TM 1010

TECHNICAL MANUAL
for
MODEL 1200
CENTRAL PROCESSING UNIT

September 1, 1972

Revised: August 10, 1973

basic / four corporation
A Subsidiary of Management Assistance Inc. (MAI)
1335 South Claudina Street, Anaheim, California 92805

Printed in U.S.A.

List of Effective Pages

Page	Revision Date
iii thru xii	Original
xiii, xiv	August 10, 1973
xv	February 1, 1973
xvi	August 10, 1973
1-1 thru 1-54	Original
2-1 thru 2-46	Original
2-47	February 1, 1973
2-48 thru 2-54	Original
2-55	February 1, 1973
2-56 thru 2-115	Original
3-1 thru 3-25	Original
4-1 thru 4-9	Original
5-1	February 1, 1973
5-2 thru 5-4	Original
5-5	February 1, 1973
5-6 thru 5-10	Original
5-11	February 1, 1973
5-12 thru 5-19	Original
5-19.1 thru 5-19.3	August 10, 1973
5-20	August 10, 1973
5-21	Original
5-22	August 10, 1973
5-23	Original
S1-i, S1-ii	Original
S1-1 thru S1-13	Original
S2-i	February 1, 1973
S2-1 thru S2-6	February 1, 1973
S3-i	Original
S3-1 thru S3-13	Original
S4-i	Original
S4-1 thru S4-4	Original
S5-i	February 1, 1973
S5-1 thru S5-13	February 1, 1973
S6-i, S6-ii	August 10, 1973
S6-1 thru S6-26	August 10, 1973
S7-i, S7-ii	February 1, 1973
S7-1 thru S7-32	February 1, 1973

TABLE OF CONTENTS

Page

SECTION 1. INTRODUCTION (continued)

Page

SECTION 1. INTRODUCTION (continued)

Page

SECTION 1. INTRODUCTION (continued)

SECTION 2. CPU THEORY OF OPERATION

Page

SECTION 2. CPU THEORY OF OPERATION (continued)

Page

SECTION 2. CPU THEORY OF OPERATION (continued)

Page

SECTION 2. CPU THEORY OF OPERATION (continued)

Page

SECTION 2. CPU THEORY OF OPERATION (continued)

Page

SECTION 2. CPU THEORY OF OPERATION (continued)

SECTION 3. OPERATION

Page

SECTION 3. OPERATION (continued)

SECTION 4. MAINTENANCE

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (continued)

SECTION 1

GENERAL DESCRIPTION

1.1 SCOPE

This manual contains the general description, installation procedures, operation, theory of operation, and maintenance data for the Model 1200 Central Processing Unit (CPU), Part Number 400100, manufactured by the Basic/Four Corporation, 18552 MacArthur Boulevard, Santa Ana, California 92707. The CPU is an integral part of the Basic/Four data processing systems. The logic diagrams for the CPU are in LD 1010.

1.2 MANUAL ORGANIZATION

Sections 1 through 5 of this manual cover the CPU portion of the computer system. The CPU includes the following basic elements:

a. Control Board
b. Data Board
c. Front Panel Board
d. Interface Board

The basic elements are common to all Basic/Four Model 1200 Central Processing Units. Sections 1 through 5 provide detailed descriptions and instructions for these elements, together with interface descriptions that relate the CPU to other elements of a computer system. These other elements, including core memory, control memory, power supply, and I/O device controllers are either optional or variable, depending on the requirements of each application. Supplements, following Section 5, describe these elements in a specific application.

Sections 1 through 5 contain the following information:

a. Section 1 - General Information. This section contains physical and functional descriptions of the CPU. Also included are detailed descriptions of the microcommand repertoire, and reference data.

b. Section 2 - Theory of Operation. This section contains block diagrams and data flow descriptions, detailed logic descriptions, timing descriptions, and instruction flow diagram descriptions for the CPU. Overall and detailed block diagrams support the block diagram and data flow descriptions. Timing

diagrams support the timing descriptions. The instruction flow diagrams and descriptions describe the sequence of actions performed by the CPU for each microcommand in the CPU repertoire. Interfaces between the CPU and the other elements of a computer system are also described.

c. Section 3 - Operation. This section describes the operator controls and indicators that are included on the system front panel (part of the front panel board). The various front panel modes of operation are described, and detailed procedures are provided for using the front panel controls and indicators to exercise the CPU.

d. Section 4 - Maintenance. This section describes the use of the procedures in Section 3 for checkout and troubleshooting of the CPU. Adjustment procedures are also included in this section.

e. Section 5 - Parts Lists. This section contains assembly drawings and parts lists for the major elements of the CPU.

1.3 EQUIPMENT PURPOSE AND CAPABILITIES

The CPU is a microprogrammed digital data processor. Microprogramming is a technique for obtaining maximum utilization of computer capabilities by efficient use of the machine subcommands. These hardware-implemented subcommands cause the CPU to execute specific storage, retrieval, arithmetic, or logical operations. A sequence of subcommands are employed to perform specific data processing routines or subroutines. These subcommand sequences (microprograms) are translated into a sequence of 16-bit microcommands that are stored in control memory. The microcommands stored in control memory constitute the firmware for a specific data processing application. A firmware routine or subroutine is executed by the CPU as follows:

a. The 16-bit microcommands that compose a particular firmware routine, or subroutine, are stored in consecutive locations in control memory.

b. The CPU sequentially accesses each microcommand storage location in control memory.

c. Each 16-bit microcommand read from control memory is stored, decoded and executed by the CPU. In some cases, the microcommand includes an 8-bit literal for storage or computation purposes. In other cases, the microcommand contains a core memory address for a core memory read or write operation.

d. At the end of a firmware routine, or subroutine, a jump microcommand may be utilized to proceed to the next firmware routine or subroutine. The jump microcommand causes the CPU to access the first location of the next firmware routine, or subroutine, to be executed.

The flexibility of the CPU is enhanced by the jump capability, plus the capability to modify microcommands received from control memory with software instructions read

from core memory. The jump capability makes more efficient use of the firmware in control memory. For example, a firmware routine can be made up of a number of multi-purpose firmware subroutines, each stored in different segments of control memory. In this case, the microcommands in the firmware routine include jump instructions that direct the CPU to jump to the start of each firmware subroutine in control memory.

At the end of each subroutine, a return jump instruction causes the CPU to return to the next instruction in the firmware routine. The capability to modify microcommands with instructions received from the core memory permits data processing operations to be performed under the combined control of firmware and software. In this case, the firmware-to-software transition is achieved by inserting a firmware subroutine that typically performs the following actions:

a. The next software instruction (two 8-bit words) is read from core memory and stored in the CPU.

b. A Modify Lower Command (MLC) microcommand is issued, notifying the CPU to modify the low-order bits of the next microcommand.

c. An Execute, Literal Type (ELT) microcommand is issued. This microcommand would typically contain all zeros in the eight high-order bits, and all ones in the eight low-order bits. The resulting effective microcommand stored in the instruction (R) register would then be identical to the two 8-bit words read from core memory. The CPU then executes the software-derived instruction in the R Register. Typically, this instruction causes the CPU to jump to the beginning of a firmware routine, or subroutine, specified by the address in the effective microcommand.

The described capabilities permit the CPU to be applied directly to problem solutions by full programming at the micro level, by emulating the operation of a macro level computer in which software instructions are stored in core memory, or by a combination of micro- and macro-level programming. Options are available that permit the user to modify the microprograms stored in control memory, permitting greater flexibility in the application of the system.

The arithmetic/logic unit (ALU) of the CPU typically performs arithmetic and logic operations on a selected 8-bit operand and the 8-bit contents of a selected file register. The CPU contains 30 eight-bit general-purpose file registers for storage of user's data. The selected operand may be an 8-bit word from the I/O bus or the core memory, a literal from control memory, or the result of a previous arithmetic,

logical or storage operation. A typical microcommand for an arithmetic or logical operation designates the file register, the source of the operand, and the type of computation to be performed by the ALU. The result produced by the ALU is then stored in a file register or in one of the other CPU registers.

1.4 REFERENCE DATA

Table 1-1 provides complete reference data for the CPU. Included in Table 1-1 are power requirements, environmental conditions, dimensions, and circuit characteristics.

TABLE 1-1. SYSTEM SPECIFICATIONS

PARAMETER	CHARACTERISTICS
Input Power Requirements	115/230 Vrms ±10%, single-phase, 47-63 Hz, 350 watts maximum
Power Supply:	
Type	Series regulator power supply with three regulated dc outputs.
DC Outputs	+12 Vdc, 1.2A, ±5% adjustable; +5 Vdc, 20A, ±5% adjustable; -16.75 Vdc, 3.5A, ±2% adjustable
Overload/Overvoltage Protection	+7 Vdc limit on +5 Vdc output; overload current limiting with automatic recovery
Power Fail Detect	Power failure is detected two milliseconds before loss of dc regulation; automatic recovery after ac is restored.
Line Filtering	Integral line filter for ac line transient protection
Circuit Types	Hermetically sealed semiconductors and ceramic integrated circuits used throughout
Environmental:	
Operating Temperature	0-50 degrees C (32-122 degrees F)
Relative Humidity	10-90% without condensation
CPU Circuit Types	Integrated LSI, MSI, and SSI circuit design throughout. TTL logic is used internally, and DTL logic is used for input/output drivers.

TABLE 1-1. SYSTEM SPECIFICATIONS (cont)

PARAMETER	CHARACTERISTICS
Dimensions:	
Rack-Mounted Enclosure	10.5-inch height by 20-inch depth by 19-inch width
Internal Logic Levels	Logical 1 = +5 V (nominal); logical 0 = 0 V
External Logic Levels	Logical 1 = 0 V (nominal); logical 0 = +3 V (nominal)
Remote Power Supply (rack-mounted)	8-3/4-inch height by 10.5-inch depth by 19-inch width
Clock Rate	5-MHz clock rate (crystal-controlled)
Command Repertoire	See Table 1-4
Command Execution Times	See Table 1-4
Arithmetic/Logic Unit	8-bit parallel, binary, fixed point. One's and two's complement for negative numbers.
File Registers	30 general-purpose 8-bit file registers plus one 8-bit flag register
Control Memory	Up to 2048 words (16 bits) of control memory. Control memory is implemented with bipolar read only memory (ROM) devices. Minimum system has 1280 words of control memory.
Core Memory	Core memory containing up to 65,536 eight-bit bytes can be accommodated in basic enclosure. Core memory is implemented with 8192-byte modules. Wide-temperature-range lithium cores are utilized. Either full- or half-cycle memory read or write operations can be performed. Full-cycle time is 1 microsecond; half-cycle time is 600 nanoseconds.
Power Fail and Restart	Automatic shut-down in the event of a power failure; automatic restart when power is restored.
Input/Output Interface	Separate 8-bit parallel input and output data buses for communication with I/O device controllers in either programmed or concurrent (block transfer) modes.
Real Time Clock	Real Time Clock produces an interrupt once every millisecond.

TABLE 1-1. SYSTEM SPECIFICATIONS (cont)

PARAMETER	CHARACTERISTICS
Direct Memory Access/Disc Controller	DMA channel allows the disc memory to communicate directly with the core memory at data transfer rates up to 1-million bytes per second.
TTY controller	An optional parallel TTY controller is available.
Miscellaneous Controllers	Magnetic tape controller, card reader controller, paper tape controller, multi-channel TTY controllers, line printer controller, and 2400 baud controller are available.
Software	A variety of supporting standard software, including special firmware development packages, are available. These packages include assemblers, simulators, generators for direct conversion of language assembler outputs into control memory bit patterns, diagnostics, and alterable control memory operating systems.

1.5 PHYSICAL DESCRIPTION

1.5.1 ENCLOSURES

The CPU is enclosed in an 18-slot card-cage enclosure with removable front panel and power supply. The entire assembly is mounted, along with a Disc Drive Unit and power supply in a Basic/Four system cabinet. Controllers are inserted into slots in the CPU enclosure and interface with external equipment.

1.5.2 BACKPLANE

The backplane contains connector slots for all computer system elements except the front panel board. The backplane can be supplied with 18-connector slots.

Printed circuit (PC) boards installed in the backplane connector slots are accessible from the rear of the enclosure. The connector pins of the backplane are accessible with the front panel removed. All PC boards are equipped with handles for convenient extraction, insertion, and lock-down. The boards may be installed on extender boards for troubleshooting and checkout. All PC boards have dimensions of 8.575 inches by 12.5 inches. PC boards that serve as controllers for I/O devices include connectors which provide connection by ribbon cables to the devices.

The first three slots of the backplane are dedicated to the CPU.

Slot No.	Dedicated Use
J1	Interface Board
J2	Control Board
J3	Data Board

A typical CPU card cage for a system with maximum core memory would have this configuration: J1 - interface with integral TTY controller, J2 - control, J3 - data, J4 through J11 - 8K core memory, J12 - control memory, J13 through J16 - priority selected controllers, J17 - DMA/disc controller, and J18 - power supply extender board. The normal priority sequence for the seven available types of controllers is (from highest to lowest priority): eight channel, four channel, 2400 baud, magnetic tape, paper tape, card reader, and line printer. The magnetic tape controller requires two slots; all other controllers require only one slot. The eight channel, four channel, and 2400 baud controllers provide the interface for video display terminals and/or accounting machine terminals.

Slots J4 through J17 are used for core memory, control memory, the DMA channel, I/O device controllers, and other options. Slot J18 is reserved for the power supply extender board. The DMA/disc controller is installed in slot J17. The PC boards for the core memory are installed in the slots immediately following slot J3, and the PC board for the control memory is installed immediately following the last core memory board. Optional I/O and device controllers are installed in the slots that follow the last control memory board. The order of installation in these slots is usually dependent on the order of device priority, such that the highest-priority device or I/O controller is installed in the first slot following the control memory, and the lowest-priority controller is installed in the slot with the highest number. Controllers must not be separated by empty slots.

1.5.3 INTERCONNECTIONS

The elements of the CPU are interconnected as shown in Figure 1-1. The front panel board is connected to the other elements of the CPU by means of two ribbon cables attached to the interface board in backplane slot J1. The signals to and from the front panel board are distributed via the backplane between slot J1 (interface board) and slots J2 (control board) and J3 (data board). Interconnections between the

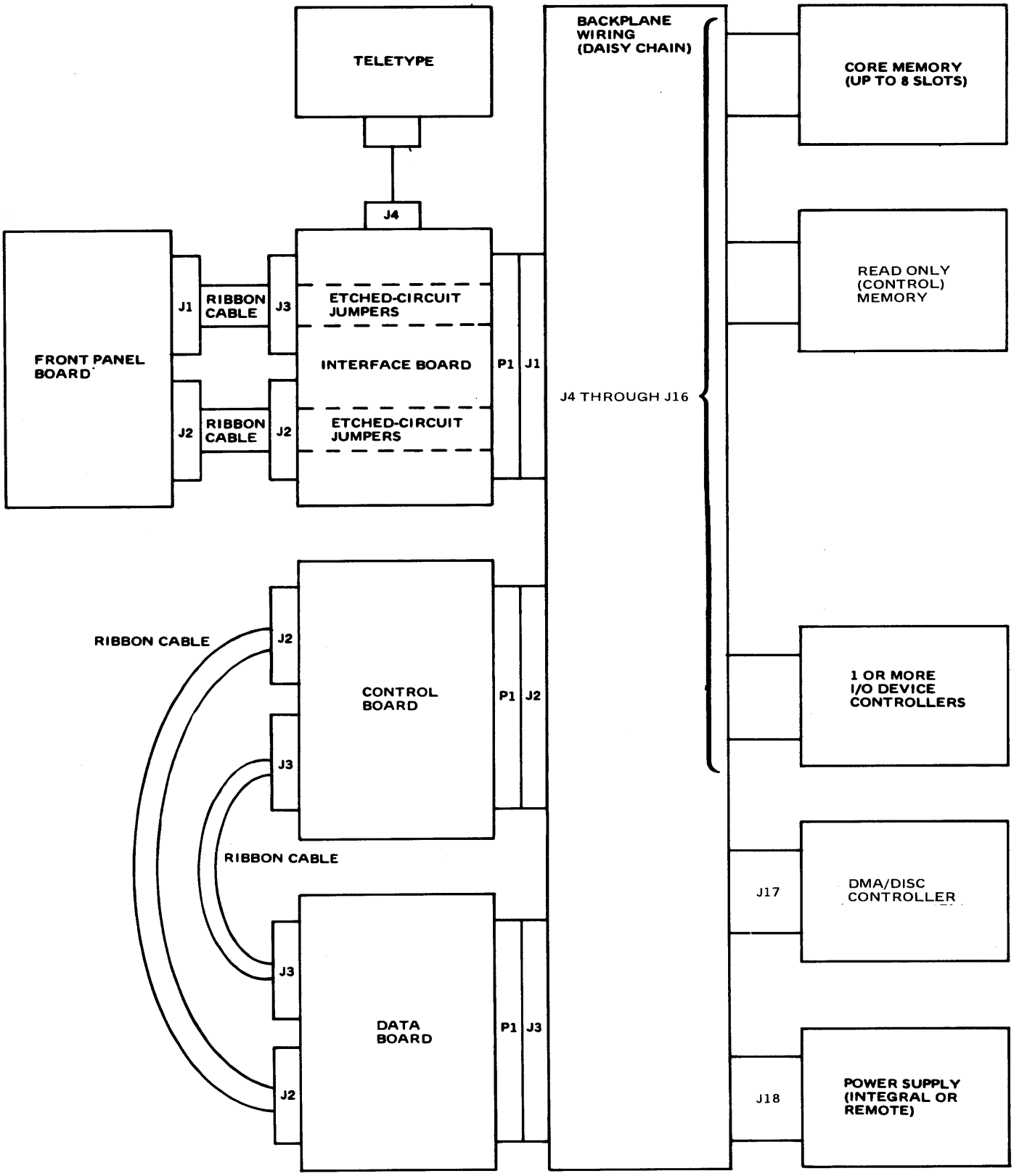


Figure 1-1. CPU Interconnections

control and data boards are made by means of two ribbon cables. Interconnections between the control and data boards and other elements mounted in the backplane are made via the backplane wiring. Power distribution is from the power supply (slot J14 or J18) to all elements connected in the backplane.

