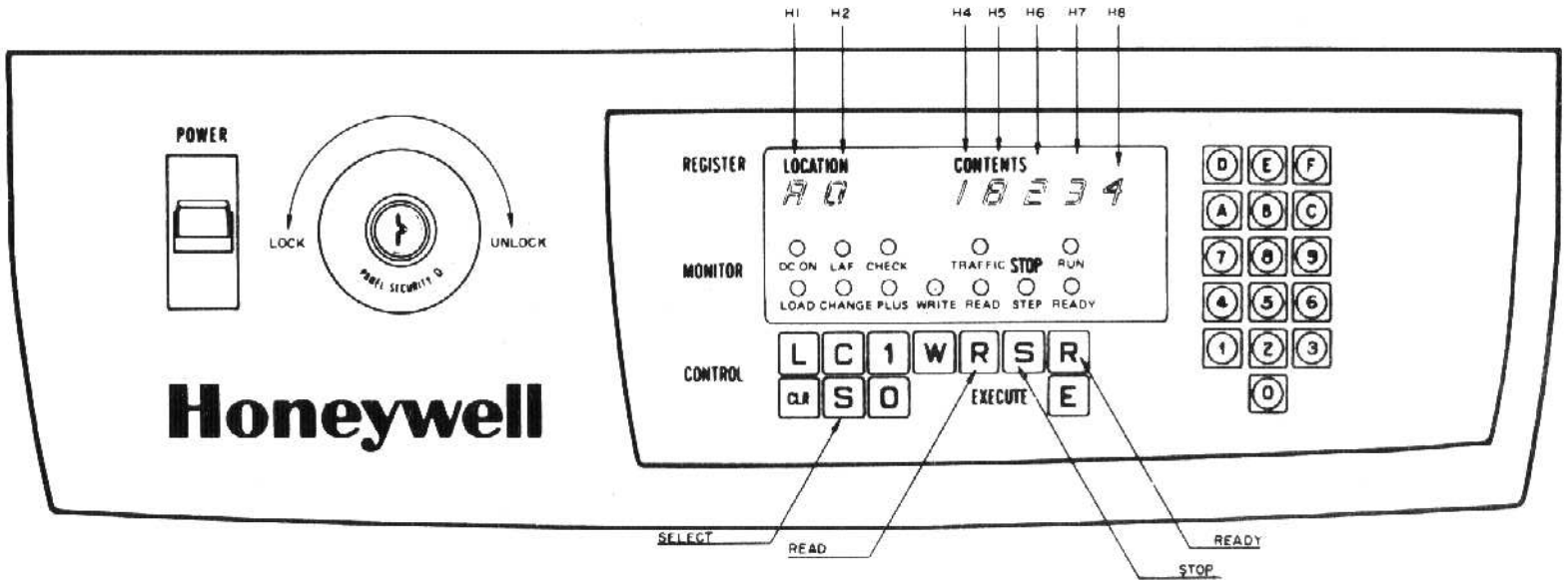


Figure 2-2 Full Control Panel

Table 2-1 Full Control Panel Switches and Indicators
(Sheet 1 of 6)

SWITCH/INDICATOR	FUNCTION
POWER Switch	Two-position switch used for engaging/disengaging the system power. Up position turns the power on; down position turns the power off. When the power is switched on, Master Clear is automatically activated and the DC ON indicator illuminates after DC power is attained.
PANEL SECURITY Switch	Two-position keylock switch used for enabling/disabling the control panel. Counterclockwise (locked) position disables all control panel switches/touch keys (except POWER) to prevent alteration of the memory or register contents; the register display is disabled (not illuminated). Clockwise (unlocked) position enables all control panel switches/touch keys; the register display illuminates, indicating that the control panel is operational.
0, 1, ... F Hexadecimal Pad Keys	Sets the appropriate digits into hexadecimal displays (H1, H2, H4 - H8).
CLR (Master Clear) Pushbutton	Activates Master Clear for the system, but is not effective while the CPU in the Run state. Depressing the CLR pushbutton: • Clears the Program Counter (E0), the H register, and the Instruction Register (D0). • Clears all pending interrupts and traps. • Stops the Real-Time Clock (RTC) and the Watchdog Timer (WDT). • Sets Status Register (S) bits 1 and 2 (ring number); clears bits 10 through 15 (interrupt priority level). • Creates trivial map in MMU, if present. • Starts the Quality Logic Test (QLT) in each controller. • Sets CPU address mode (LAF/SAF) from preselected switch setting.

Table 2-1 Full Control Panel Switches and Indicators
(Sheet 2 of 6)

SWITCH/INDICATOR	FUNCTION
CLR (Master Clear) Pushbutton (cont'd)	• Clears memory address (A0) and data (B0) registers.
Rocker Arm Switch Assembly	Located on the underside (component side) of the logic board and controls the following logical functions: • If switch 1 is enabled (On) and power fails, the CPU performs an auto boot operation (i.e., memory is volatile). If switch 1 is disabled and power fails, the CPU performs an auto restart operation (i.e., memory is non-volatile). • Switch 3 is set to the enabled position when used with the full control panel (i.e., the full control panel drives its own LAF indicator). Switch 3 is set to the disabled position when used with the portable panel (i.e., the basic panel drives the portable panel LAF indicator). • If switch 4 is enabled, depressing the CLEAR pushbutton places the CPU in Long Address Form (LAF). If switch 4 is disabled, depressing the CLEAR pushbutton places the CPU in Short Address Form (SAF).
L (Bootload) Pushbutton	Causes the control panel to enter the Load mode. When the Execute (E) pushbutton is depressed, the QLT routine is entered. After successful completion of the QLT, the TRAFFIC indicator extinguishes; the P Register contains 00002. Depressing the Execute (E) pushbutton at this time initiates the Bootload operation. The default load device is automatically chosen by the firmware unless an alternate input device is selected (prior to the second depression of Execute (E) pushbutton) by inserting its channel number into Data Register 1 (D1). To select D1, refer to Table 2-2.
S (Select) Pushbutton	Causes the control panel to enter the Select mode, thereby enabling register selection via the 16 key hexadecimal entry pad:

Table 2-1 Full Control Panel Switches and Indicators
(Sheet 3 of 6)

SWITCH/INDICATOR	FUNCTION
S (Select) Pushbutton (Cont'd)	• H1 displays keys 8 through F • H2 displays keys 0 through 7 • H4 through H8 display contents of register selected by H1-H2. The Select mode may be initiated in any state. NOTE Selection of any D (0 through 7) register causes H4 to extinguish; thus, 16 bits are displayed. When register B0 is selected, H4 is illuminated, but its value may usually be ignored.
1 (Plus One) Pushbutton	Causes the control panel to enter Plus One mode. Enables incrementing of the address register before reading or writing successive memory locations from the control panel if the control panel is in either Read or Write mode. Each depression of the Execute (E) pushbutton while in these modes causes the memory address register to be incremented by 1 prior to its use.
0 (Plus Zero) Pushbutton	Causes the control panel to exit Plus One mode; consequently, the memory address register is not incremented during the memory read or write operations.
W (Write) Pushbutton	Clears Plus One mode and places the control panel in Write mode. Clears Load mode. When the Execute (E) pushbutton is depressed in Write mode, the contents of the selected register are written into the Memory Address Register (A0).
R (Read) Pushbutton	Clears Plus One mode and places the control panel in Read mode. When the Execute (E) pushbutton is depressed, the contents of the memory location addressed by the Memory Address Register (A0) are read into the selected register. If in Load mode when the Execute (E) pushbutton is depressed in Read mode, the contents of the Bootload location addressed by the Memory Address

Table 2-1 Full Control Panel Switches and Indicators
(Sheet 4 of 6)

SWITCH/INDICATOR	FUNCTION
R (Read) Pushbutton (Cont'd)	Register (A0) are read into the selected register.
S (Stop) Pushbutton	Causes the control panel to enter Stop mode and halts instruction execution. Each depression of the Execute (E) pushbutton causes the CPU to execute one instruction.
R (Ready) Pushbutton	Causes the control panel to enter Ready mode. If the Execute (E) pushbutton is subsequently depressed, the control panel enters Run mode (RUN and TRAFFIC indicators illuminate) and program execution begins.
C (Change) Pushbutton	Causes the control panel to enter Change mode, thereby enabling entry of one to five hexadecimal digits into a selected register. Any hexadecimal key that is depressed is shifted into display position H8 and the corresponding position of the selected register. At the same time, H8 is shifted into H7, H7 into H6, H6 into H5, and H5 into H4; must not be in Run mode.
E (Execute) Pushbutton	Initiates control panel operation appropriate to the current mode: • If in Ready mode, depressing the Execute (E) pushbutton places the control panel in Run mode, executing instructions that start with the one in the instruction register and continue at the location specified by the program counter. Execution continues until a Stop (S), Read (R), or Write (W) pushbutton is depressed. • If in Step mode, depressing the Execute (E) pushbutton causes execution of one instruction; the control panel remains in Step mode. • If in Read or Write mode, depressing the Execute (E) pushbutton displays or changes the contents of the memory location selected by A0 (see also Plus One).

Table 2-1 Full Control Panel Switches and Indicators
(Sheet 5 of 6)

SWITCH/INDICATOR	FUNCTION
E (Execute) Pushbutton (Cont'd)	• If in Load mode, and in Ready or Stop mode, depressing the Execute (E) pushbutton initiates the QLT operation. For further details concerning the Execute (E) pushbutton, refer to the L, R, S, R, and W pushbutton descriptions.
H1, H2 Hexadecimal Display	Displays selected register number: H1 displays register types 8-F. H2 displays register numbers 0-7. Table 2-2 shows the H1, H2 format.
H4, H5, H6, H7, H8 Hexadecimal Display	Displays the contents of the selected register. Display is four or five hexadecimal digits, depending on register type: H4 is the most significant digit. H8 is the least significant digit.
DC ON Indicator	Illuminates when operational DC power is available in the system.
CHECK Indicator	Illuminates when at least one bus element has not successfully completed its logic tests (QLT) or a bus element is not properly plugged into the bus.
TRAFFIC Indicator	Illuminates when the CPU is executing any instruction, excluding Halt.
RUN Indicator	Illuminates when the control panel is in Run mode (i.e., executing programs). If the TRAFFIC indicator extinguishes while the RUN indicator is illuminated, the CPU is executing a Halt instruction.
LOAD Indicator	Illuminates when the control panel is in Load mode; extinguishes when the Load operation is successfully completed.

Table 2-1 Full Control Panel Switches and Indicators
(Sheet 6 of 6)

SWITCH/INDICATOR	FUNCTION
CHANGE Indicator	Illuminates when the control panel is in Change mode. In this mode, the contents of the selected register can be modified by key-in data from the hexadecimal pad keys (except in Run mode).
PLUS Indicator	Illuminates when the Plus One (1) key has been depressed. When PLUS is illuminated, sequential memory locations can be read or written. The PLUS indicator extinguishes when the Plus Zero (0), Read (R), or Write (W) key is depressed.
WRITE Indicator	Illuminates when the control panel is in Write mode. When WRITE is illuminated, data can be written into memory from the control panel.
READ Indicator	Illuminates when the control panel is in Read mode. When READ is illuminated, data can be read from memory via the control panel.
STOP/STEP Indicator	Illuminates when the control panel is in Step mode. One instruction is executed with each depression of the Execute (E) pushbutton.
READY Indicator	Illuminated when the Ready (R) key is depressed, placing the control panel in Ready mode. If the Execute (E) pushbutton is depressed, the control panel enters Run mode.
LAF Indicator	Illuminates when the CPU is set to execute instructions in LAF (long address form). In this mode, pointers in memory are 32 bits in length. Extinguishes when the CPU is set to execute instructions in SAF (short address form). In this mode, pointers in memory are 16 bits in length. NOTE The CPU QLT routine always executes in LAF mode and with the LAF indicator ON, regardless of the LAF/SAF switch setting.