1.5.4 CPU PRINTED CIRCUIT BOARDS (Figures 1-2 through 1-4)

With the exception of transistor lamp drivers on the front panel board, all circuits are implemented by LSI, MSI, and SSI integrated circuits (IC). TTL logic is used internally, and DTL logic is used for input/output drivers.

Physical organization of the PC boards in the CPU are shown in Figures 1-2 through 1-4. In these figures a grid system is used to identify the integrated circuit elements. An integrated circuit reference designation consists of the column number followed by the row letter. Example: 5C is the integrated circuit in column 5, row C.

1.5.4.1 FRONT PANEL ASSEMBLY. The front panel assembly (Figure 1-2) consists of a front panel circuit board mounted to a panel that contains the labels for switches and indicators. All switches and indicators are physically mounted to the front panel board, and protrude through holes in the front panel. Four pins in the front

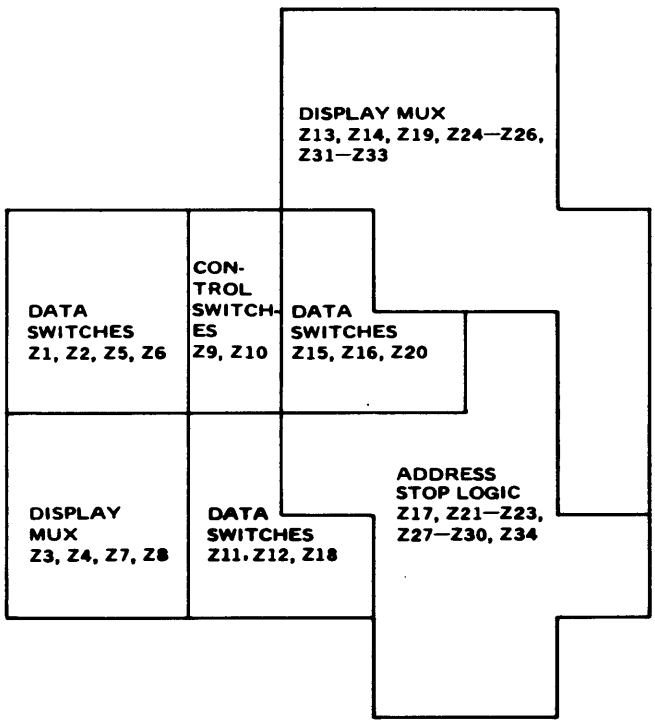


Figure 1-2. Front Panel Board Circuit Functions

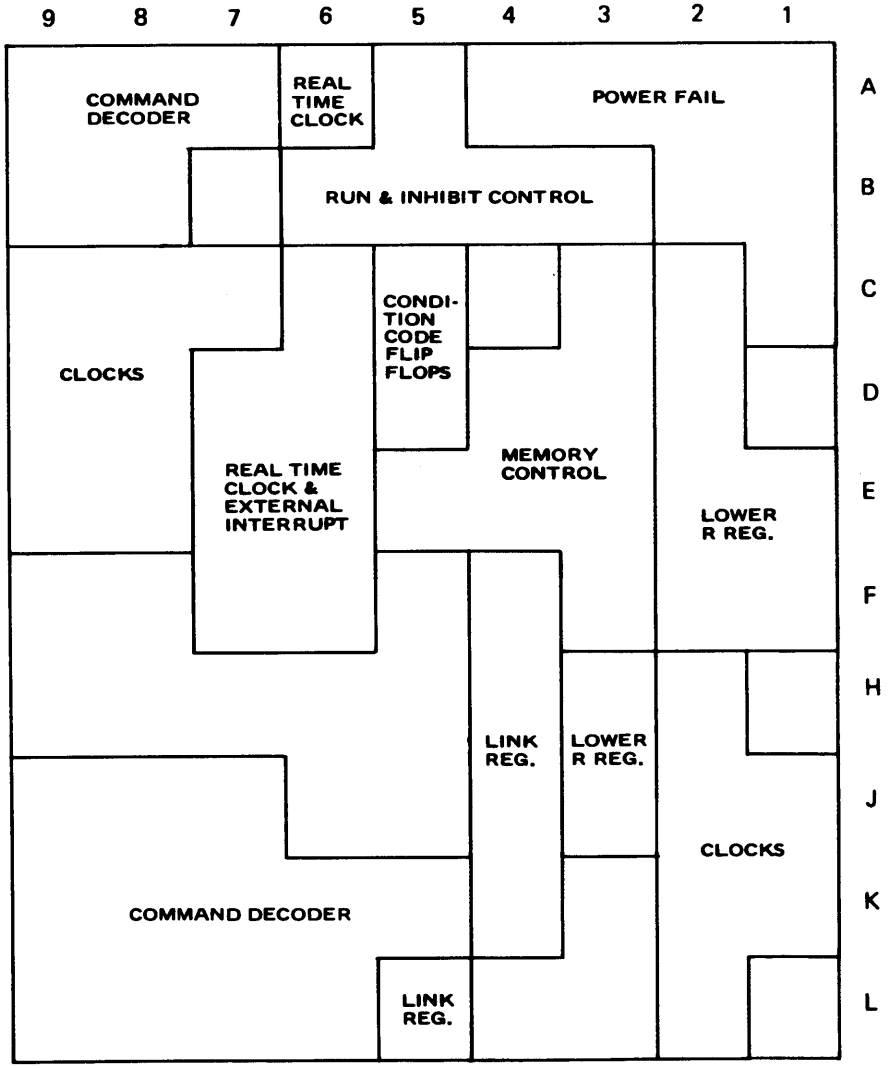


Figure 1-3. Control Board Circuit Functions

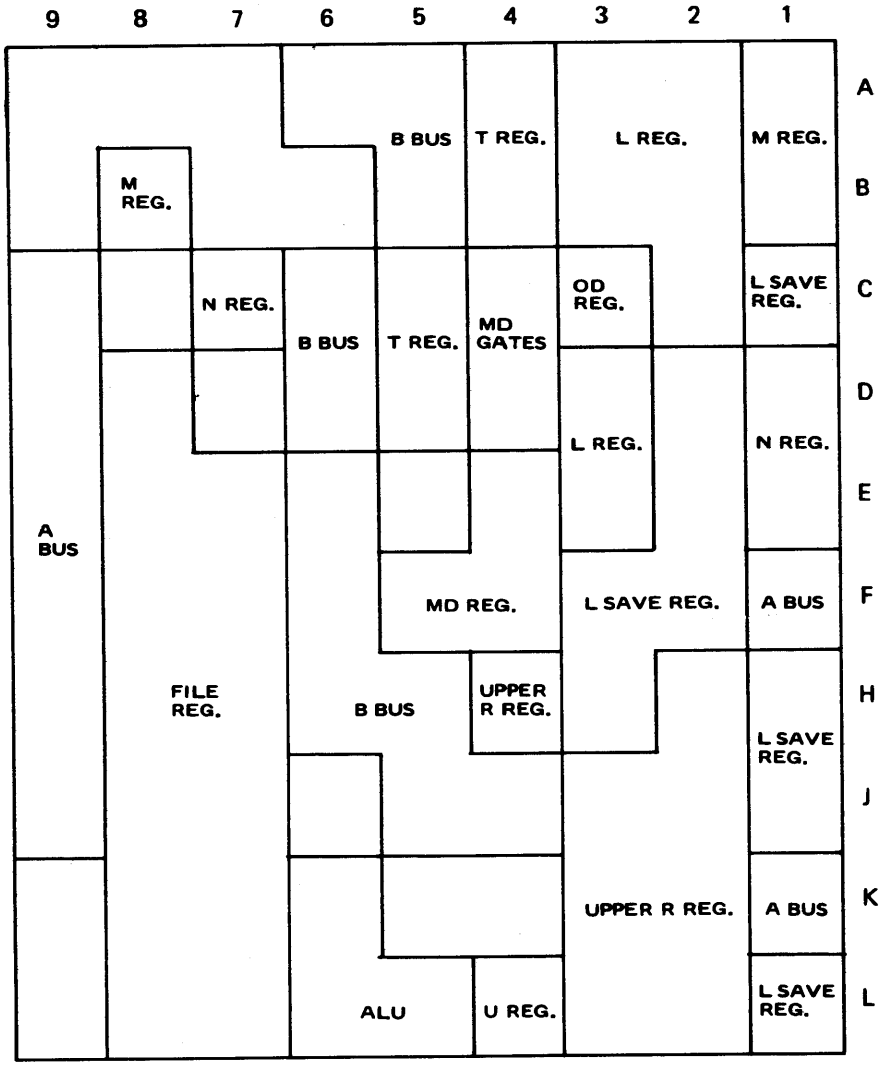


Figure 1-4. Data Board Circuit Functions

panel assembly provide the means for mounting the assembly in the enclosure. These pins mate with spring-loaded fasteners in the enclosure. The ribbon cables that connect the front panel board to the interface board are of sufficient length to permit removal or installation of the front panel board without stressing the cable connectors.

Two versions of the front panel assembly are available. The basic front panel contains a key-lock power switch, five machine state control switches, two machine state control indicators, and four sense switches. The system front panel contains identical switches and indicators to those located on the basic front panel. In addition, the system front panel contains a panel mode selector switch (with associated indicators), 16 data switches, 16 data display indicators, address stop and scan features, and associated display control pushbutton switches. The basic front panel is included in the unit, and provides all controls for normal operation. The basic front panel is removed and replaced by the system front panel for CPU maintenance.

Figure 1-2 illustrates the CPU front panel board, and identifies the integrated circuit elements associated with the various functional sections of the front panel.

These functional elements include the following:

a. Front Panel Control Switches
b. Sense Switches
c. Upper and Lower Data Switches
d. L Address Drivers and Comparator
e. Address Stop Logic
f. RS Bus Drivers
g. Front Panel Display Indicators

1.5.4.2 INTERFACE BOARD. The interface board serves both as the interface for the front panel, and as the controller for a Teletype. The front panel interface section consists of printed circuit jumpers between the backplane and the ribbon cables that connect to the front panel. The integral TTY controller receives 8-bit parallel data from the CPU, and converts this data to a serial form for transmittal via connector J4 to the Teletype. Serial data from the Teletype is converted by the TTY controller to an 8-bit parallel form for transmittal via the backplane to the CPU.

The front panel interface section of the interface board is described in Section 2.

1.5.4.3 CONTROL BOARD. Figure 1-3 identifies the integrated circuits associated with the various CPU elements contained on the control board. These functional elements include the following:

a. Clock Generator and Clock Gating
b. Run and Inhibit Control
c. Lower R Register
d. Command Decoder
e. Memory Control
f. Link Register
g. Initial Carry Logic
h. Condition Code Flip-Flops
i. External Interrupt Control
j. Power Fail Circuits
k. Real Time Clock

1.5.4.4 DATA BOARD. Figure 1-4 identifies the integrated circuits associated with the various CPU elements contained on the data board. These functional elements include the following:

a. L Register
b. L Save Register
c. U Register
d. Upper R Register
e. M Register
f. N Register
g. Memory Data Gates
h. T Register
i. MD Register
j. B Bus Multiplexer
k. A Bus Multiplexer
l. Arithmetic/Logic Unit (ALU)
m. File Registers
n. Internal Status Interrupt Control
o. Input Data Bus Gates

p. IC Register

q. OD Register

1.5.5 POWER SUPPLY

The power supply is mounted remotely. An extension cable connects between the remote power supply and a printed circuit board that is installed in power supply slot J18 of the backplane. The printed circuit board contains etched-circuit jumpers to the appropriate pins of the backplane. Usually a single power supply is used for both the CPU and the Disc Drive Unit.

The power supply has ample capacity to power all system configurations. A basic system consisting of the following elements would require +5 V current:

Item	Current (amps)
Basic CPU (control board and data board)	5.0
System front panel assembly	0.9
ROM	1.5
Four core memory modules containing 8192 words each	2.4
	9.8 amps total.

The basic system would occupy eight slots in the backplane. Each additional core memory requires 0.5 amps at +5 V. Since the power supply is capable of delivering 20 amps at +5 V, the basic system allows 10 amps for expansion in the remaining backplane slots. This leaves more than 1.5 amps for each I/O device controller.

1.6 FUNCTIONAL ORGANIZATION

As shown in Figure 1-5, the CPU is divided into four major functional sections. The functional sections are: Control Unit, Core Memory, Processor, and Input/Output Section.

a. Control Unit. This section consists of the control memory and its associated address and instruction storage registers, plus the command decoder that decodes the microcommands. The basic control memory may contain up to 2048 16-bit words.

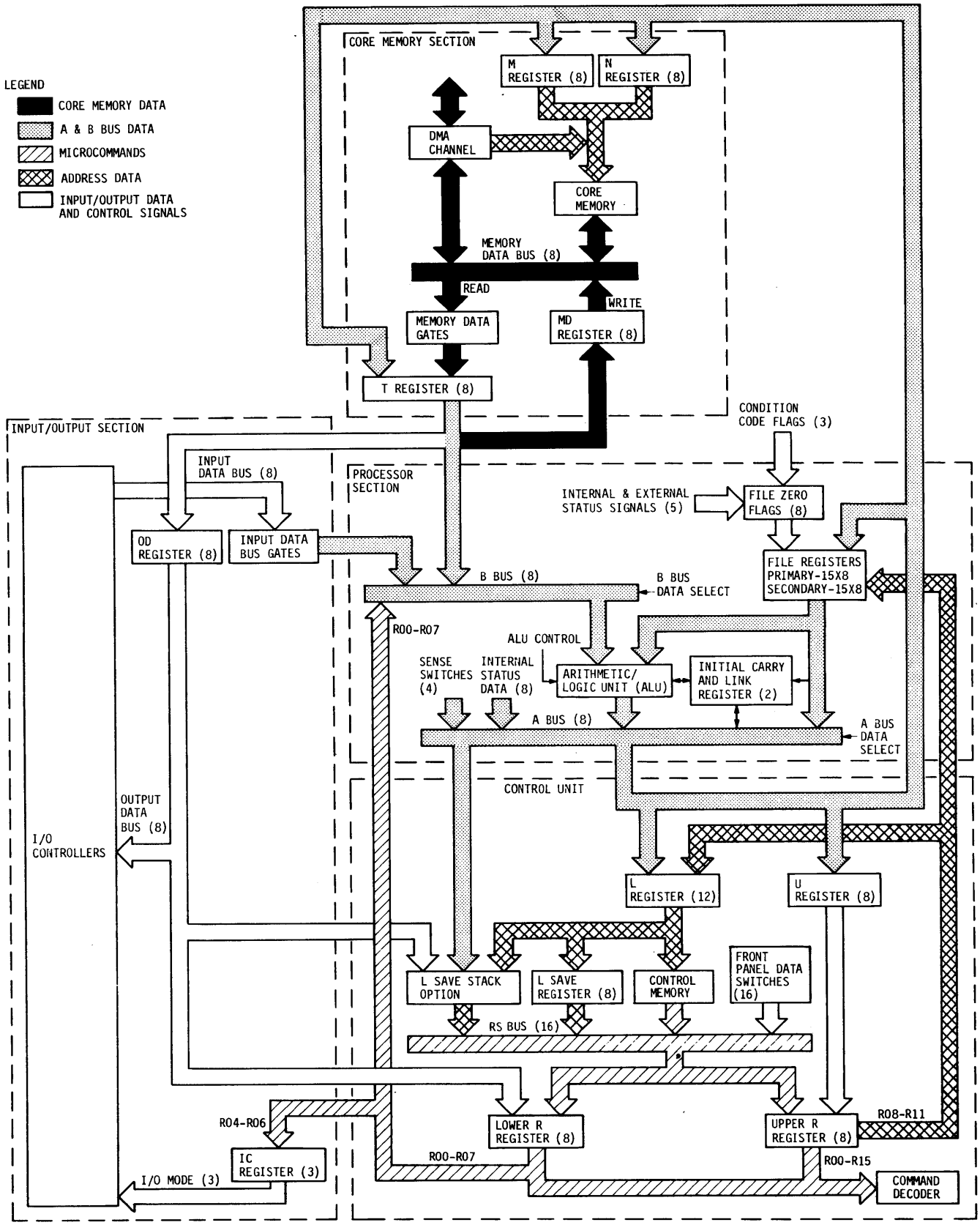


Figure 1-5. CPU Functional Block Diagram

b. Core Memory Section. This section consists of the variable-sized core memory and associated memory control logic, address and data registers, and an direct memory access (DMA) channel.

c. Processor Section. The processor section is organized around the arithmetic logic unit (ALU), with two buses (A and B) provided for data transfers. The T Register (indicated as part of the core memory section) is also part of the processor section, and serves as a buffer for transient data. Also included are general-purpose file registers and file zero flags.

d. Input/Output Section. This section provides the communications interface with the I/O device controllers connected to the byte I/O interface. Included are data gates associated with the input data bus, an OD (output data) Register associated with the output data bus, and an IC Register that produces the 3-bit I/O mode control used to control the I/O controllers.

1.6.1 REGISTERS

1.6.1.1 T REGISTER. The 8-bit T Register serves as the operand register for most operate-type commands, and as a buffer for data being written into, or read from, core memory. Data to be supplied over the output data bus is also entered into the T Register and supplied to the OD Register.

1.6.1.2 M AND N REGISTERS. The 8-bit M and N Registers hold the current address of the location being accessed in core memory. The M Register holds the eight high-order bits of the address, and the N Register holds the eight low-order bits.

1.6.1.3 U REGISTER. The 8-bit U Register stores data to be used in modifying the eight high-order bits of execute-type microcommands received from control memory. The contents of the U Register are OR'ed with the eight high-order bits from control memory, and the result is stored in the upper R Register.

1.6.1.4 FILE REGISTERS. The File Registers include the file zero flags (File 0), a primary file containing fifteen 8-bit registers, and a secondary file that also contains fifteen 8-bit registers. Either the primary or secondary file is selected by microcommand. Then, each microcommand that involves a manipulation of data in the selected file, specifies which of the 15 registers is to be accessed for a file read and/or write operation. File 0 cannot be written into, but is used as a source of data. Data written into the File Registers is from the A bus and data read from a File Register is supplied to the ALU and to the A bus.

1.6.1.5 L REGISTER. The 12-bit L Register holds the address of the next microcommand to be read from control memory, permitting direct addressing of any location in a 2048-word control memory bank. While sequencing through a firmware routine, or subroutine, the L Register is incremented by one as each instruction is executed, unless it is loaded with a jump or return jump-type microcommand. In this case, the L Register is parallel-loaded with the jump address. The following types of jump addresses can be loaded into the L Register:

a. Jump Extended. The jump extended (JE) microcommand contains a 12-bit jump address, permitting a jump to any location within a 2048-word bank of control memory.

b. Return Jump. When a return jump microcommand is executed, the contents of the 12-bit L Save Register are loaded into the L Register, permitting a return to any location within a 2048-word bank of control memory.

c. Jump in 1K. The jump in 1K (JP) microcommand contains a 10-bit jump address, permitting a jump to any location within a 1024-word sector of control memory.

d. Jump in 1/2K. The jump in 1/2K microcommand is an operate-type command that contains the most significant bit (MSB) of a 9-bit jump address. The 8 least significant bits (LSB) of the jump address are placed on the A bus as the result of the operate-type command execution. The 9-bit jump address is loaded into the L Register, permitting a jump to any location within a 512-word sector of control memory.

1.6.1.6 L SAVE REGISTER. When a jump extended or return jump microcommand is executed, the incremented contents of the L Register (before the jump address is loaded) are stored in the L Save Register unless storage was inhibited by a previously executed Inhibit L Save (ILS) microcommand. Subsequent execution of a return jump microcommand causes the L Save contents to be transferred back to the L Register. In this manner, the L Save Register acts as a linkage register for a single level subroutine.

When the L Save Stack Option is included in the system, the L Save Register is disabled, and is replaced by an L Save Stack containing 16 registers that perform the same function as the L Save Register. In this case, however, multi-level subroutines are permitted.

1.6.1.7 R REGISTER. The 16-bit R Register is divided into an upper R Register and a lower R Register. Each microcommand from control memory is supplied via the RS bus and stored in the R Register. For execute-type microcommands, the eight

high-order bits from control memory are OR'ed with the contents of the U Register before storage in the R Register. Following execution of a Modify Lower Command (MLC) microcommand, the eight low-order bits of the next microcommand from control memory are AND'ed with the contents of the OD Register before storage in the R Register. Following execution of an Enter Console Switches (ECS) microcommand, the eight low-order bits of the next microcommand from control memory are AND'ed with the states of the eight low-order data switches (on the front panel) before storage in the R Register.

During execution of a return jump microcommand, the 12-bit contents of either the L Save Register or the selected register of the L Save Stack are loaded into the 12 low-order positions of the R Register. These bits are then supplied to the L Register.

During test operations, when the CPU is controlled from the front panel, the front panel data switches serve as the source of the microcommands stored in the R Register.

1.6.1.8 LINK REGISTER. The 2-bit Link Register contains storage for an arithmetic link bit and a memory link bit. For a given arithmetic operation (add, increment, subtract, decrement, compare, or shift), only one of the two link register stages are operative. For operations involving storage in the M or N Registers (memory address), the memory link stage is active (the arithmetic link stage is active for all other registers). The selected link register stage stores the carry-out of the high-order bit position in the ALU (add, increment, subtract, decrement, or compare), or the out-shifted bit produced by a shift operation.

1.6.1.9 MD REGISTER. The 8-bit MD (memory data) Register is a buffer for the next word to be written into core memory. The MD Register automatically copies the contents of the T Register when a write operation is in progress.

1.6.1.10 IC REGISTER. The 3-bit IC Register is loaded with a non-zero code when an external input/output (I/O) command is being executed. The 3-bit code loaded into the IC Register is received from stages 4 through 6 of the R Register. The output of the IC Register is supplied to I/O device controllers where it is decoded. The decoded I/O mode control command causes the I/O device controller to execute a specified input or output operation.

1.6.1.11 OD REGISTER. The 8-bit OD (output data) Register supplies bytes to the output data bus. When the IC Register contains an all-zero code, the OD Register copies the contents of the T Register. Then, when the IC Register contains a non-zero code, the current contents of the OD Register are held (freeing the T Register for other uses), and supplied to the output data bus.

1.6.2 MAIN DATA BUSES

1.6.2.1 MEMORY DATA BUS. The bidirectional memory data bus connects the CPU and the optional DMA channel to the core memory. When the DMA channel is selected, the M and N register address lines to core memory, the MD register output lines, and the memory data gates are all inhibited.

During a CPU-initiated memory read or write operation, the core memory is addressed by the M and N Registers. When the CPU initiates a memory read operation, MD register outputs are inhibited, and data read from the addressed memory location is supplied via the memory data bus and the memory data gates to the T Register. When the CPU initiates a memory write operation, the memory data gates are inhibited, and the contents of the MD Register (received from the T Register) are supplied via the memory data bus to the addressed memory location.

1.6.2.2 INPUT DATA BUS. The input data bus connects I/O controllers to the CPU. The 8-bit input data bytes from I/O controllers are supplied via the input data bus and the input data bus gates to the B bus.

1.6.2.3 OUTPUT DATA BUS. The output data bus connects the CPU to I/O controllers. The 8-bit output data bytes from the OD Register are supplied via the output data bus to I/O controllers.

1.6.2.4 B BUS. The B bus multiplexer is used to select the data that is placed on the B bus. The B bus multiplexer may select any of the following as the data on the B bus:

a. Complemented or uncomplemented data from the input data bus gates.
b. Complemented or uncomplemented data from the T Register.
c. An 8-bit literal or jump address contained in the lower R Register.

The data placed on the B bus is supplied to the ALU.

1.6.2.5 A BUS. An A bus multiplexer is used to select the data that is placed on the A bus. The A bus multiplexer may select any of the following for placement on the A bus:

a. An 8-bit internal status byte.
b. The states of the four front panel sense switches.
c. The output of the ALU.
d. The right-shifted contents of a selected file register.

The data placed on the A bus is distributed to the following points:

a. L Save Stack Option (if this option is included)
b. L Register
c. U Register
d. File Registers
e. M Register
f. N Register
g. T Register

Destination codes in the microcommand being executed specify which register stores the A bus data.

1.6.2.6 RS BUS. The RS bus supplies microcommands and return jump addresses to the R Register. During normal operation, the data on the RS bus is the microcommand from the control memory location addressed by the L Register, except when a return jump is being executed. In this case, the microcommand output from control memory is inhibited, and the RS bus contains the return jump address received from either the L Save Register or the L Save Stack Option. When an Enter Console Switches (ECS) microcommand is executed, the eight low-order bits of the next microcommand from control memory are AND'ed on the RS bus with the states of the eight low-order data switches (on the front panel). During test operation, when the CPU is being controlled by the front panel, the microcommand outputs from control memory are inhibited, and the RS bus contains the states of the front panel data switches.

1.6.3 ARITHMETIC/LOGIC UNIT (ALU)

With the exception of right shift operations, which are performed by the A bus multiplexer, the ALU performs all arithmetic and logical operations in the CPU. In

addition, the ALU serves, for some microcommands, as a port for passing unmodified data (from either the selected file register or the B bus) to the A bus. The specific arithmetic or logical operation performed by the ALU is determined by the microcommand in the R Register. The following operations can be performed:

a. Produce an all 1 byte.

b. Add contents of selected file register to selected operand on B bus.

c. Subtract operand on B bus from contents of selected file register (either one's or two's complement subtraction).

d. Shift contents of selected file register left one bit position and insert 1 or 0 in vacated bit position.

e. Increment contents of selected file register.

f. Decrement contents of selected file register.

g. Logical AND of selected file register and B bus operand.

h. Logical OR of selected file register and B bus operand.

i. Exclusive OR of selected file register and B bus operand.

The Initial Carry Logic determines whether a 1 or 0 is entered into the low-order bit position of the ALU during arithmetic operations. During an add operation, a 1 is added to the sum when initial carry is a 1. The initial carry is also set to 1 when the contents of the selected file register are to be incremented or decremented. During subtract operations, the state of initial carry determines whether a one's complement or two's complement subtraction is performed. During a shift operation, the state of initial carry determines whether a 1 or 0 is entered into the vacated bit position. In some cases, the state of the initial carry bit is determined unconditionally by the microcommand being executed. In other cases, the state of initial carry is determined by the state of the selected link register bit.

1.6.4 CONTROL MEMORY

The control memory is a 16-bit high-speed memory that is normally implemented with semiconductor bipolar read only memory (ROM) devices. In special applications, the control memory can be implemented by a read-write memory that provides an alterable control memory (ACM).

Standard systems can contain up to 2048 words of ROM, all of which are packaged on a single PC board. This board can contain any number of words between 256 and 2048, in increments of 256.

The execution and accessing of control memory microcommands are overlapped. That is, while one command is in the R Register being executed, the L Register is incremented to access the next command in the sequence. Access time, including logic delays, is less than 200 nanoseconds. Most commands require one 200 nanosecond microstep for execution. Commands that require more than one microstep for execution include jump-type commands (two or three microsteps required), skip-type commands (two microsteps required if the skip is taken), and core memory read/write operations (three or five microsteps).

1.6.5 CORE MEMORY

Core memory can be implemented with 8192-word modules, each of which is contained on a separate printed circuit board. Each word in core memory contains 8 bits. The M and N Registers provide the capability for addressing up to 65,536 words.

The core memory is addressed either by the M and N Registers in the CPU, or by the DMA channel. The DMA channel has priority when both the CPU and DMA are requesting service simultaneously. Basic memory timing for both the CPU and the DMA is controlled by the CPU.

Either half-cycle or full-cycle memory read/write operations can be executed by the CPU. The DMA is limited to full-cycle operations. For half-cycle operation, the memory cycle time is 600 nanoseconds (three microsteps). For full-cycle operations, the memory cycle time is 1000 nanoseconds (five microsteps). When a memory cycle is in progress, commands that specify the M or N Register are delayed until the cycle is terminated, since changing of the M or N Register contents would destroy the address of the core location being accessed. Operate-type commands that select the T Register are not executed at the first or second microstep of a memory read operation. This allows the memory read data to be placed in the T Register before these commands are executed. In this case, the command is executed when the memory cycle terminates.

1.6.6 REAL TIME CLOCK

The real time clock on the control board is enabled and disabled under microcommand control. When enabled, the real time clock produces an interrupt at one-millisecond intervals.

1.6.7 POWER FAIL AND RESTART FEATURE

The power fail and restart feature is a standard part of the CPU. When loss of ac power or a low voltage condition is detected by the power supply, an interrupt is produced. The power supply is capable of holding all voltages within the operating range for at least one millisecond after the interrupt occurs. During this period, the firmware responds by saving all volatile flags and registers in core memory and then executing a processor halt. When the processor halt occurs, a system reset is produced to provide for an orderly loss of power.

When power is restored following a power loss, a system reset is applied until all voltages are stabilized. After the reset terminates, an interrupt is issued, and the firmware responds by restoring all volatile registers and flags to their pre-fail states.

1.6.8 FILE ZERO FLAGS

As mentioned in Paragraph 1.6.1.4, file register zero is common to both the primary and secondary files, and contains a set of flags that reflect the results of previous operations and external conditions that require frequent testing. When designated by microcommand, the file zero flag byte is supplied to the ALU. Table 1-2 describes the contents of the file zero flag byte.

TABLE 1-2. FILE ZERO FLAG BYTE

BIT	DESCRIPTION
0	Overflow Condition: The overflow condition flag stores the overflow condition of an add, increment, decrement, compare, or shift operation. Overflow occurs when the carry-out from the high-order bit position of the ALU differs from the carry-in, or (for a shift operation) when the out-shifted bit is a 1.
1	Negative Condition: The negative condition flag stores the high-order bit of the result. When overflow occurs this flag will be the complement of the true sign.

TABLE 1-2. FILE ZERO FLAG BYTE (cont)

BIT	DESCRIPTION
2	Zero Condition: The zero condition flag stores the zero condition of the result. The zero test can be linked over multiple byte operations under control of the L modifier (bit 7) of operate instructions. When this bit is 1, the zero condition flag may not be set to indicate the zero condition of the current byte, but may only be reset to indicate a non-zero result. For this flag to indicate zero over multiple bytes it must be set by a zero result on the first operation which will have the L modifier zero and not be reset by non-zero conditions on succeeding bytes which will have the L modifier a one.
3	I/O Request: The I/O request flag is turned on by one or more external I/O units requesting an I/O operation.
4	Internal Interrupt: The internal interrupt flag is turned on when an internal interrupt condition is present. The internal interrupt is identified in the internal status byte (Table 1-3).
5	I/O Reply: The I/O reply flag is turned on by the external I/O unit currently communicating with the processor. This flag is normally not used.
6	Serial TTY or Stack Overflow: The serial teletype input flag indicates the state of the serial teletype input. A zero indicates that the input is in a MARK state. This flag is used for Micro 800 compatibility. The stack overflow flag is turned on when the L Save Stack has overflowed.
7	External Interrupt: The external interrupt flag is turned on by one or more external I/O units requesting an interrupt. This flag must result in a command which reads the address of the interrupt vector and resets the request.

1.6.9 INTERNAL STATUS BYTE

The internal status byte reflects the state of the internal interrupts. When an interrupt is present, bit 4 of the file zero flag byte is active. The firmware normally responds by executing an Enter Internal Status (EIS) microcommand, which places the internal status byte on the A bus. The panel, real time clock, and power fail interrupts are reset when the EIS command is executed. The format of the internal status byte is given in Table 1-3.

TABLE 1-3. INTERNAL STATUS

BIT	STATUS MEANING
0	Panel Interrupt
1	DMA Termination
2	Real Time Clock Interrupt
3	Spare
4	Spare
5	Spare
6	Panel Step Switch
7	Power Fail (Restart Interrupt)

1.7 MICROCOMMAND DESCRIPTIONS

Table 1-4 is a numerical listing of all microcommands in the CPU repertoire. The microcommands are grouped as follows for descriptive purposes:

a. Command Modification Class. Commands in this class produce modifications in the high- or low-order bits of the microcommand from control memory. That is, the effective microcommand stored in the R Register differs from the command supplied from control memory.

b. Jump and Return Jump Class. Commands in this class are associated with producing jumps in the L Register address used to access microcommands in control memory. The L Register, which normally accesses control memory locations sequentially, is loaded with a jump address.

c. Literal Class. Commands in this class contain an 8-bit literal in the low-order bits of the command. The eight high-order bits define the literal, and the actions performed with the literal. The literal class is divided into the following categories:

   1. Discrete-Function Literals. Each bit of the literal in commands of this type has a discrete function, permitting multiple operations to be executed by a single command.

   2. Load Literal in Selected Register. The literal is entered into the register specified by the high-order bits.

   3. Skip-Type Commands. The literal is tested against the contents of the selected file register. If the test is affirmative, the microcommand in the control memory location that follows the skip-type command is not executed.