Table 2-2 Control Panel Register Selection
(Sheet 1 of 2)

LOCATION CODE		SELECTED REGISTER CONTENTS					
H1	H2	H3	H4	H5	H6	H7	H8
8	0**		0	0	0	CIP Indicators	
8	1**		0	0	0	SIP Indicators	
8	2**		0	0	0	CIP Indicators	
8	3**		0	0	0	SIP Indicators	
8	4**		0	SIP Accumulator 1, Word 0			
8	5**		0	SIP Accumulator 1, Word 1			
8	6**		0	SIP Accumulator 1, Word 2			
8	7**		0	SIP Accumulator 1, Word 3			
9	0**		0	SIP Accumulator 2, Word 0			
9	1**		0	SIP Accumulator 2, Word 1			
9	2**		0	SIP Accumulator 2, Word 2			
9	3**		0	SIP Accumulator 2, Word 3			
9	4**		0	SIP Accumulator 3, Word 0			
9	5**		0	SIP Accumulator 3, Word 1			
9	6**		0	SIP Accumulator 3, Word 2			
9	7**	R	0	SIP Accumulator 3, Word 3			
A	0		Memory Address (Y)				
A	1**	F	Physical Address equivalent to A0				
A	2**		Stack Pointer				
A	3**	U	Remote Descriptor Base				
A	4**		Latest CIP Instruction Address				
A	5**		Latest SIP Instruction Address				
A	6**		Work Location				
A	7**		Work Location				
B	0		Memory Data				
B	1		Base (B1)				
B	2		Base (B2)				
B	3		Base (B3)				
B	4		Base (B4)				
B	5		Base (B5)				
B	6		Base (B6)				
B	7		Base (B7)				
C	0**		0	Status (S)			
C	1**		0	Indicators (I)		Mode (M1)	
C	2**		0	(RFU)		Mode (M2)	
C	3**		0	Indicators (I)		Mode (M3)	

Table 2-2 Control Panel Register Selection
(Sheet 2 of 2)

LOCATION CODE		SELECTED REGISTER CONTENTS			
H1	H2	H3	H4	H5 H6	H7 H8
C	4**	R F U	0	Real Time Clock (RTC)*	Mode (M4)
C	5**		0	(RFU)	Mode (M5)
C	6**		0	Watch Dog Timer (WDT)*	Mode (M6)
C	7**		0	Multi NATSAP / RFU	Mode (M7)
D	0		***	Instruction (D0)	
D	1			Data (D1)	
D	2			Data (D2)	
D	3			Data (D3)	
D	4			Data (D4)	
D	5			Data (D5)	
D	6			Data (D6)	
D	7			Data (D7)	
E	0			Program Counter (P)	

*WDT/RTC value (displayed in HEX indicators 4 and 5) equal 00_{16} if disabled and equal FF_{16} if enabled.
**Read (Display) only.
***D register selection causes HEX indicator 4 to extinguish.

2.1.9.9.2 Basic Control Panel

The operation of the basic control panel and its system display capabilities are confined to two switchable control functions and five system status indicators (as shown in Figure 2-3). A brief functional description of each switch and indicator is provided in Table 2-3. The basic control panel is designed primarily to provide an operator with the minimum number of controls required to activate a system. However, a portable full panel assembly is available for expansion to a full panel status.

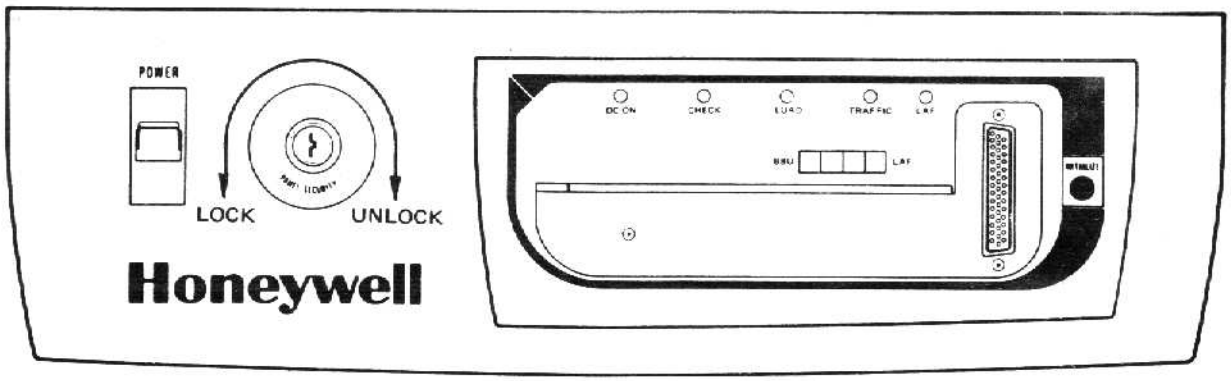


Figure 2-3 Basic Control Panel

Table 2-3 Basic Control Panel Switches and Indicators
(Sheet 1 of 3)

SWITCH/INDICATOR	FUNCTION
POWER Switch	Two-position switch used for engaging/ disengaging the system power. Up position turns the power off. When the power is switched on, Master Clear is automatically activated and the DC ON indicator illuminates after DC power is attained.
PANEL SECURITY Switch	Two-position keylock switch used for enabling/disabling the control panel. Counterclockwise (locked) position disables the INITIALIZE switch and the interface to full control panel.
INITIALIZE Pushbutton	Depressing the INITIALIZE pushbutton: • Clears the Program Counter (E0), the M registers, and the Instruction Register (D0). • Clears all pending interrupts and traps. • Stops the Real-Time Clock (RTC) and the Watchdog Timer (WDT). • Sets Status Register (S) bits 1 and 2 (ring number); clears bits 10 through 15 (interrupt priority level). • Creates trivial map in MMU, if present. • Starts the Quality Logic Test (QLT) in each controller. • Sets CPU address mode (LAF/SAF) from preselected switch setting.