TABLE 1-4. MICROCOMMAND REPERTOIRE

	MICROCOMMAND CODE (HEX)				COMMAND	MNEMONIC	EXECUTION TIME (NANOSECONDS)	DESCRIPTION (PARA. NO.)
	BITS 15-12	BITS 11-8	BITS 7-4	BITS 3-0				
	0	X	X	X	EXECUTE, LITERAL TYPE	ELT	200	1.7.1.1
	0	X	X	X	EXECUTE, OPERATE TYPE	EOT	200	1.7.1.2
	0	X	X	X	JUMP EXTENDED	JE	400	1.7.2.1
LOAD ZERO (LZ)	1	0	0	0	NO OPERATION	NOP	200	1.7.3.1.1.5
	1	0	0	1	ENABLE COMM. RATES	ECR	200	1.7.3.1.1.4
	1	0	0	2	DISABLE COMM. RATES	DCR	200	1.7.3.1.1.4
	1	0	0	4	INPUT COMM. RATES	ICR	200	1.7.3.1.1.3
	1	0	2	0	RETURN	RTN	600	1.7.2.4
	1	0	4	0	SELECT PRIMARY FILE	SPF	200	1.7.3.1.1.1
	1	0	6	0	RETURN, SELECT PRIMARY FILE	RSP	600	1.7.3.1.1.2
	1	0	8	0	SELECT SECONDARY FILE	SSF	200	1.7.3.1.1.1
	1	0	A	0	RETURN, SELECT SECONDARY FILE	RSS	600	1.7.3.1.1.2
	1	1	X	X	LOAD T	LT	200	1.7.3.2.1
	1	2	X	X	LOAD M	LM	200	1.7.3.2.2
	1	3	X	X	LOAD N	LN	200	1.7.3.2.3
	1	4,5,C, OR D	X	X	JUMP IN 1K	JP	400	1.7.2.2
	1	6	X	X	LOAD U	LU	200	1.7.3.2.4
LOAD SEVEN (LS)	1	7	0	0	NO OPERATION	NOP	200	1.7.3.1.2.4
	1	7	0	4	DISABLE EXTERNAL INTERRUPTS	DEI	200	1.7.3.1.2.3
	1	7	0	8	ENABLE EXTERNAL INTERRUPTS	EEI	200	1.7.3.1.2.3
	1	7	1	0	DISABLE REAL TIME CLOCK	DRT	200	1.7.3.1.2.2
	1	7	2	0	ENABLE REAL TIME CLOCK	ERT	200	1.7.3.1.2.2
	1	7	8	0	HALT	HLT	200	1.7.3.1.2.1
	1	8	X	X	LOAD EIGHT	LE	200	1.7.3.1.3
	1	9	X	X	RETURN, LOAD T	RLT	600	1.7.2.5
	1	A	0	0	MODIFY LOWER COMMAND	MLC	200	1.7.1.4
L SAVE COMMANDS	1	B	0	0	INHIBIT L SAVE	ILS	200	1.7.2.6
	1	B	0	1	INCREMENT STACK POINTER	ISP	200	1.7.2.6
	1	B	0	2	DECREMENT STACK POINTER	DSP	200	1.7.2.6
	1	B	0	4	CLEAR STACK POINTER	CSP	200	1.7.2.6
	1	B	0	8-F	BANK SELECT	BSL	200	1.7.2.6
	1	B	9	0	SELECT STACK POINTER	SSP	200	1.7.2.6
	1	B	A	0	SELECT STACK UPPER	SSU	200	1.7.2.6
	1	B	C	0	SELECT STACK LOWER	SSL	200	1.7.2.6.
	2	Z	X	X	LOAD FILE WITH LITERAL	LF	200	1.7.3.2.5
	3	Z	X	X	ADD TO FILE REGISTER	AF	200	1.7.3.2.6
	4	X	X	X	TEST IF ZERO	TZ	200*	1.7.3.3.1
	5	X	X	X	TEST IF NOT ZERO	TN	200*	1.7.3.3.2
	6	X	X	X	COMPARE FILE	CP	200*	1.7.3.3.3
OPERATE CLASS	7	X	1	X	ENTER SENSE SWITCHES	ESS	200**	1.7.4.1.1
	7	X	2	X	SHIFT FILE RIGHT FOUR	SRF	200**	1.7.4.2.13
	7	X	4	X	ENTER INTERNAL STATUS	EIS	200**	1.7.4.1.2
	7	0	7	0	ENTER CONSOLE SWITCHES	ECS	200	1.7.1.5
	7	X	8	X	CLEAR I/O MODE	CIO	200**	1.7.4.1.3
	7	X	9	X	CONTROL OUTPUT	COX	200**	1.7.4.1.3
	7	X	A	X	DATA OUTPUT	DOX	200**	1.7.4.1.3
	7	X	B	X	SERIAL TTY SPACE, OR SPARE	SOX	200**	1.7.4.1.3
	7	X	C	X	CONCURRENT ACKNOWLEDGE	CAK	200**	1.7.4.1.3
	7	X	D	X	INTERRUPT ACKNOWLEDGE	IAK	200**	1.7.4.1.3
	7	X	E	X	DATA INPUT	DIX	200**	1.7.4.1.3
	7	X	F	X	STACK INPUT OR SPARE	SIX	200**	1.7.4.1.3
	8	X	0,1	X	MOVE FILE	MOV	200**	1.7.4.2.10
	8	X	4,5	X	INCREMENT	INC	200**	1.7.4.2.5
	8	X	8,9	X	ADD LINK TO FILE	ALF	200**	1.7.4.2.4
	8	X	2,3,6,7,A-F	X	ADD	ADD	200**	1.7.4.2.1
	9	X	0,1	X	MOVE FILE	MOV	200**	1.7.4.2.10
	9	X	4,5	X	DECREMENT FILE	DEC	200**	1.7.4.2.6
	9	X	2,3,8-B	X	SUBTRACT FILE, TWO'S COMPLEMENT	SBT	200**	1.7.4.2.2
	9	X	6,7,C-F	X	SUBTRACT FILE, ONE'S COMPLEMENT	SBO	200**	1.7.4.2.2
	A	X	0,4,8,C	X	READ MEMORY, FULL CYCLE	RMF	1000	1.7.4.4
	A	X	2,6,A,E	X	READ MEMORY, HALF CYCLE	RMH	600	1.7.4.4
	A	X	1,5,9,D	X	WRITE MEMORY, FULL CYCLE	WMF	1000	1.7.4.4
	A	X	3,7,B,F	X	WRITE MEMORY, HALF CYCLE	WMH	600	1.7.4.4
	B	X	0,1	X	ZERO FILE	ZOF	200**	1.7.4.2.7
	B	X	4,5	X	PLUS ONE TO FILE	POF	200**	1.7.4.2.8
	B	X	8,9	X	COPY LINK TO REGISTER	CLN	200**	1.7.4.2.9
	B	X	2,3,6,7,A-F	X	COPY	CPY	200**	1.7.4.2.3
	C	X	0,1,8,9	X	MOVE FILE	MOV	200**	1.7.4.2.10
	C	X	2-5,A-D	X	LOGICAL OR	LOR	200**	1.7.4.3
	C	X	0,1,8,9	X	MOVE FILE	MOV	200**	1.7.4.2.10
	D	X	6,7,E,F	X	COMPLEMENT FILE	CF	200**	1.7.4.2.12
	D	X	2-5,A-D	X	EXCLUSIVE OR	XOR	200**	1.7.4.3
	D	X	0,1,8,9	X	ZERO FILE	ZOF	200**	1.7.4.2.7
	E	X	6,7,E,F	X	MOVE FILE	MOV	200**	1.7.4.2.10
	E	X	2-5,A-D	X	LOGICAL AND	AND	200**	1.7.4.3
	E	X	0,1,8,9	X	SHIFT FILE LEFT	SFL	200**	1.7.4.2.14
	F	X	4,5,C,D	X	SHIFT FILE LEFT AND INSERT	SLI	200**	1.7.4.2.14
	F	X	2,3,A,B	X	SHIFT FILE RIGHT	SFR	200**	1.7.4.2.14
	F	X	6,7,E,F	X	SHIFT FILE RIGHT AND INSERT	SRI	200**	1.7.4.2.14
	F	X	6,7,E,F	X	ENTER 225 INTO REGISTER	255	200**	1.7.4.2.11

*EXECUTION TIME IS 400 NANOSECONDS IF SKIP IS TAKEN
**EXECUTION TIME IS 400 NANOSECONDS WHEN r FIELD CONTAINS CODE 4_8 OR 5_8 (JUMP IN 1/2 K)

LEGEND

X = ANY VALUE BETWEEN 0 AND F
Z = ANY VALUE BETWEEN 1 AND F I AND F

d. Operate Class. All commands in this class have the same format. The operate class is divided into the following categories:

1. Input/Output Operations. An internal or external input/output operation is executed.

2. Arithmetic Operations. An add, subtract, move, copy, or shift operation is performed on the contents of the selected file and/or the operand on the B bus.

3. Logical Operations. An AND, OR, or Exclusive OR (XOR) operation is performed with the contents of the selected file and the operand on the B bus.

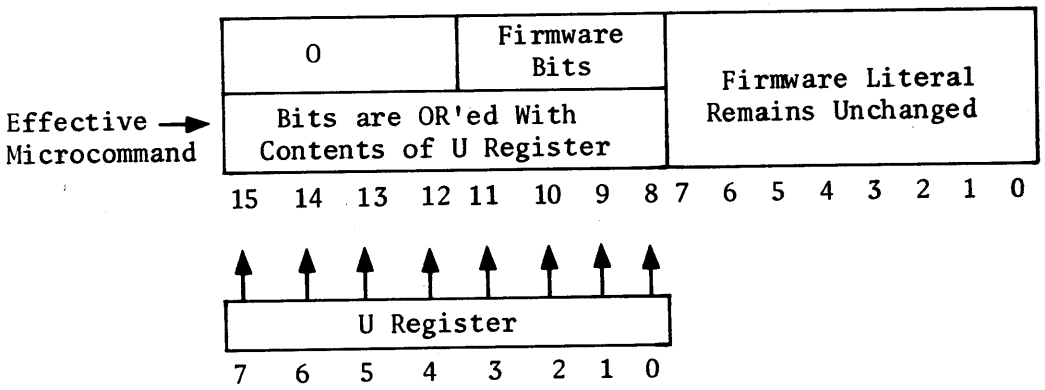
4. Core Memory Operations. A read or write operation is performed in core memory. In addition, an arithmetic operation is performed on the contents of the selected file.

Note

In the discussions that follow, all codes are given in hexadecimal form, unless otherwise specified.

1.7.1 COMMAND MODIFICATION CLASS

1.7.1.1 EXECUTE, LITERAL TYPE (ELT)



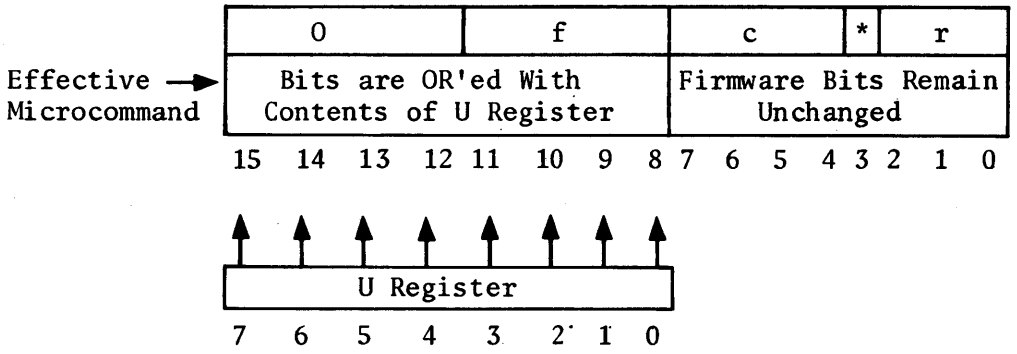
The Execute, Literal Type (ELT) command is used to produce an effective microcommand having op codes 1 through 6. Op code 0 is bits 15 through 8 of the ELT command causes the R Register to be loaded with an effective microcommand, as follows:

a. The op code in bits 15 through 12 is the code contained in the U registers bits 7 through 4.

b. The code in bits 11 through 8 is the result of OR'ed U register bits 3 through 0 with bits 11 through 8 of the ELT command.

c. The literal in bits 7 through 0 is the same as the literal in the ELT command.

1.7.1.2 EXECUTE, OPERATE TYPE (EOT)



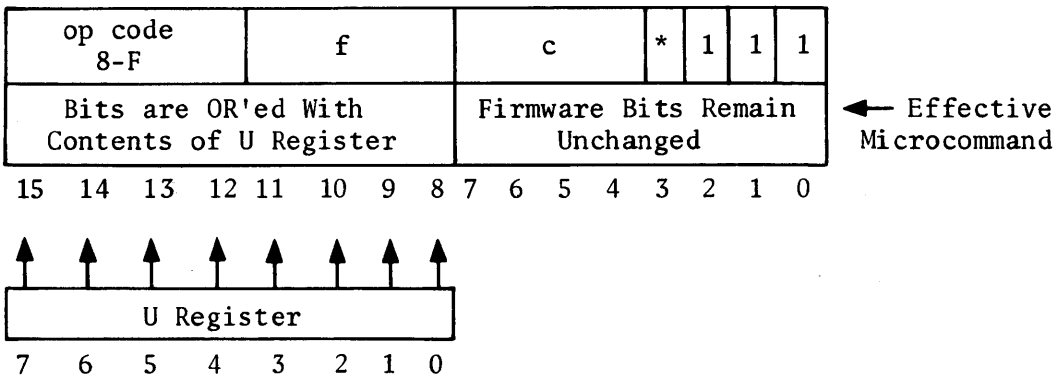
The Execute, Operate Type (EOT) command is used to produce an effective microcommand having op codes 7 through F. The op code 0 in bits 15 through 8 of the EOT command causes the R Register to be loaded with an effective microcommand, as follows:

a. The op code in bits 15 through 12 is the code contained in U register bits 7 through 4.

b. The f field (bits 11 through 8) contains the result of OR'ed U register bits 3 through 0 with bits 11 through 8 of the EOT command.

c. The c and r fields in bits 7 through 0 are the same as those contained in the EOT command.

1.7.1.3 MODIFY HIGH-ORDER BITS OF OP CODES 8 THROUGH F



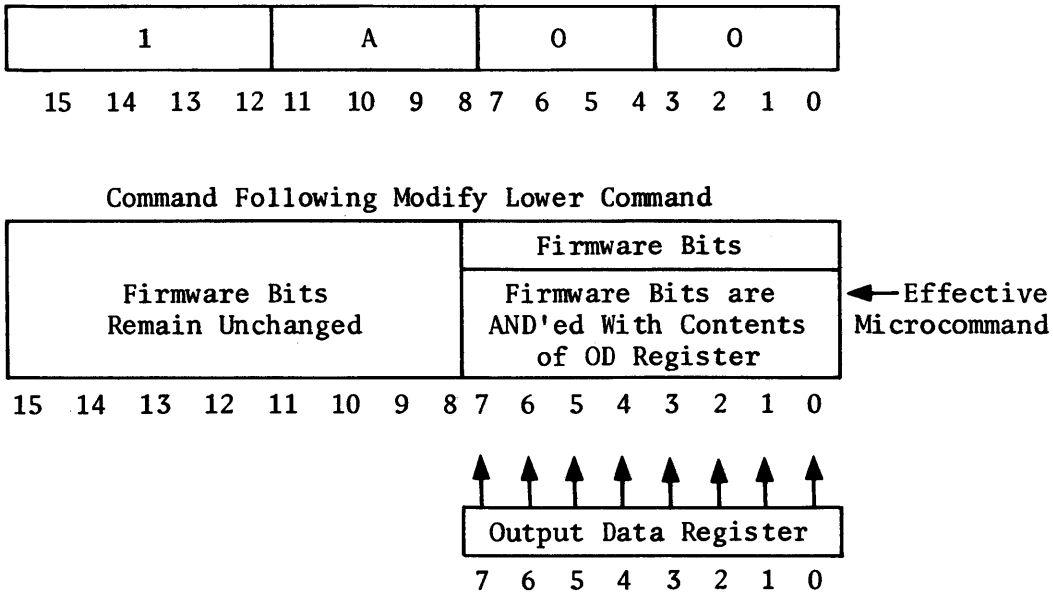
The eight high-order bits of any operate-class microcommand with op codes 8 through F are modified by the U Register when the r field of the command contains code 7_8 in r field bits 3 through 0. The effective microcommand loaded into the R Register consists of the following:

a. The op code in bits 15 through 12 is the result of OR'ed U register bits 7 through 4 with bits 15 through 12 of the command.

b. The f field in bits 11 through 8 is the result of OR'ed U register bits 3 through 0 with the f field of the command.

c. The c and r fields (bits 7 through 0) are unchanged.

1.7.1.4 MODIFY LOWER COMMAND (MLC)

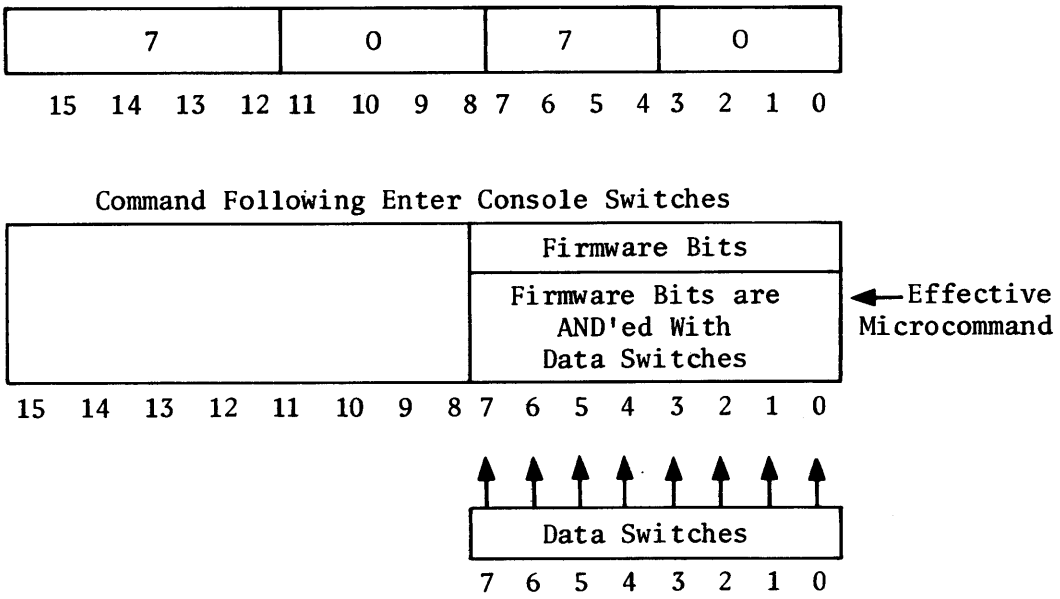


The Modify Lower Command (MLC) is used to modify the eight low-order bits of the command that follows the MLC command in control memory. The resulting effective microcommand loaded into the R Register is formed as follows:

a. Bits 15 through 8 are unchanged.

b. Bits 7 through 0 are the result of combining bits 7 through 0 of the output data (OD) register with bits 7 through 0 of the command.

1.7.1.5 ENTER CONSOLE SWITCHES (ECS)

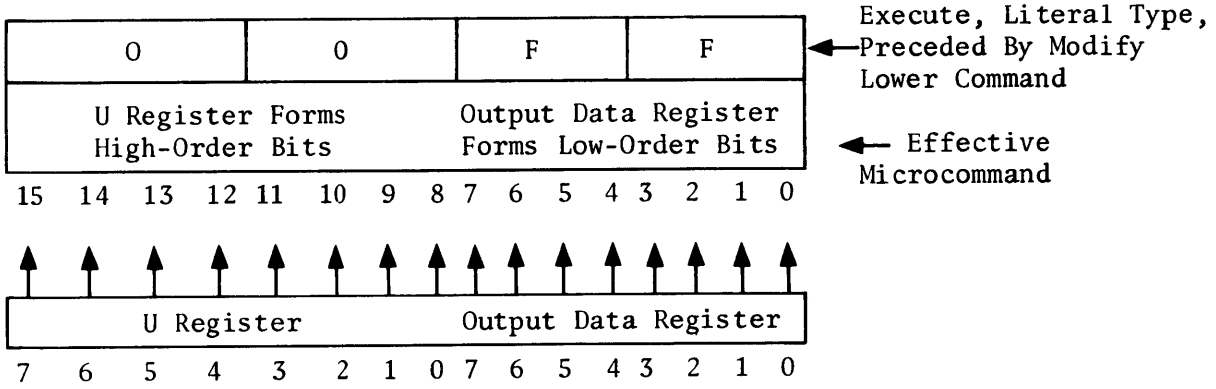


The Enter Console Switches (ECS) command is used to modify the eight low-order bits of the command that follows the ECS command in control memory. The resulting effective microcommand loaded into the R Register is formed as follows:

a. Bits 15 through 8 are unchanged.

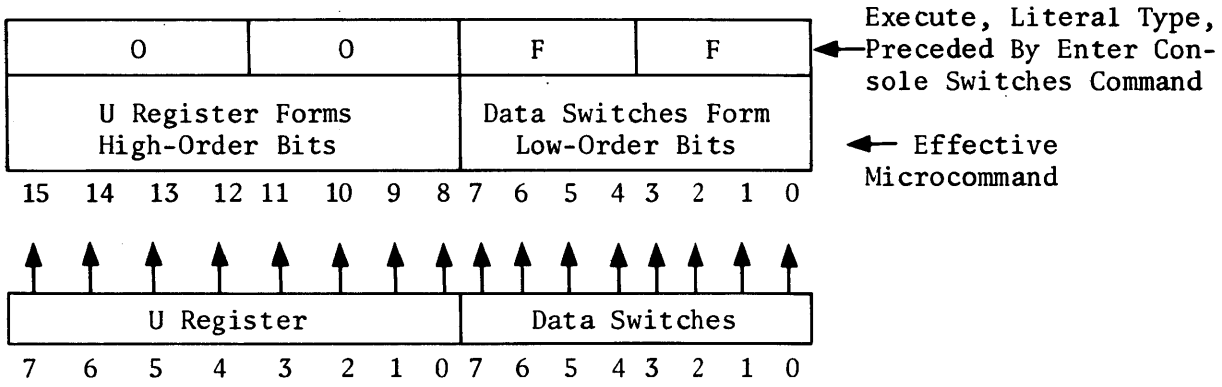
b. Bits 7 through 0 are the result of combining the states of front panel data switches 7 through 0 with bits 7 through 0 of the command.

1.7.1.6 FORMATION OF EFFECTIVE MICROCOMMAND FROM U AND OD REGISTERS



By executing a Modify Lower (MLC) command, followed by an Execute, Literal Type (ELT) command containing code OOFF, an effective microcommand is formed from the contents of the U and OD Registers. The U Register forms the eight high-order bits, and the OD (output data) Register forms the eight low-order bits.

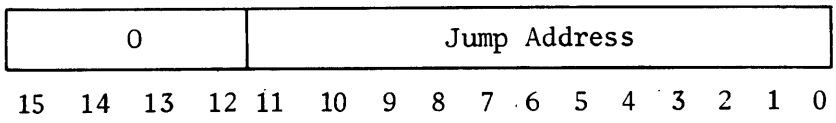
1.7.1.7 FORMATION OF EFFECTIVE MICROCOMMAND FROM U REGISTER AND DATA SWITCHES



By executing an Enter Console Switches (ECS) command, followed by an Execute, Literal Type (ELT) command containing code OOFF, an effective microcommand is formed from the contents of the U Register and front panel data switches 7 through 0. The U Register forms the eight high-order bits, and the front panel data switches form the eight low-order bits.

1.7.2 JUMP AND RETURN JUMP CLASS

1.7.2.1 JUMP EXTENDED (JE)

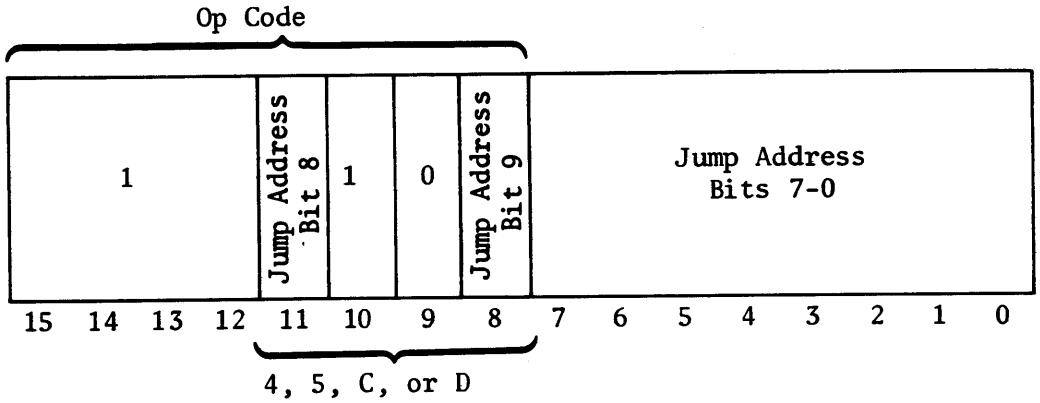


The Jump Extended (JE) command is actually a special version of the Execute, Literal Type (ELT) command in which the U Register contains all zeros. Consequently, the command loaded into the R Register is unchanged from the command received from control memory. Op code 0 causes the L Register to be loaded with the 12-bit address in JE bits 11 through 0. This permits a jump to any desired location in control memory. If L Save is not inhibited by an earlier Inhibit L Save (ILS) command, the

address in the L Register (loaded prior to the jump address) is transferred to the L Save Register (or the L Save Stack Option). This permits a subsequent return to this address.

For correct execution of the Jump Extended command, U register bits 7 through 4 must all be in the zero state. Initially, U register bits 3 through 0 are also set to zero, however, U register bits 3 through 0 may be used to modify the four high-order bits of the jump address in the command.

1.7.2.2 JUMP IN 1K (JP)

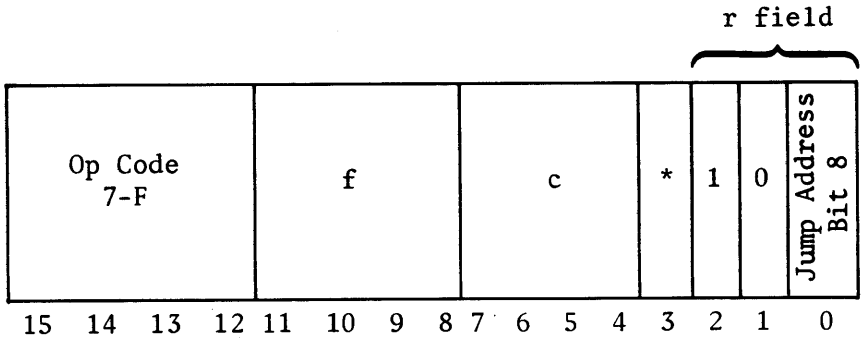


The Jump in 1K (JP) command contains a 10-bit jump address that is loaded into bits 9 through 0 of the L Register. The two high-order bits of the L Register remain unchanged, thus the Jump in 1K command permits a jump to any desired address within a 1024-word segment of control memory. This 1024-word segment is divided into four 256-word sub-segments, defined by the op code in bits 15 through 8 of the command as follows:

Op Code	Sub-Segment
14	Lowest 256 words
15	Second-Lowest 256 words
1C	Second-Highest 256 words
1D	Highest 256 words

The specific location within the selected 256-word sub-segment is selected by the address in bits 7 through 0 of the command.

1.7.2.3 JUMP IN 1/2K

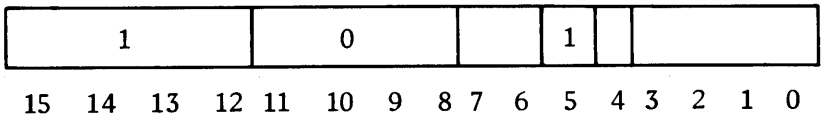


Any operate-class microcommand with op code 7 through F that contains code 4_8 or 5_8 in the r field produces a jump in 1/2K. The L Register is loaded with a 9-bit jump address. The three high-order bits of the L Register remain unchanged, thus the jump in 1/2K permits a jump to any desired address within a 512-word segment of control memory. This 512-word segment is divided into two 256-word sub-segments, defined by the code in the r field as follows:

r Field Code	Sub-Segment
4_8	Lower 256 words
5_8	Upper 256 words

Bit 0 of the command becomes bit 8 of the jump address. The eight low-order bits (bits 7-0) of the jump address are placed on the A bus as the result of the operate-class command execution. These eight bits are then entered into the eight low-order bits of the L Register to specify the desired core memory location within the selected 256-word sub-segment.

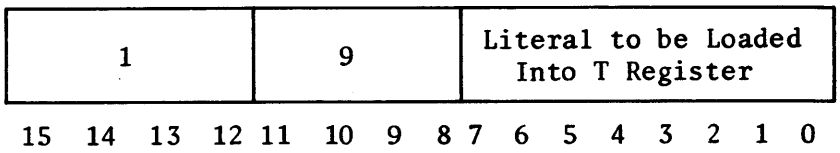
1.7.2.4 RETURN (RTN)



A Load Zero (LZ) command (op code 10) containing a 1 in bit 5 initiates a return jump (RTN). First, the contents of the L Save Register (or the L Save Stack Option) are temporarily stored in bits 11 through 0 of the R Register. Also, the current

address in the L Register is entered into the L Save Register (or the L Save Stack Option) to permit a subsequent return. Then, the L Save address in the R Register is loaded into the L Register, causing a return jump to any location within the control memory.

1.7.2.5 RETURN, LOAD T (RLT)



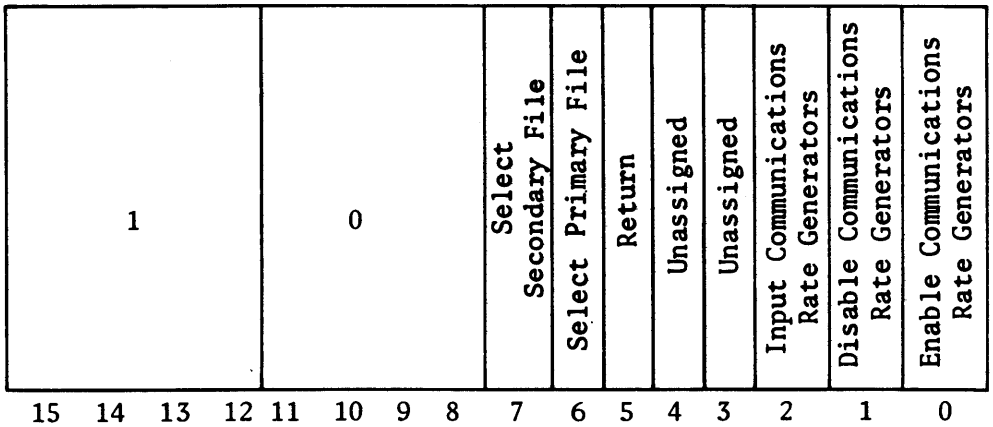
A Return, Load T (RLT) produces the same return jump actions that were performed for the Return (RTN) command. In addition, the literal contained in bits 7 through 0 of the command is placed in the T Register.

1.7.3 LITERAL CLASS

The following paragraphs describe the various categories of commands in the literal class.

1.7.3.1 DISCRETE-FUNCTION LITERALS. The following paragraphs describe command formats for discrete-function literals.

1.7.3.1.1 Load Zero (LZ). The Load Zero command has the following format:



The 8-bit literal field of the Load Zero (LZ) command is used to specify single or combined operations associated with file selection, return jumps and control of optional communications rate generators. A single LZ command in a vertical sequence

of microprogramming contains horizontal microprogramming characteristics because of the ability to perform separate and distinct actions in a single microstep. Bits 4 and 3 of the LZ literal are unassigned.

1.7.3.1.1.1 Primary/Secondary File Selection (SPF and SSF). Bits 7 and 6 of the LZ literal control the selection of the primary or secondary file. The primary file is always selected at power on, or after a master reset. Subsequently, an LZ command with the appropriate code in bits 7 and 6 must be executed to change the file selection. Only the registers in the selected file are available to the CPU. The file selection codes are as follows:

Bit 7	Bit 6	Selected File	Mnemonic
0	0	No Change	None
0	1	Primary File	SPF
1	0	Secondary File	SSF
1	1	Change to Opposite File	None

1.7.3.1.1.2 Return Jump (RTN). When bit 5 of the LZ literal is a 1, a return jump is executed (RTN), as described in Paragraph 1.7.2.4. When either bit 7 or bit 6 is also a 1, the following combined commands are formed.

Bit 7	Bit 6	Bit 5	Command	Mnemonic
0	1	1	Return, Select Primary File	RSP
1	0	1	Return, Select Secondary File	RSS

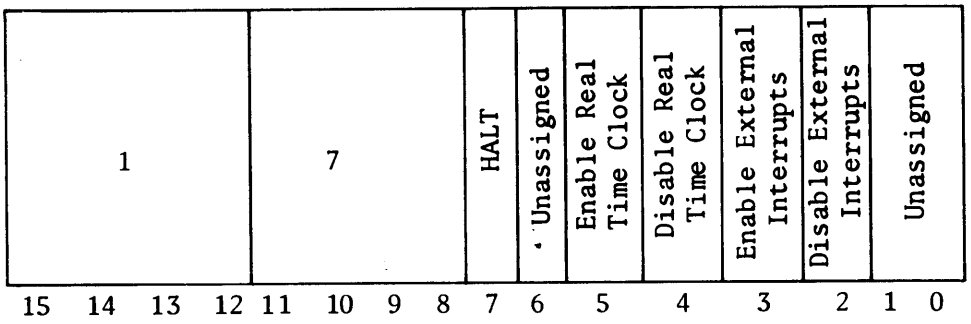
1.7.3.1.1.3 Input Communication Rate Generators (ICR). Bit 2 of the LZ literal controls interrupts by the optional/communication rate generators. When bit 3 is a 1 (ICR command), the communication rate generators are enabled, and one or more of the communication rates is requesting service, the internal status interrupt is activated, setting file zero bit 3 to a 1. The command is followed immediately by an Enter Internal Status (EIS) command, causing the four communication rate generator flags to be read and reset. Bits 5 through 2 of the internal status byte contain the current states of the four generators. Bits 7, 6, 1, and 0 of the internal status byte should be ignored.

1.7.3.1.1.4 Enable/Disable Communication Rate Generators (ECR and DCR). Bits 1 and 0 of the LZ literal control the enabling and disabling of the optional communication rate generators. The enable/disable codes are as follows:

Bit 1	Bit 0	State of Communication Rate Generators	Mnemonic
0	0	No Change	None
0	1	Enabled	ECR
1	0	Disabled	DCR
1	1	Change to Opposite State	None

1.7.3.1.1.5 No Operation (NOP). An LZ literal containing all zeros produces no operation (NOP), and can be used to insert a delay of 200 nanoseconds (one microstep).

1.7.3.1.2 Load Seven (LS). The Load Seven class of commands have the following formats:



The 8-bit literal field of the Load Seven (LS) command is used to specify single or combined operations associated with a processor HALT, control of the Real Time Clock, and control of External Interrupts. A single LS command in a vertical sequence of microprogramming contains horizontal microprogramming characteristics because of the ability to perform separate and distinct actions in a single microstep. Bits 6, 1, and 0 of the LS literal are unassigned and unavailable.

1.7.3.1.2.1 Processor Halt (HLT). When bit 7 of the LS literal is a 1, the processor is halted, stopping all microcommand execution. If the Direct Memory Access Channel (DMA) is being serviced when a HALT occurs, the DMA continues until its operation is completed.

1.7.3.1.2.2 Enable/Disable Real Time Clock (ERT and DRT). Bits 5 and 4 of the LS literal control the enabling and disabling of the real time clock. The enable/disable codes are as follows:

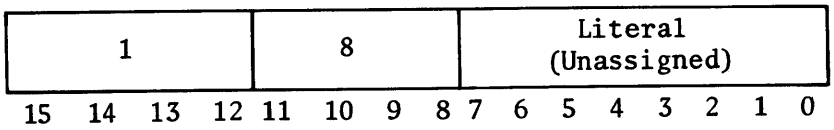
Bit 5	Bit 4	Status of Real Time Clock	Mnemonic
0	0	No Change	None
0	1	Disable	DRT
1	0	Enable	ERT
1	1	Change to Opposite State	None

1.7.3.1.2.3 Enable/Disable External Interrupts (EEI and DEI). Bits 3 and 2 of the LS literal control the enabling and disabling of external interrupts. The enable/disable codes are as follows:

Bit 3	Bit 2	Status of External Interrupts	Mnemonic
0	0	No Change	None
0	1	Disable	DEI
1	0	Enable	EEI
1	1	Change to Opposite State	None

1.7.3.1.2.4 No Operation (NOP). An LS literal containing all zeros produces no operation (NOP), and can be used to insert a delay of 200 nanoseconds (one microstep).

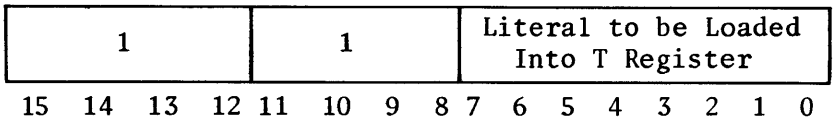
1.7.3.1.3 Load Eight (LE). The Load Eight command has the following format:



The Load Eight (LE) command is not implemented in the basic CPU. The LE command can be defined for special user applications, employing the discrete-function literal techniques employed for the Load Zero (LZ) and Load Seven (LS) commands.

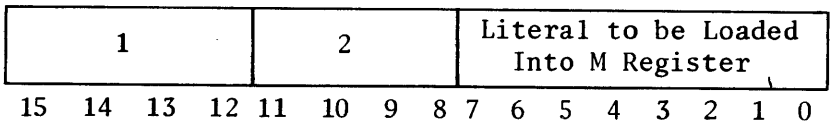
1.7.3.2 LOAD LITERAL IN SELECTED REGISTER. The following paragraphs describe command formats for load literal in selected register.