Table 2-3 Basic Control Panel Switches and Indicators
(Sheet 2 of 3)

SWITCH/INDICATOR	FUNCTION
INITIALIZE Pushbutton (Cont'd)	• Clears memory address (A0) and data (B0) registers. • If memory is volatile, runs the CPU QLT, followed by a memory test on the first 8K of memory (checks memory and zeros data), then executes an auto boot. • If memory is non-volatile, runs the CPU QLT, then starts program execution with the instruction beginning at main memory location 00000.
Rocker Arm Switch Assembly	Controls the following logical functions: • If switch 1 is enabled (On) when power is applied, the CPU performs an auto boot operation (i.e., memory is volatile). If switch 1 is disabled when power is applied, the CPU performs an auto restart operation (i.e., memory is non-volatile). • If switch 4 is enabled, depressing the INITIALIZE pushbutton places the CPU in Long Address Form (LAF). If switch 4 is disabled, depressing the INITIALIZE pushbutton places the CPU in Short Address Form (SAF).
DC ON Indicator	Illuminates when operational DC power is available in the system.
CHECK Indicator	Illuminates when at least one bus element has not successfully completed its logic test (QLT) or a bus element is not properly plugged into the bus.
TRAFFIC Indicator	Illuminates when the CPU is executing any instruction, excluding Halt.
LOAD Indicator	Illuminates when the CPU is in Load mode; extinguishes when the Load operation is successfully completed.

Table 2-3 Basic Control Panel Switches and Indicators
(Sheet 3 of 3)

SWITCH/INDICATOR	FUNCTION
LAF Indicator	Illuminates when the CPU is set to execute instructions in LAF (Long Address Form). In this mode, pointers in memory are 32 bits in length. Extinguishes when the CPU is set to execute instructions in SAF (Short Address Form). In this mode, pointers in memory are 16 bits in length. NOTE The CPU QLT routine always executes in LAF mode and with the LAF indicator On, regardless of the LAF/SAF switch setting.

2.1.9.9.3 Portable Control Panel

The portable control panel, a self-contained full panel, is mounted on the basic panel and connects to the system through the 50-pin connector on the basic panel. Functionally, the portable panel is similar in operation to the full panel with two exceptions. Rocker arm switch assembly 1 is set to the auto restart position (memory is non-volatile), and rocker arm switch assembly 3 is set to the SAF position (Off), allowing the basic panel to drive the LAF indicator on the portable panel.

NOTE

With the basic panel PANEL SECURITY switch unlocked the portable panel functions exactly as if a full panel was configured. This includes execution of the CPU QLT for all available memory rather than just the first 8K which would be accomplished when only the basic panel is installed in a system.

2.2 MEGABUS OPERATIONS

The Megabus (see Figure 2-4) provides a common communication path (interface) among all units of the Level 6 system. The Megabus is asynchronous in design, permitting units of varying speeds to operate efficiently on the same system. Five types of communication are permitted over the Megabus: (1) memory read requests, (2) non-memory read requests, (3) read responses, (4) memory write requests, and (5) non-memory write requests, including interrupts.

The Megabus can accommodate a maximum of 23 units/controllers. The number of I/O devices supported by a single Megabus may be greater than 23 because several I/O devices may be connected through a single controller. For larger systems,

several Megabuses may be interconnected by using the Intersystem Links (ISLs).

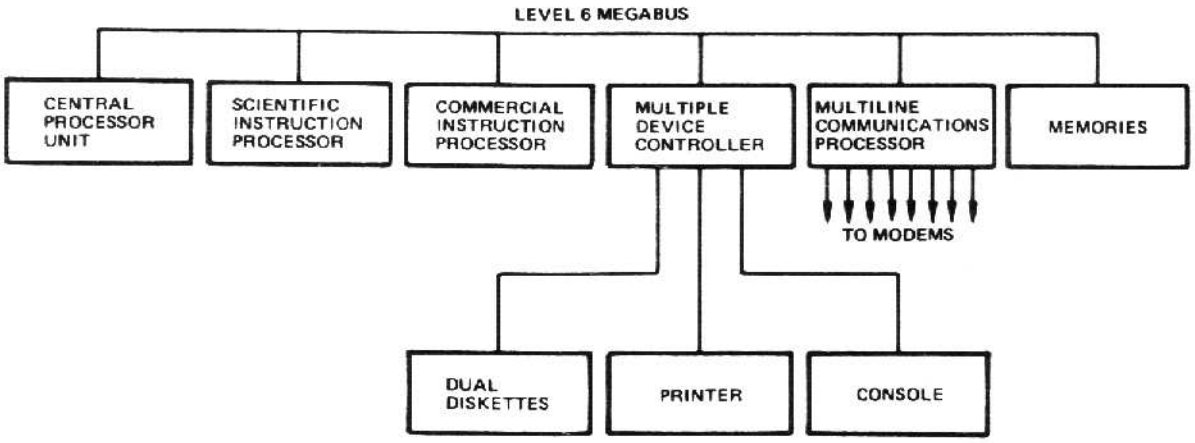


Figure 2-4 Typical Level 6 System

2.2.1 Master/Slave Relationship

The Megabus is bidirectional, thereby permitting any two units/controllers to communicate with each other at a given time. The transfer of information between units/controllers forms a temporary master/slave relationship (i.e., the unit/controller requesting and receiving access to the bus becomes the master unit while the unit/controller being addressed by the master unit becomes the slave unit). If the communication requires a reponse, the responding slave unit/controller assumes the role of master unit and the requesting unit/controller (previous master) becomes the slave unit.

All information transfers are from master unit to slave unit and each transfer is referred to as a bus cycle. This cycle is comprised of the following: (1) the requestor (master unit) asks for use of the bus, (2) if no other unit/controller of a higher priority is making a bus request, use of the bus is granted to the requestor (master unit), and (3) the master unit then transmits its information to the slave unit and the slave unit acknowledges or refuses the communication.

Communication between a master unit and slave unit requires a response from the slave unit when the master unit is requesting data (e.g., a memory read command). In this case, the request for information requires one bus cycle and the transmission of information back to the requestor requires an additional bus cycle to complete the task.

2.2.2 Megabus Usage

Common types of Megabus operations are listed in Table 2-4. These various operations require either one, two, or three bus cycles. Information transfers that are considered write operations require one bus cycle while transfers that are considered read operations require an additional bus cycle for the response.

NOTE

Once a bus cycle is granted, the types of operations performed between the master unit and the slave unit are a function of the specific functionality of the two units/controllers.

Other types of Megabus operations, such as controller-to-controller transfers, are not listed in Table 2-4; however, the bus architecture makes no restrictions in this regard.