1.7.3.2.1 Load T (LT). The Load T command has the following format:



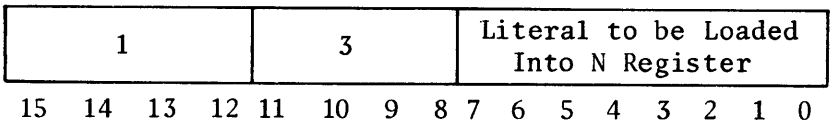
In a Load T (LT) execution, the contents of the 8-bit literal field are placed in the T Register. The condition code flags and link are unaffected. The same operation, plus a return jump, is performed for the Return, Load T (RLT) command (refer to Paragraph 1.7.3.5).

1.7.3.2.2 Load M (LM). The Load M command has the following format:



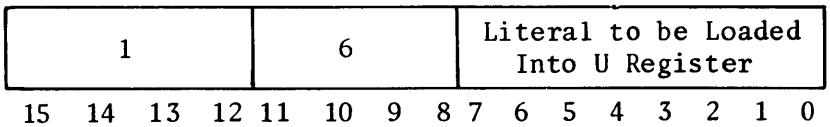
In a Load M (LM) execution, the contents of the 8-bit literal field are placed in the M Register. If a core memory read or write operation is in progress (memory busy), the M Register is not loaded until the memory operation is completed. The condition code flags and link are unaffected.

1.7.3.2.3 Load N (LN). The Load N command has the following format:



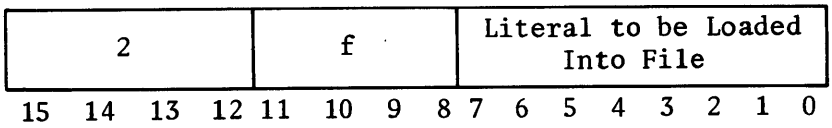
In a Load N (LN) execution, the contents of the 8-bit literal field are placed in the N Register, and the M Register is cleared. If a core memory read or write operation is in progress (memory busy), execution of Load N is delayed until the memory operation is completed. The condition code flags and link are unaffected.

1.7.3.2.4 Load U (LU). The Load U command has the following format:



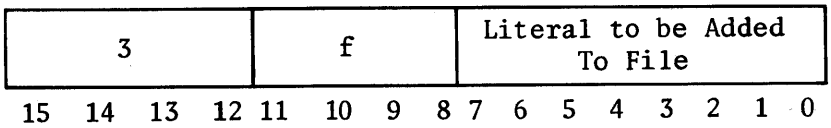
In a Load U (LU) execution, the contents of the 8-bit literal field are placed in the U Register. The condition code flags and link are unaffected. Due to the look-ahead access method used for the control memory, the new contents of the U Register are not available for use in command modification during the machine cycle that immediately follows execution of the Load U command.

1.7.3.2.5 Load File (LF). The Load File command has the following format:



In a Load File (LF) execution, the contents of the 8-bit literal field are placed in the file register designated by the f field (bits 11 through 8). Since file zero cannot be used for storage, it is not loaded by this command. The condition code flags and link are unaffected.

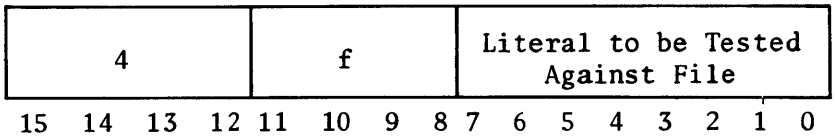
1.7.3.2.6 Add To File Register (AF). The Add To File Register command has the following format:



In an Add To File Register (AF) execution, the contents of the 8-bit literal field are added to the contents of the file register designated by the f field (bits 11 through 8). Since file 0 cannot be used for storage, it cannot be altered by this command. The condition code flags and link are unaffected. The AF command can also be used to perform a two's complement subtraction of the literal from the contents of the file register. In this case, the two's complement of the operand is placed in the literal field.

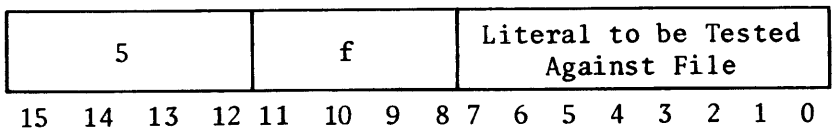
1.7.3.3 SKIP-TYPE COMMANDS

1.7.3.3.1 Test If Zero (TZ). The Test If Zero command has the following format:



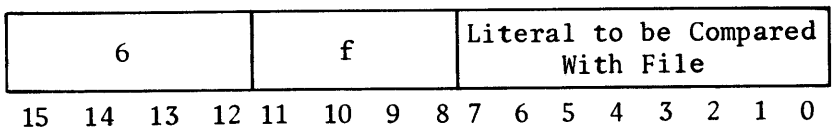
In a Test If Zero (TZ) execution, the 8-bit literal field is combined with the contents of the file register designated by the f field (bits 11 through 8). Consequently, the result of the logical AND (placed on the A bus) contains a 1 in each position where the corresponding bits of the literal field and the file register both contained 1's, and contains a 0 in each position where either the literal bit or the file register bit is a 0. If the result contains all zeros, the command that immediately follows the TZ command in control memory is skipped (not executed). The contents of the designated file register, the condition code flags, and the link are all unaffected by the TZ command.

1.7.3.3.2 Test If Not Zero (TN). The Test If Not Zero command has the following format:



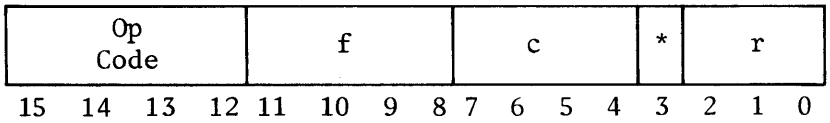
In a Test If Not Zero (TN) execution, the 8-bit literal field is combined with the contents of the file register designated by the f field (bits 11 through 8), as in the TZ command. If the result on the A bus contains a 1 in any bit position, the command that immediately follows the TN command in control memory is skipped (not executed). The contents of the designated file register, the condition code flags, and the link are all unaffected by the TN command.

1.7.3.3.3 Compare File (CP). The Compare File command has the following format:



In a Compare File (CP) execution, the 8-bit literal field and the contents of the file register designated by the f field (bits 11 through 8) are summed. If the sum is greater than 255, the arithmetic link register is set to 1, and the next command is skipped (not executed). The contents of the designated file register, and the condition code flags, are unaffected by the CP command.

1.7.4 OPERATE CLASS



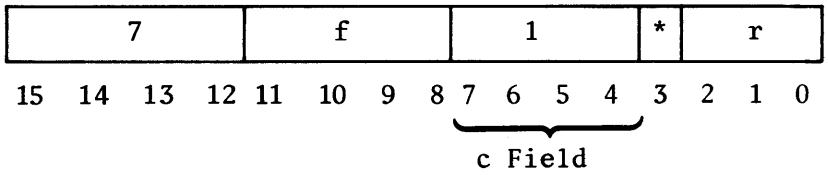
All commands in this class (op codes 7 through F) have the format shown above. The op code field (bits 15 through 12), combined with the c field (bits 7 through 4) specify the operation to be performed. The f field (bits 11 through 8) specifies the file register whose contents are involved in the operation. If the op code field does not produce a file inhibit, the contents of the file register designated by the f field are presented to the ALU. If bit 3 (file write inhibit) is a 0, the result on the A bus is written into the file register designated by the f field. When bit 3 is a 1, no entry into file is made. The symbol for bit 3 is an asterisk (*). The r field (bits 2 through 0) specifies the register that is to be loaded with the result on the A bus (codes 1_8 through 6_8 in r field). For op codes 8 through F, code 7_8 in the r field produces a command modification, as described in Paragraph 1.7.2.3. The r field codes and their definitions are as follows:

Bit 2	Bit 1	Bit 0	Octal Code	Register Selected
0	0	0	0	None
0	0	1	1	T Register
0	1	0	2	M Register*
0	1	1	3	N Register*
1	0	0	4	L Register (jump in 1/2K, as described in Paragraph 1.7.3.3)
1	0	1	5	L Register (jump in 1/2K, as described in Paragraph 1.7.3.3)
1	1	0	6	U Register
1	1	1	7	R Register (modify high-order bits of op codes 8 through F, as described in Paragraph 1.7.2.3)

*If the core memory is busy, command execution is delayed until the memory operation is completed.

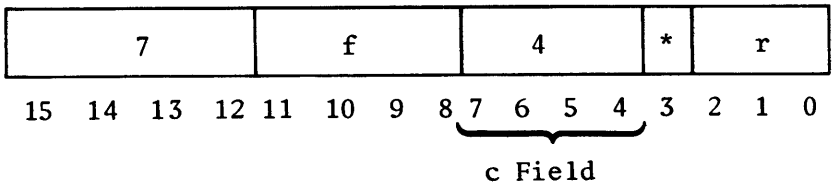
1.7.4.1 INPUT/OUTPUT OPERATIONS. The following paragraphs describe command formats for input/output operations.

1.7.4.1.1 Enter Sense Switches (ESS). The Enter Sense Switches command has the following format:



In an Enter Sense Switches (ESS) execution, the states of the four front panel sense switches are entered into the four high-order bits on the A bus, and 1's are entered in the four low-order bits. The sense switch data on the A bus is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field.

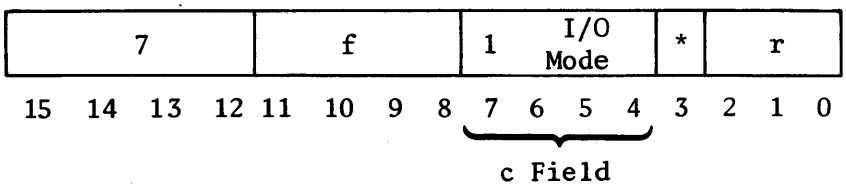
1.7.4.1.2 Enter Internal Status (EIS). The Enter Internal Status command has the following format:



In an Enter Internal Status (EIS) execution, the states of the eight internal status lines are placed on the A bus. The internal status byte on the A bus is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field. In addition, the following internal interrupts are reset:

a. Real Time Clock Interrupt
b. Power Fail/Restart Interrupt
c. Console (panel) Interrupt

1.7.4.1.3 External I/O (EXIO). The external I/O command has the following format:



An external I/O (EXIO) command is defined when the Op Code field contains a 7, and a 1 is contained in bit 7 of the c field. The type of external I/O operation is then defined by the 3-bit code (I/O Mode) in bits 6 through 4 of the c field. This 3-bit code is placed in the IC Register, which supplies the code to external I/O controllers on the byte I/O bus. The c field codes, the corresponding I/O Mode code, and the standard definitions are as follows:

I/O Type	c Field Code (Hex)	I/O Mode Code (Octal)	Description of I/O Operation	Mnemonic
Clear (no activity)	8	0	Clear I/O Mode - This code places the I/O bus in the no activity state. All standard I/O controllers require a return to the no activity state after a non-zero I/O Mode code has been placed in the IC Register.	CIO
Output	9	1	Control Output - This code notifies the I/O controller that a device order is on the output data bus.	COX
	A	2	Data Output - This code notifies the I/O controller that a data byte is on the output data bus.	DOX
	B	3	Spare	SOX
	C	4	Concurrent Acknowledge - This code notifies the I/O controller that the CPU has acknowledged a concurrent I/O request.	CAK
	D	5	Interrupt Acknowledge - This code notifies the I/O controller that the CPU has acknowledged an interrupt request.	IAK
	E	6	Data Input - This code commands the I/O controller to place a data byte on the input data bus to the CPU.	DIX
	F	7	Stack Input - If the CPU is equipped with the L Save Stack Option, this code commands the L Save Stack to place a stack input on the input data bus to the CPU. If the CPU does not include the L Save Stack Option, this code is spare.	SIX

1.7.4.1.4 File and Register Storage Actions For EXIO Commands. When an EXIO command is executed, the previous state of IC register bit 3 determines the data that

is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field. For standard I/O controllers, three possible conditions can exist, as follows:

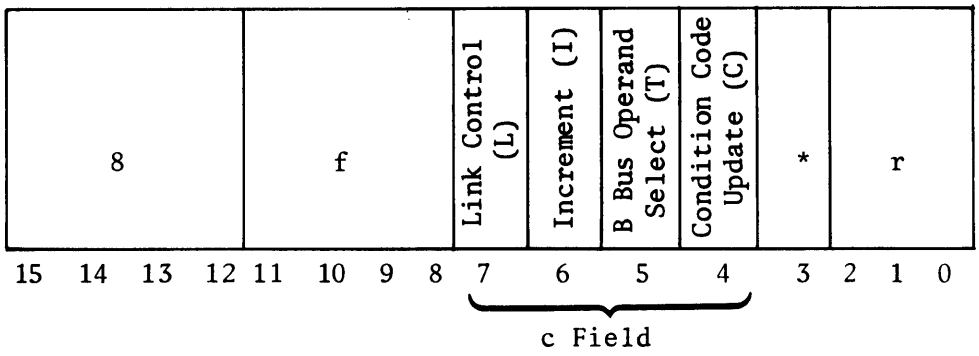
a. The IC Register was in a cleared state before the EXIO command is executed. In this case, IC register bit 3 is 0, and the contents of the file register designated by the f field are placed in the register designated by the r field.

b. The IC Register contained the I/O mode code for a clear or output-type command (c field codes 8 through B) before the EXIO command is executed, and the EXIO command is Clear I/O Mode (CIO). In this case, IC register bit 3 is a 0, and the contents of the file register designated by the f field are placed in the register designated by the r field.

c. The IC Register contained the I/O mode code for an input-type command (c field codes C through F) before the EXIO command was executed, and the EXIO command is Clear I/O Mode (CIO). In this case, IC register bit 3 is a 1, and the bits on the input data bus are combined with the contents of the file register designated by the f field. The logical product (on the A bus) is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field.

1.7.4.2 ARITHMETIC OPERATIONS. The following paragraphs describe command formats for arithmetic operations.

1.7.4.2.1 Add. The Add command has the following format:



In the basic Add operation, the selected operand (placed on the B bus) is added to the contents of the file register designated by the f field. The sum (on the A bus) is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field. Also, the selected link register is updated by

the state of the carry-out from the high-order bit in the ALU. The definitions of the bits in the c field are as follows:

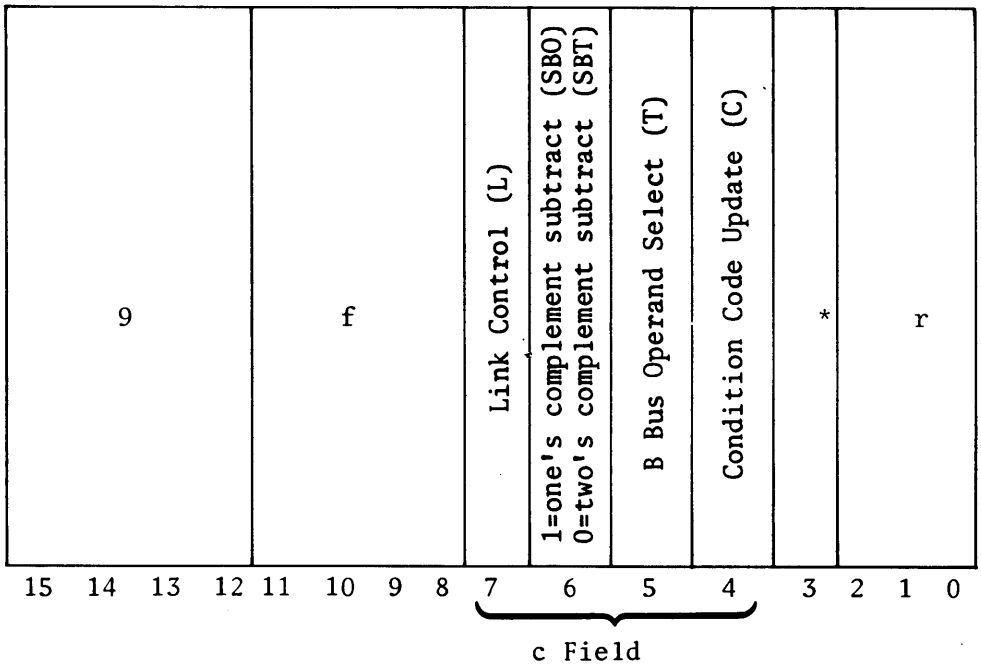
L (Bit 7) - Link Control: When bit 7 is a 1, the previous state of the selected link register is added to the sum (if bit 6 is a 0). Also, updating of the zero condition code flag is inhibited unless the flag is already set.

I (Bit 6) - Increment: When bit 6 is a 1, a one is added to the sum.

T (Bit 5) - Select Operand: When bit 5 is a 1, the contents of either the T Register or the input data bus are selected as the operand on the B bus. The T Register is selected if IC register bit 3 is 0. The input data bus is selected if IC register bit 3 is a 1. When bit 5 is a 0, an all-zero operand is selected.

C (Bit 4) - Condition Code Flag Update: When bit 4 is a 1, the condition code flags are updated according to the sum on the A bus.

1.7.4.2.2 Subtract (SBT or SBO). The Subtract command has the following format:



In the Subtract (SBT or SBO) operation, the selected operand (placed on the B bus) is subtracted from the contents of the file register designated by the f field. The difference (on the A bus) is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field. Also, the selected link register is updated by the state of the carry-out from the high-order bit in the ALU.

The subtraction operation is actually performed by complementing the operand selected for placement on the B bus, and then adding. If bit 6 of the c field is a 0, the result is a two's complement difference (SBT). If bit 6 is a 1, the result is a one's complement difference (SBO).

The definitions of the c field bits are as follows:

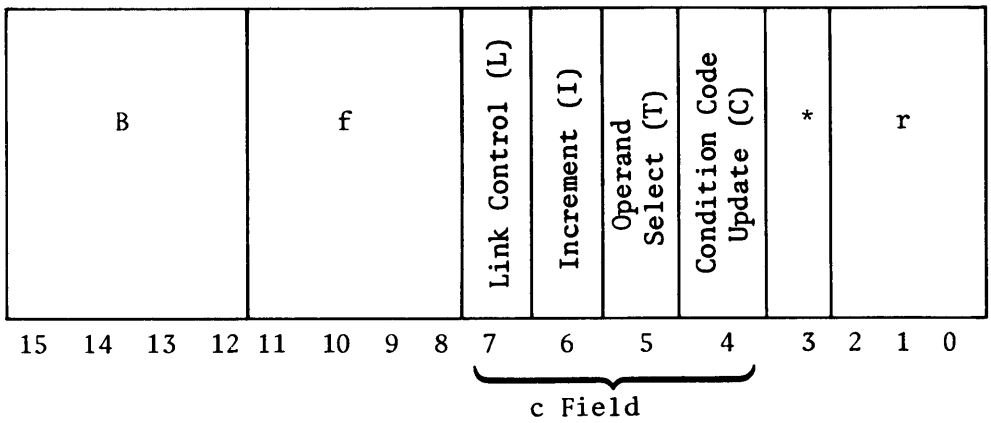
L (Bit 7) - Link Control: When bit 7 is a 1, the previous state of the selected link register is added to the difference (if bit 6 is a 0). Also, updating of the zero condition code flag is inhibited unless the flag is already set.

Bit 6 - Subtract Control: When bit 6 is a 0, a two's complement subtraction (SBT) is performed; when bit 6 is a 1, a one's complement subtraction (SBO) is performed.

T (Bit 5) - Select Operand: When bit 5 is a 1, the contents of either the T Register or the input data bus are selected as the operand. The T Register is selected if IC register bit 3 is a 0. The input data bus is selected when IC register bit 3 is a 1. When bit 5 is a 0, an all-zero operand is selected (all ones are placed on the B bus).

C (Bit 4) - Condition Code Flag Update: When bit 4 is a 1, the condition code flags are updated according to the difference on the A bus.

1.7.4.2.3 Copy (CPY). The Copy command has the following format:



In the Copy (CPY) operation, the selected operand is placed on the A bus and entered in the register designated by the r field, and (if * is 0) in the file register

designated by the f field. The Link Register is unaffected. The definitions of the bits in the c field are as follows:

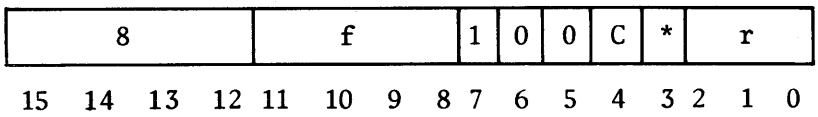
L (Bit 7) - Link Control: When bit 7 is a 1, the previous state of the selected link register is added to the selected operand being copied (if bit 6 is a 0). Also, updating of the zero condition code flag is inhibited unless the flag is already set.

I (Bit 6) - Increment: When bit 6 is a 1, a one is added to the selected operand being copied.

T (Bit 5) - Select Operand: When bit 5 is a 1, the contents of either the T Register or the input data bus are selected as the operand. The T Register is selected if IC register bit 3 is a 0. The input data bus is selected if IC register bit 3 is a 1. When bit 5 is a 0, an all-zero operand is selected.

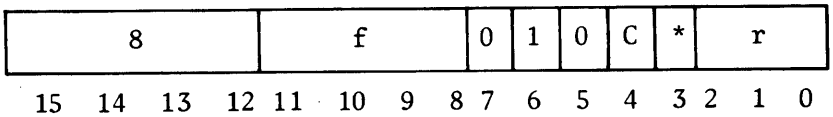
C (Bit 4) - Condition Code Flag Update: When bit 4 is a 1, the condition code flags are updated according to the operand on the A bus.

1.7.4.2.4 Add Link To File (ALF). The Add Link To File command has the following format:



The Add Link To File (ALF) command is a form of the Add command. The state of the selected link bit is added to the contents of the file register designated by the f field. The result is placed in the register designated by the r field and (if * is 0) in the file register designated by the f field. The selected link register is updated with the state of the carry-out from the high-order bit in the ALU. Updating of the zero condition code flag is inhibited unless the flag is already set.

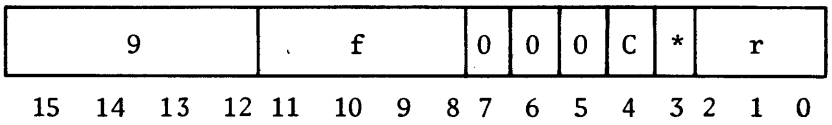
1.7.4.2.5 Increment (INC). The Increment command has the following format:



The Increment (INC) command is a form of the Add command. The contents of the file register designated by the f field are incremented by one. The result is placed in the register designated by the r field, and (if * is 0) in the file register

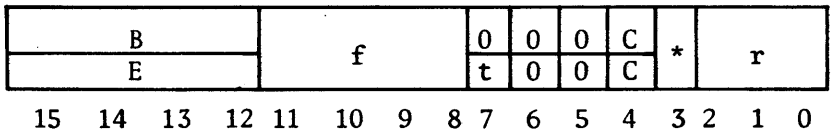
designated by the f field. The selected link register is updated with the state of the carry-out from the high-order bit in the ALU.

1.7.4.2.6 Decrement (DEC). The Decrement command has the following format:



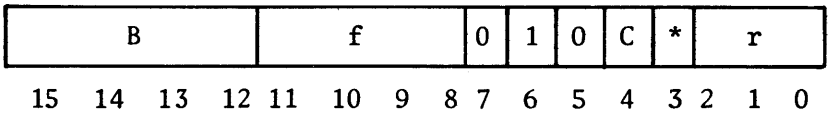
The Decrement (DEC) command is a form of the Subtract (SBO) command. The contents of the file register designated by the f field are decremented by one. The result is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field. The selected Link Register is updated with the state of the carry-out from the high-order bit in the ALU.

1.7.4.2.7 Zero File (ZOF). The Zero File command has the following format:



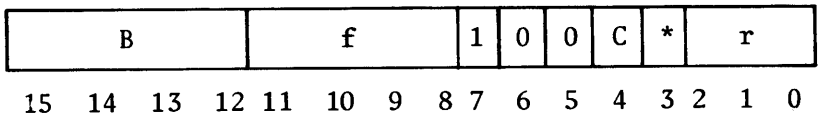
The Zero File (ZOF) command is a form of the Copy (CPY) or AND command. An all-zero byte is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field. The Link Register is unaffected.

1.7.4.2.8 Plus One To File (POF). The Plus One To File command has the following format:



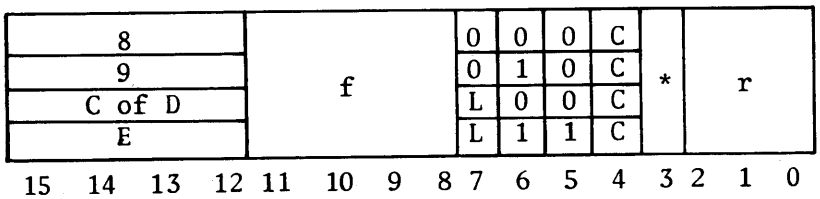
The Plus One To File (POF) command is a form of the Copy (CPY) command. A value of one is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field. The Link Register is unaffected.

1.7.4.2.9 Copy Link To Register (CLN). The Copy Link To Register command has the following format:



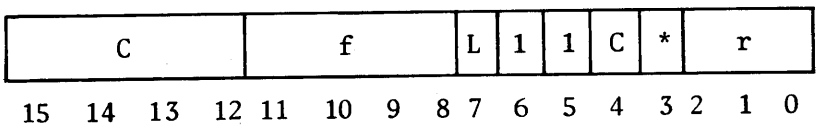
The Copy Link To File (CLF) command is a form of the Copy (CPY) command. The current state of the selected link bit is entered into the LSB of the register selected by the r field, and (if * is 0) in the file register selected by the f field. The Link Register is unaffected.

1.7.4.2.10 Move File (MOV). The Move File command has the following format:



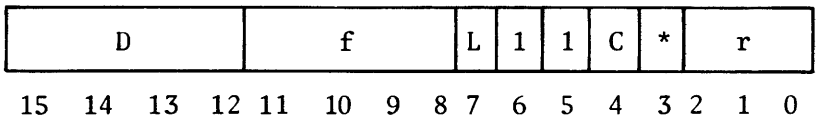
The Move File (MOV) command is a form of the Add (op code 8), SBO (op code 9), LOR (op code C), XOR (op code D), or AND (op code E) command. The contents of the file register designated by the f field are moved to the register designated by the r field, except for the SBT (op code 9) form of the MOV command. The Link Register is unaffected. For the version of the MOV command that utilized Op Code C, the Link Control Bit (bit 7) can be used to inhibit the updating of the zero condition code flag, except when the flag is already set. This allows for propagation of a zero test over multiple byte operations, the first of which would have this bit off.

1.7.4.2.11 Enter 255 Into Register (255). The Enter 255 Into Register command has the following format:



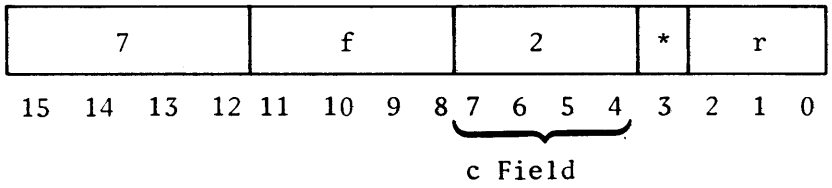
This command is a form of the Logical OR (LOR) command. An all one (decimal 255) byte is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field.

1.7.4.2.12 Complement File (CF). The Complement File command has the following format:



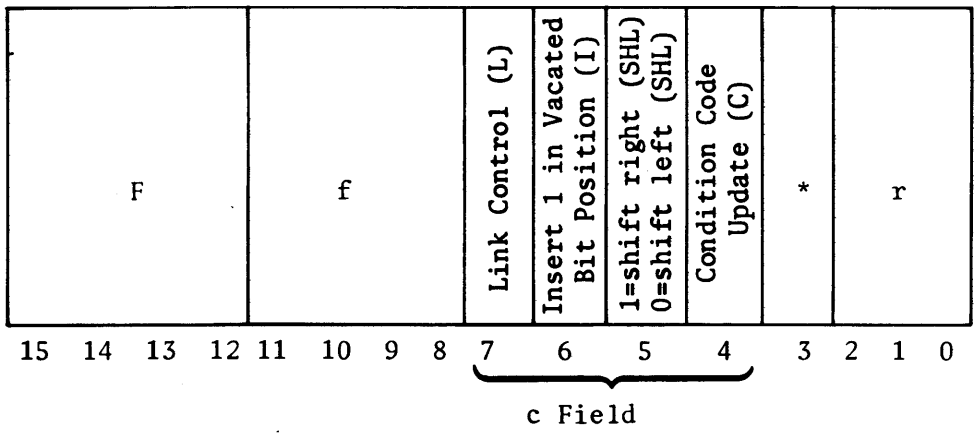
The Complement File (CF) command is a form of the XOR command. The complemented contents of the file register designated by the f field are placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field. The Link Register is unaffected.

1.7.4.2.13 Shift File Right Four (SRF). The Shift File Right Four command has the following format:



In the Shift File Right Four (SRF) execution, the four high-order bits of the file register designated by the f field are placed in the four low-order bits of the register designated by the r field, and (if * is 0) in the file register designated by the f field. Ones are entered into the four high-order bits of the selected register and/or file. The Link Register is unaffected.

1.7.4.2.14 Shift Left or Right (SFL, SLI, SFR, SRI). The Shift Left or Right command has the following format:



In a shift command execution, the contents of the file register designated by the f field are shifted 1-bit position to the right or left, as determined by bit 5 of the c field. The state of the out-shifted bit is entered into the selected link register. A 1, 0, or the previous state of the selected link register is entered into the vacated bit position in the shifted data, as determined by c field bits 7 and 6. The shifted data is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field. There are four basic versions of the shift command, which are defined by c field bits 6 and 5 as follows:

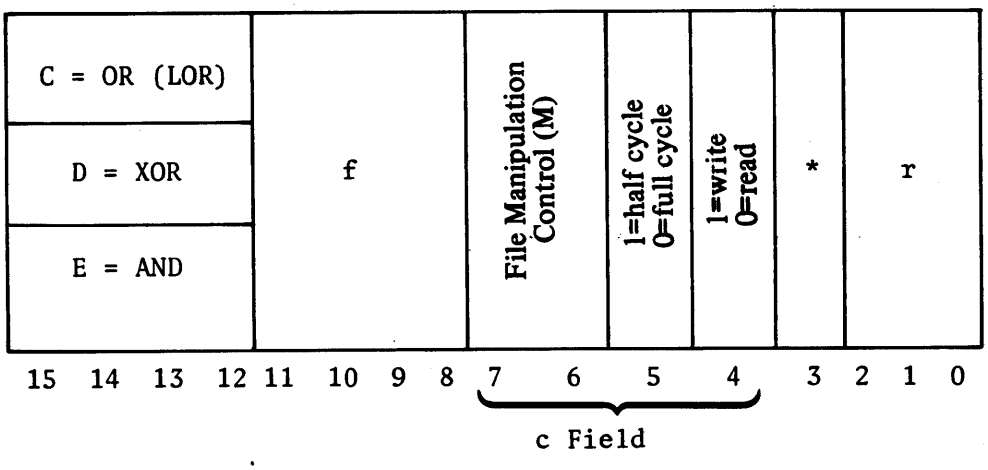
Bit 6	Bit 5	Command	Mnemonic
0	0	Shift File Left	SFL
1	0	Shift File Left and Insert	SLI
0	1	Shift File Right	SFR
1	1	Shift File Right and Insert	SRI

The definitions of c field bits 7 and 4 are as follows:

L (Bit 7) - Link Control: When bit 7 is a 1 the previous state of the selected link register is entered into the vacated bit position in the shifted data (if bit 6 is a 0). Also, updating of the zero condition code flag is inhibited unless the flag is already set.

C (Bit 4) - Condition Code Flag Update: When bit 4 is a 1, the condition code flags are updated according to the value in the shifted data.

1.7.4.3 LOGICAL OPERATIONS. The following paragraphs describe the command formats for logical operations.



The three logical operations are Logical OR (LOR), Exclusive OR (XOR), and Logical AND (AND). The logical operations are performed on the contents of the file register designated by the f field, and the selected operand on the B bus. The result of the logical operation (on the A bus) is placed in the register designated by the r field, and (if * is 0) in the file register designated by the f field. The link is unaffected by logical operations. The definitions of the c field bits for logical operations are as follows:

L (Bit 7) - Link Control: When bit 7 is a 1, the updating of the zero condition code flag is inhibited unless the flag is already set.

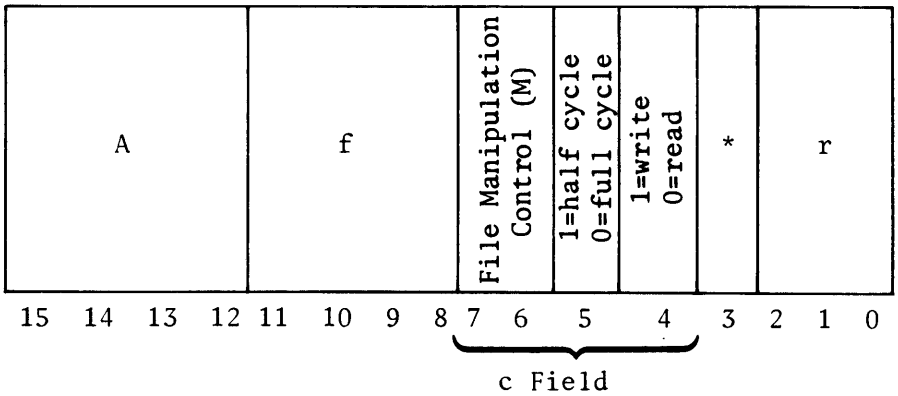
F and T (Bits 6 and 5) - Operand Select Control: Bits 6 and 5 select the operand on the B bus as follows:

Bit 6	Bit 5	Selected Operand
0	0	All zeros
0	1	If IC register bit 3 is a 1, select uncomplemented bits from input data bus. If IC register bit 3 is a 0, select uncomplemented bits from T Register.
1	0	If IC register bit 3 is a 1, select complemented bits from input data bus. If IC register bit 3 is a 0, select complemented bits from T Register.
1	1	All ones

c (Bit 4) - Condition Code Update: When bit 4 is a 1, the condition code flags are updated according to the result of the logical operation on the A bus.

1.7.4.4 CORE MEMORY OPERATIONS. The following paragraphs describe command formats for core memory operations.

CORE MEMORY READ OR WRITE (RMF, RMH, WMF, WMH)



Bits 5 and 4 of the c field define four basic types of core memory operations.

Bit 5	Bit 4	Command	Mnemonic
0	0	Read Memory, Full Cycle - A full-cycle memory read operation is performed on the location specified by the M and N Registers. The byte written into core memory is contained in the T Register.	RMF
1	0	Read Memory, Half Cycle - A half-cycle memory read operation is performed on the location specified by the M and N Registers. The byte written into core memory is contained in the T Register.	RMH
0	1	Write Memory, Full Cycle - A full-cycle memory write operation is performed on the location specified by the M and N Registers. The byte read from core memory is placed in the T Register.	WMF
1	1	Write Memory, Half Cycle - A half-cycle memory write operation is performed on the location specified by the M and N Registers. The byte read from core memory is placed in the T Register.	WMH

The m bits (bits 7 and 6 of c field) specify the operation performed on the file register designated by the f field.

Bit 7	Bit 6	Operation Performed
0	0	The unaltered contents of the selected File Register are placed in the register designated by the r field.
0	1	The contents of the selected file register are decremented by one. When the r field is selecting the M or N Register, the state of the memory link register is added to the decremented file contents. The result is placed in the register designated by the r field, and (if * is 0) in the File Register designated by the f field.
1	0	The state of the selected Link Register is added to the contents of the selected File Register. The result is placed in the register designated by the r field, and (if * is 0) in the File Register designated by the f field.