Table 2-4 Common Types of Megabus Operations

TYPE OF OPERATION	ORIGINAL MASTER	ORIGINAL SLAVE	NUMBER OF BUS CYCLES
Instruction Fetch (one word)	CPU	Memory	2
Instruction Fetch (two words)	CPU	Memory	3
Operand Fetch (one word)	CPU	Memory	2
Operand Fetch (two words)	CPU	Memory	3
Operand Store (word)	CPU	Memory	1
Operand Store (byte)	CPU	Memory	1
DMA Read (word)	Controller	Memory	2
DMA Read (byte)	Controller	Memory	2
DMA Write (word)	Controller	Memory	1
DMA Write (byte)	Controller	Memory	1
I/O Output Command Word	CPU	Controller	1
I/O Input Command Word	CPU	Controller	2
Interrupt	Controller	CPU	1

2.3 MEGABUS FORMATS

Figure 2-5 depicts the information format for the data and address lines during various Megabus cycle operations. Each format shown reflects the occurrence of a single Megabus cycle. The following subsections provide a description of the formatting associated with the address and data lines.

OPERATION	NO. OF CYCLES	MASTER	SLAVE	ADDRESS LINES (BSAD)	DATA LINES (BSDT)
MEMORY READ REQUEST	1	CPU+CU	MEM	0 23: BYTE ADDRESS	0 9 10 15: MASTER CHANNEL NUMBER | VARIABLE USAGE
I/O READ REQUEST	1	CPU	CU	0 7 8 17 18 23: SLAVE CHANNEL NUMBER | FUNCTION CODE	0 9 10 15: MASTER CHANNEL NUMBER | VARIABLE USAGE
READ RESPONSE	1	MEM+CU	CPU+CU	0 7 8 17 18 23: SLAVE CHANNEL NUMBER | VARIABLE USAGE	0 15: DATA
MEMORY WRITE	1	CPU+CU	MEM	0 23: BYTE ADDRESS	0 7 8 15: DATA | DATA
I/O DATA OUTPUT	1	CPU	CU	0 7 8 17 18 23: SLAVE CHANNEL NUMBER | FUNCTION CODE	0 15: DATA
I/O ADDRESS OUTPUT	1	CPU	CU	0 7 8 17 18 23: SLAVE CHANNEL NUMBER | FUNCTION CODE	0 15: BYTE ADDRESS
INTERRUPT	1	CU	CPU	0 7 8 17 18 23: SLAVE CHANNEL NUMBER | MBZ	0 9 10 15: MASTER CHANNEL NUMBER | INTERRUPTING LEVEL

Figure 2-5 Megabus Formats

2.3.1 Channel Numbers

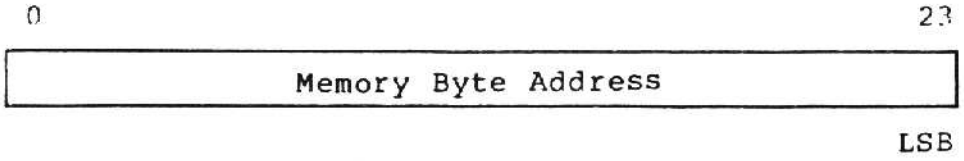
A channel number will exist for every end point in a particular system except for memory, which is identified only by a memory address.

The channel number of the slave unit will appear on the address bus for all non-memory transfers. Each unit will compare that number with its own internally stored number. The unit which achieves a comparison is, by definition, the slave and must respond to that cycle. No two end points on a single Megabus will be assigned the same channel number.

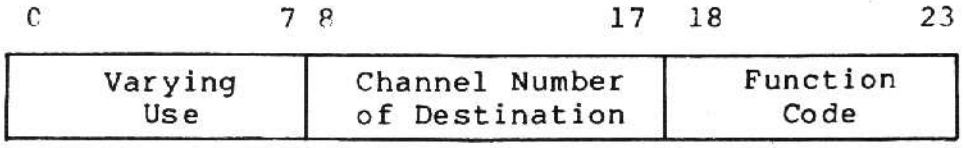
Processor channel numbers are restricted to the range of 0000 through $03C0_{16}$. The six upper bits of the channel number are fixed as Zeros by the processor logic and only the lower two bits are variable via a DIP switch. Processor channel numbers are not used by any other units.

2.3.2 Unit Addressing

A master unit may address any other unit on the bus as the slave unit. It does this by placing the slave address on the address lines. There are 24 address lines, which can have either of two interpretations, depending on the state of the Memory Reference signal. If Memory Reference is true, the following format applies to the address lines:

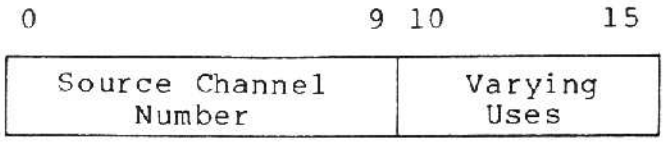


If Memory Reference is false, the following format applies to the address lines:



When units are passing control information, data, or interrupts, they address each other by channel number. Along with the channel number, a 6-bit function code is passed that specifies which function this transfer implies.

When a master unit requires a response from the slave unit, it indicates this to the slave unit by forcing the Bus Write signal false. In addition, the master unit provides its own identity to the slave unit by means of a channel number. This is coded on the data lines of the bus as follows:



The response cycle is directed to the requesting (master) unit by a non-memory reference transfer. The Second Half Bus Cycle accompanies the transfer to indicate that this is the awaited cycle.

2.3.3 Memory Read (Word)

During a read operation, a 16-bit data word is requested from memory. The CPU accepts the complete data word into one of the bus input buffers, from where it is routed (under firmware control) to the internal bus for use, as appropriate.

2.3.4 Memory Write (Word or Byte)

If during a write operation the byte control line is false, a 16-bit data word is written into main memory. However, if the byte control signal is true, the memory unit is alerted to enable an 8-bit write mask for either the left or right byte, depending on the state of address bit 23. If bit 23 is true, the right byte is written into memory as new data; if bit 23 is false, the left byte is written into memory.

2.4 CPU FIRMWARE OVERVIEW

Figure 2-6 is an overview flow chart of the 15 major firmware subdivisions:

- Initialize
- Instruction Fetch (XF)
- WDT/RTC Servicing (TIX)
- Control Panel Service
- Address Formation (XA)
- Branch Instructions (XA)
- Generic Instructions (XA)
- Address Formation (XB)
- Indirect and/or Indexing (XR)
- Operand Read (XR)
- Instruction Execution (XE)

- Result Write (XW)
- Trap
- Interrupt
- Quality Logic Test (QLT).

It must be understood that the above subdivisions are only for convenience of discussion. Most firmware routines are parts of several subdivisions.

2.4.1 Initialize

The initialize fimrware is entered by Master Clear and performs the following:

- Clears P and D0
- Clears various control flops
- Clears RAM
- Loads a trivial map into the Memory Management Unit
- Determines whether a SIP and/or CIP is present.

If the control panel is locked, the Instruction Fetch firmware is entered; otherwise, the Control Panel Service firmware is entered.

2.4.2 Instruction Fetch (XF)

The XF firmware obtains the first word of the next instruction for execution. Once the instruction is received from memory, it is loaded into the F register. Copies of the instruction are written into D0 and into RAM0. During XF, checks are performed for the following conditions:

- Device interrupt
- External processor trap
- WDT/RTC/Control panel service.

2.4.3 WDT/RTC Servicing (TIX)

Every 8-1/3 milliseconds (independent of line frequency), the watchdog timer and real-time clock are updated, if enabled. Once these tasks are performed, the Control Panel Service firmware is entered.

2.4.4 Control Panel Service

Every 8-1/3 milliseconds the control panel interface is interrogated to determine whether the operator wishes a new display or wishes to stop program execution. The display is then updated.

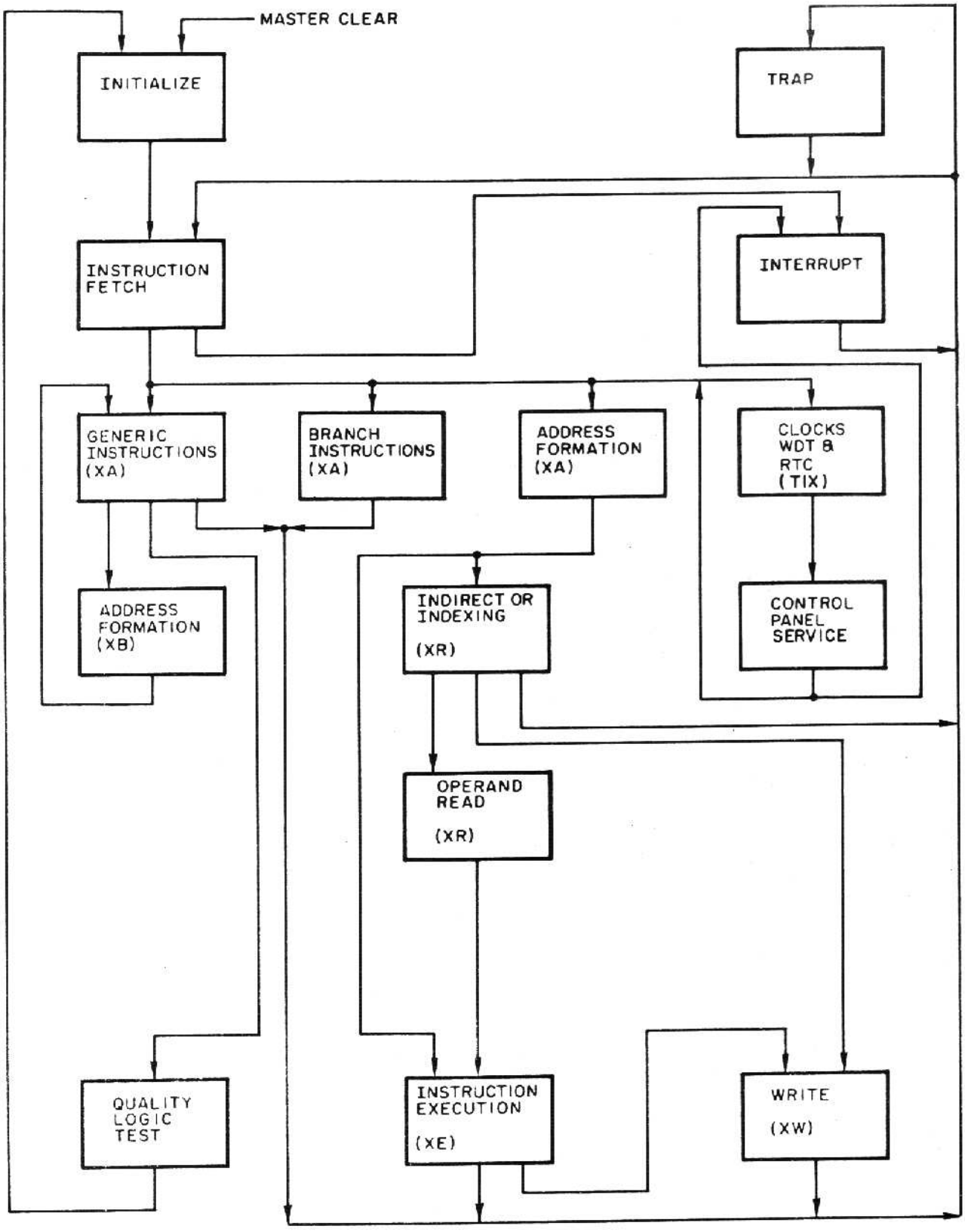


Figure 2-6 CPU Firmware Overview Flow Chart

2.4.5 Address Formation (XA)

The XA firmware examines the instruction, subdividing it into categories and subcategories:

- Category 1: Standard 7-bit address syllable (subcategories: P + D/B + D, B/@B, etc.).
- Category 2: Branches (subcategories: long displacement, short displacement, etc.).
- Category 3: Generics (subcategories: HLT, RTCN, MCL, MMM, etc.).
- Category 4: Shifts (subcategories: SOL, DCR, etc.).
- Category 5: Short-value Immediates (subcategories: +, -).

For instructions using the 7-bit address syllable, XA starts effective address generation. If the address syllable calls for a register or if the operand is immediate, the XA firmware exits to XE; otherwise, the XA firmware exits to XR.

2.4.6 CIP Address Formation (XB)

The XB firmware is similar to XA, but interprets a different address syllable format for commercial data descriptors.

2.4.7 Indirect/Indexing/Operand Fetch (XR)

The XR firmware performs indirection and indexing if needed. It then fetches an operand if the instruction requires it. Some instructions, such as jumps, are executed within XR (i.e., without further major branching). For others, XR exits to XE.

2.4.8 Instruction Execution (XE)

The XE firmware selects one of several op-code dependent entry points where instruction execution begins.

2.4.9 Result Write (XW)

The XW firmware stores the result, if necessary, after the instruction is executed. This firmware is entered only by instructions that must return their results to a palce specified by an address syllable.

2.4.10 Trap and Interrupt

Traps are distinguished from interrupts by being synchronous with and caused by the program currently being executed. On the other hand, interrupts are generally related to the program currently being executed, or at best are asychronous with it.

2.4.10.1 Traps

When processing traps, the processor generates addresses to dedicated memory locations (see Figures 2-7 and 2-8) that contain the Next Available Trap Save Area Pointer (NATSAP) and the trap vectors (i.e., pointers to the trap handler procedures). The various events that can cause a trap and their associated vector numbers are listed in Table 2-5. For a detailed description of the trap firmware refer to subsection 3.2.

The Trap firmware is enterd from any one of many locations within the firmware when a trap condition is detected.

2.4.10.2 Interrupts

The CPU interrupt hardware consists of two sections: (1) the interrupt busy flip-flop together with the logic for comparing the CPU level to the incoming interrupt level, and (2) the interrupt data register, which stores an acknowledged interrupt so that the processor can service it. Interrupts are classified by their source as follows:

1. Externally generated by the following events:

    a. Peripheral device completed an assigned activity
    b. Peripheral device changed state
    c. Power failure
    d. Dialog from other processors.

2. Internally generatged by the following events:

    a. Execution of a suspend, dispatch, or quick level change instruction
    b. Watchdog timer runout
    c. Real-time clock runout
    d. Trap save area pool exhausted.

For a detailed description of interrupts, see subsection 3.3.

The interrupt firmware is enterd from a number fo sources, depending on the operation currently being performed by the CPU. These interrupt sources are as follows:

- Use of last trap save area
- Programmed interrupt (LEV)
- External device task completion
- RTC runout
- WDT runout
- Incipient power failure.

2.4.11 Quality Logic Test (QLT)

The QLT provides a basic confidence test of the CPU logic and the memory. The QLT is entered via the MCL instruction when the control panel is in Load mode.

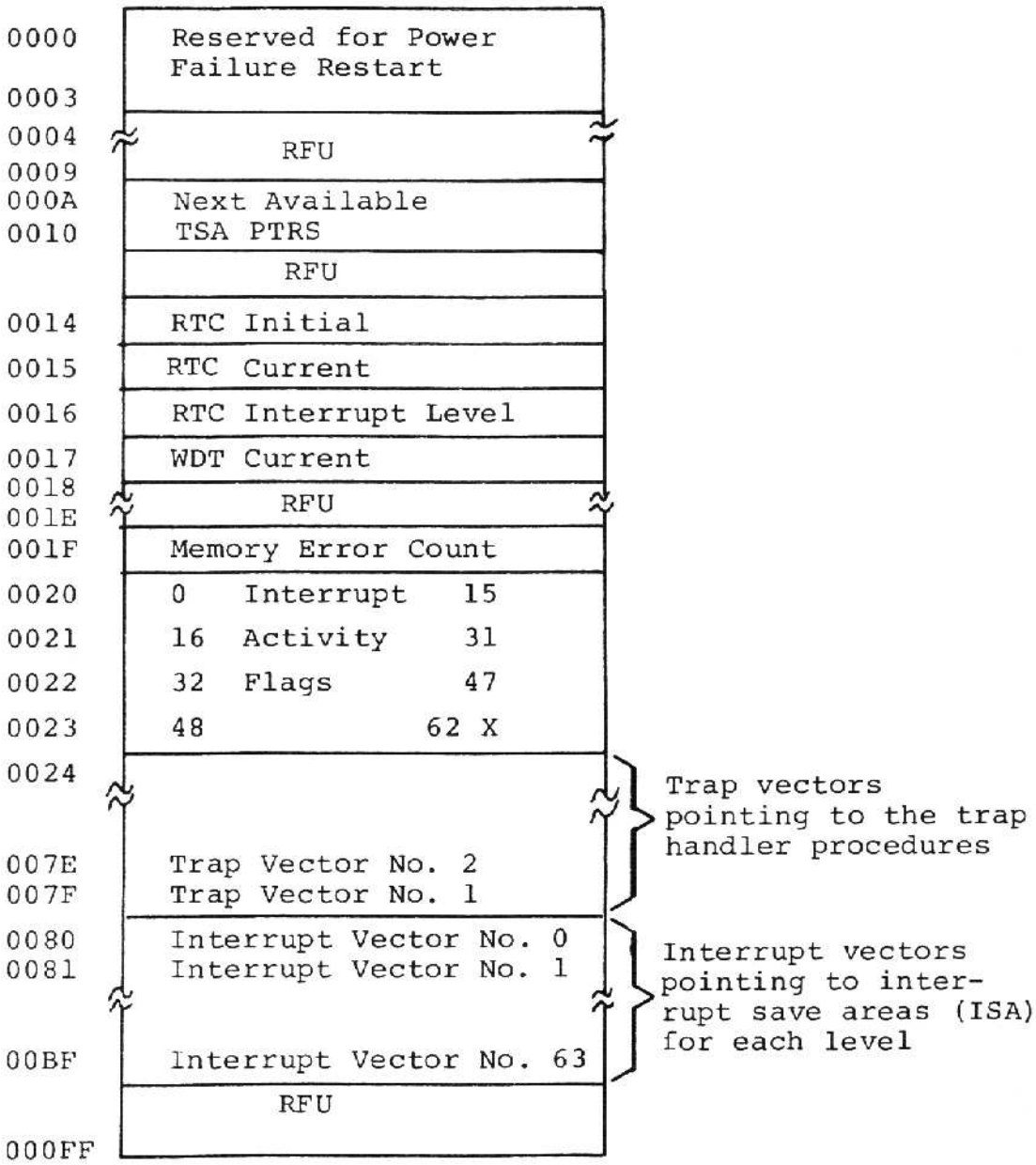


Figure 2-7 Dedicated Memory Locations in SAF

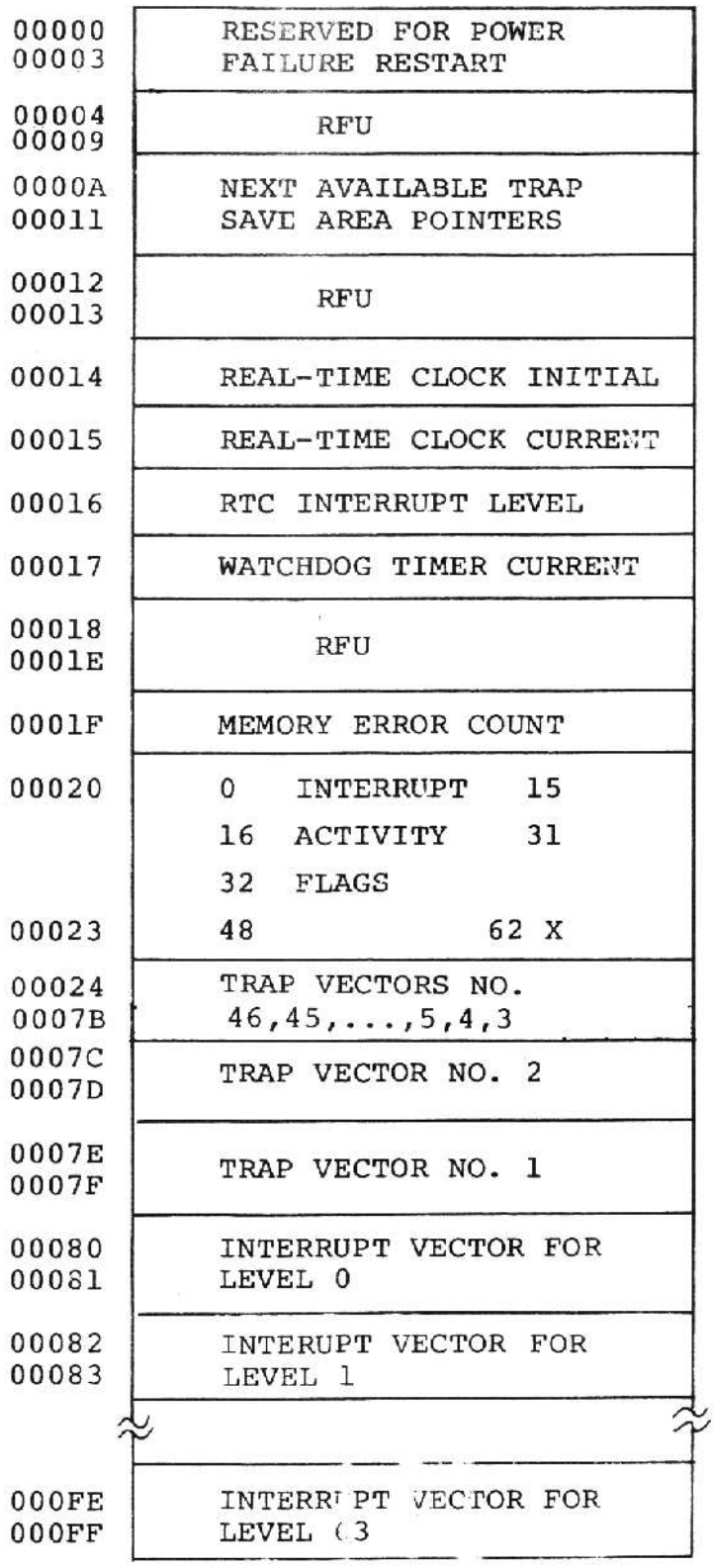


Figure 2-8 Dedicated Memory Locations in LAF

Table 2-5 Trap Event, Trap Vectors, and Halt Locations

TV NUMBER	TV		EVENT NAME	P*	
	SAF	LAF		SAF	LAF
1	7F	7E	Monitor Call	80	80
2	7E	7C	Trace/Breakpoint**	7F	7E
3	7D	7A	Unimplemented SIP Operation	7E	7C
4	7C	78	Reserved for Software Use	7D	7A
5	7B	76	Unimplemented Non-SIP Operation	7C	78
6	7A	74	Integer Register Overflow**	7B	76
7	79	72	Scientific Divide by Zero	7A	74
8	78	70	Scientific Exponent Overflow	79	72
9	77	6E	Stack Underflow	78	70
10	76	6C	Stack Overflow	77	6E
11	75	6A	-	76	6C
12	74	68	Illegal Remote Data Descriptor	75	6A
13	73	66	Privilege Violation	74	68
14	72	64	Protection Violation	73	66
15	71	62	Unavailable Resource	72	64
16	70	60	Program Error	71	62
17	6F	5E	Uncorrectable Memory Error	70	60
18	6E	5C	-	6F	5E
19	6D	5A	Scientific Exponent Underflow**	6E	5C
20	6C	58	Scientific Program Error	6D	5A
21	6B	56	Scientific Significance Error**	6C	58
22	6A	54	Scientific Precision Error**	6B	56
23	69	52	External Processor Unavailable Resource	6A	54
24	68	50	External Processor Uncorrectable Memory Error	69	52
25	67	4E	-	68	50
26	66	4C	-	67	4E
27	65	4A	-	66	4C
28	64	48	-	65	4A
29	63	46	-	64	48
30	62	44	CIP Failed QLT	63	46
31	61	42	SIP Failed QLT	62	44
32	60	40	-	61	40
33	5F	3E	-	60	40

*The CPU halts here if NATSAP is not null, but the TV is null.

**Mask controlled trap.