Bit 7	Bit 6	Operation Performed
1	1	The contents of the selected file register are incremented by one. The result is placed in the register designated by the r field, and (if * is 0) in the File Register designated by the f field.

SECTION 2

CPU THEORY OF OPERATION

2.1 INTRODUCTION

This section contains overall and detailed descriptions of the elements that make up the Central Processing Unit (CPU). Also included are data flow and timing descriptions for the CPU. Detailed descriptions of the core memory, control memory, power supply, and (if applicable) I/O device controllers, DMA channel, and L Save Stack Option are contained in the Supplements section of this manual.

Note

- In the discussions that follow, the relatively high voltage state corresponds to a logical 1 (true) level, and the relatively low voltage state corresponds to a logical 0 (false) level. A slash symbol (/) following a signal mnemonic indicates negation.

- For ease of usage, logic diagrams associated with this section are located in a separate volume. Block diagrams associated with this section are located at the end of the text.

2.2 SYSTEM BLOCK DIAGRAM DESCRIPTION (Figure 2-1)

A basic system consists of a Central Processing Unit (CPU), a control memory, and a variable-sized core memory. Input/output device options include a Teletype (TTY), a direct memory access (DMA) channel, and a variety of I/O devices that can be interfaced with a byte I/O bus. The DMA channel permits external devices to communicate directly with core memory at data transfer rates up to 1 megabyte-per-second. A Teletype may be connected to a special firmware-controlled serial TTY channel (refer to Supplements section of this manual), which permits two-way communication with the CPU. Alternatively, the Teletype can be supplied from a TTY controller (refer to Supplements section of this manual) that replaces the serial TTY interface, and instead interfaces the Teletype with the byte I/O bus. Input and output data transfers for the I/O devices are executed under control of the CPU.

The CPU contains all elements required to execute the 69 microcommands in the CPU repertoire. The microcommands received from control memory generate the control and timing signals required to perform all control operations and data manipulations in the machine. Macro-instructions fetched from a program stored in core memory can be used to modify the microcommands, and to specify firmware subroutines stored in control memory. A firmware subroutine in control memory consists of an organized set of microcommands that accomplish the sequence of actions involved in performing a subroutine.

The flexibility offered by the firmware stored in the control memory permits the CPU to be applied directly to problem solutions by full programming at the micro level; alternatively, the CPU may be applied by emulating the operation of a macro-level computer, in which software instructions are stored in core memory. In the latter case, each software macro-instruction specifies the starting address for the required firmware subroutine or routine stored in control memory.

The storage locations in control memory are accessed by a 12-bit control memory address from the CPU, permitting any one of up to 2048 16-bit storage locations to be accessed. The 16-bit word from the accessed control memory location is furnished to an R Register in the CPU. The microcommand in the R Register is decoded and executed by the CPU.

The core memory storage locations are accessed by a 16-bit core memory address received from either the CPU or the DMA channel, permitting any one of up to 65,536 8-bit locations to be accessed. The accessed location is made available for read and write operations by either the CPU or the DMA channel.

2.3 CPU BLOCK DIAGRAM DESCRIPTIONS (Figure 2-2)

The major elements of the CPU are physically contained on two printed circuit boards installed in the CPU chassis. In addition, an interface board and an optional front panel board are included. The front panel board is considered to be part of the basic CPU described herein. Either of two types of front panel board may be included. The System Front Panel Board is assumed to be part of the basic CPU for purposes of the descriptions that follow. The alternate Basic Front Panel Board does not include the data switches, display indicators, and associated logic that are contained on the System Front Panel Board.

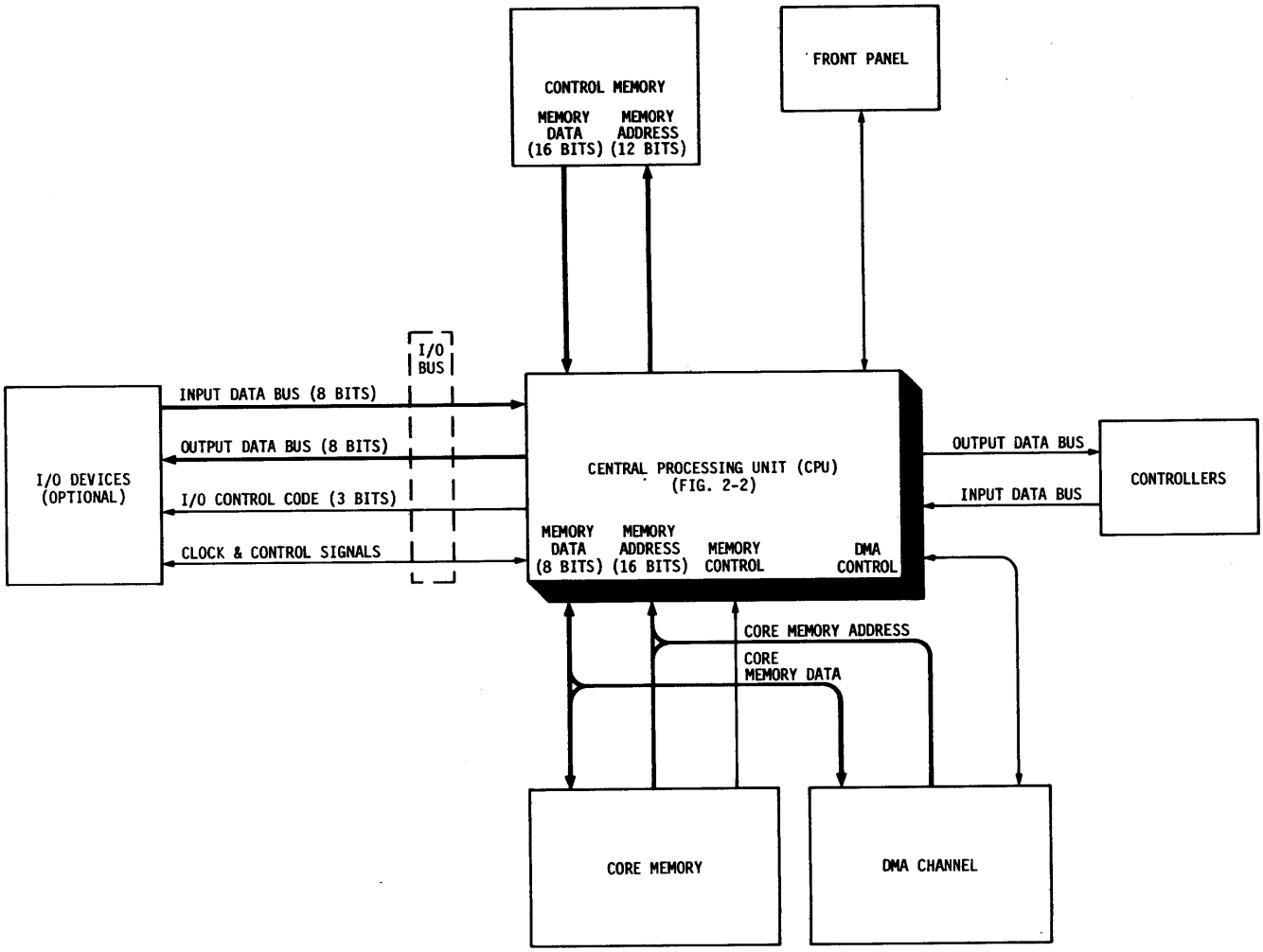


Figure 2-1. CPU Block Diagram

In the functional and detailed descriptions that follow, the abbreviations for the four boards that define the basic CPU are as follows:

Board	Abbreviation	Description
Control Board	C	Includes microcommand decoders, real time clock, power fail circuit, CPU clock circuits, core memory control circuits, and external interrupt control circuit.
Data Board	D	Includes all data and address registers, main data bus multiplexers, and arithmetic/logic circuits.

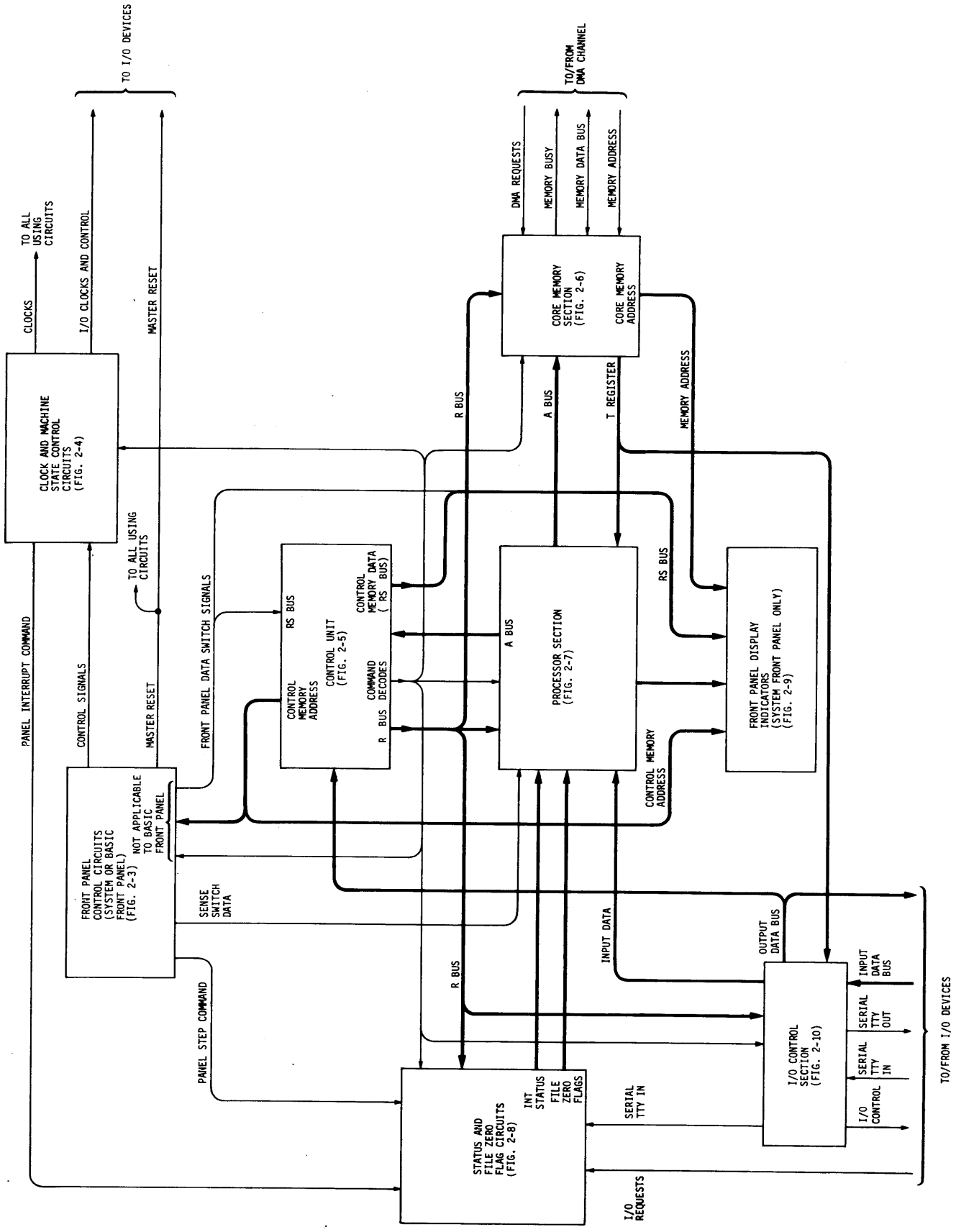


Figure 2-2. CPU Simplified Block Diagram

Board	Abbreviation	Description
Interface Board	IF	Provides interconnections for signals to and from front panel board, and includes serial or parallel teletype interface circuit. (Refer to Supplements section of this manual.)
System Front Panel Board	FP	Includes all front panel switches and indicators, and associated logic circuits.

As shown in Figure 2-2, the CPU is divided into eight major functional sections. Each block on each detailed block diagram (Figures 2-3 through 2-10) identifies the associated logic diagram figure number(s).

During normal operation, the Front Panel Control Circuits are inactive, and perform no function. When enabled, the Front Panel Control Circuits permit the operator to control CPU operation.

All internal clocks and machine state controls (run, halt, logical pause, and panel interrupt) for the CPU are generated by the Machine State Control and Clock Circuits.

The microcommands to be executed by the Processor Section are fetched from control memory, stored, and decoded by the Control Unit. Accessing of core memory for read and write operations by the DMA or the CPU, and distribution of data between core memory and the Processor Section, is accomplished by the Core Memory Section. All arithmetical and logical operations, as well as destination routing for data and addresses, are performed by the Processor Section. The Status and File Zero Flag Circuits process all internal and external status signals and interrupts for the Processor Section.

Data and memory addresses from the Processor Section, the Control Unit, and the Core Memory Section may be selected for display by the Front Panel Indicators.

The I/O Control Section controls the transfer of data between the I/O devices and the CPU.

Detailed functional descriptions of each functional section are provided in the following subparagraphs.

2.3.1 FRONT PANEL CONTROL CIRCUITS (Figure 2-3)

All circuits in this functional section are contained on the front panel (FP) board. As mentioned previously, either of two types of front panel boards may be installed in the system. All of the elements shown on Figure 2-3 are included on the System Front Panel Board. Only the control switches and control display are included on the Basic Front Panel Board.

2.3.1.1 CONTROL SWITCHES AND CONTROL DISPLAY. With the exception of the PANEL switch and indicator, which are not included on the Basic Front Panel Board, the elements in this group are contained on both the System and Basic Front Panel Boards. The control switches include the four sense switches, the key-lock power switch, the RUN, STEP, INT, CLOCK, and RESET machine state control switches, and the PANEL mode switch. The control display circuits include the indicators and associated logic for the RUN, HALT, LOCK, and PANEL indicators.

The machine state control switches, and the PANEL switch, are enabled when the key-lock switch is set to the ON position, permitting the operator to control CPU operation from the front panel. The control display indicators display the front panel status (PANEL and LOCK) and the machine state (RUN and HALT).

The states of the four sense switches are made available to the A bus multiplexer in the Processor Section. The sense switch states are examined by the Processor Section in response to an Enter Sense Switches (ESS) microcommand.

2.3.1.2 UPPER AND LOWER DATA SWITCHES AND ADST/SCAN DISPLAY. The System Front Panel Board includes 16 data switches, which are enabled when the key-lock switch is in the ON position and the PANEL switch is on. The lower data switches are also enabled when an Enter Control Switches (ECS) microcommand is being executed by the CPU, producing an enable panel (AENP) command decode. When enabled by the PANEL and key-lock switches, the upper and lower data switches replace the control memory as the source of microcommands for the CPU. When enabled by the AENP command decode, the states of the lower data switches are AND'ed with the low-order bits of the microcommand from control memory, permitting microcommands from control memory to be modified at the front panel.

When the key-lock switch is set to ON and the PANEL switch is off, upper data switches 14 and 15 have special functions. Switch 14 is used to enable the SCAN mode, in which the SCAN indicator lights, and the CPU steps sequentially through the control memory addresses without executing the resulting microcommands. This mode permits the contents of control memory to be displayed by the Front Panel Display Indicators. Switch 15 is used to enable the ADDRESS STOP mode, which employs the L address comparator and address stop logic described in the following subparagraphs. In the ADDRESS STOP mode, the ADST indicator lights and the operator selects a control memory stopping address with the upper and lower data switches. The L Register in the Control Unit contains the address of the location being accessed in control memory. When the L Register address matches the address set into data switches 11 through 0, a machine HALT is initiated. If the microcommand in the stopping address does not involve a skip, jump, or return jump, the CPU stops at the stopping address plus one. If the microcommand at the stopping address contains a jump or return jump command, the CPU stops at the jump or return jump (L Save) address. When a skip-type microcommand is at the stopping address and the skip is taken during command execution, the CPU stops at the stopping address plus two. If the selected stopping address is the next address beyond a location containing a jump or return jump microcommand, no address stop occurs. Also, no address stop occurs when the selected stopping address is the next address beyond a location containing a skip-type command, and the skip is taken.

2.3.1.3 L ADDRESS COMPARATOR. The L address comparator performs a comparison between the 12-bit address in the L Register of the Control Unit and the number set into data switches 11 through 0. When a match exists between the L Register contents and the number set into the data switches, the L address comparator produces LSYC (L address sync), which is made available for test point monitoring, and is also supplied to the address stop logic.

2.3.1.4 ADDRESS STOP LOGIC. The address stop logic is only activated in the ADDRESS STOP mode. When the L address comparator produces LSYC, indicating that the stopping address has been reached, the address stop logic activates CLKF/, producing a CPU HALT. The control memory address at which the halt occurs is dependent upon the microcommand contained in the stopping address location, as described in Paragraph 2.3.1.2.

2.3.2 MACHINE STATE CONTROL AND CLOCK CIRCUITS (Figure 2-4)

All circuits in this functional section are contained on the control (C) board. Included are the run and inhibit control, the clock generator, and the clock gates, which operate together to control machine states, and the generation of all clocks used in the CPU.

2.3.2.1 RUN AND INHIBIT CONTROL. This element serves as the machine state control in the CPU. During the execution of jump and return microcommands, or when execution of a microcommand requires a control memory address skip, the run and inhibit control initiates a logical pause lasting one microstep. During the logical pause, the generation of certain clocks produced by the clock gates are inhibited. When a HALT microcommand is being executed (when an address stop is initiated) by the Front Panel Control Circuits, or when the front panel CLOCK switch is enabled and activated while the CPU is running, the run and inhibit control initiates a CPU HALT. When a CPU HALT occurs, the CPU may be returned to the RUN state by pressing the RUN, STEP, CLOCK, or INT switch on the front panel. The CPU is also restored to the RUN state when power-up occurs following a power shutdown or power failure.

In addition to performing the machine state control functions, the run and inhibit control also controls the generation of a panel interrupt, which is initiated when the front panel INT switch is enabled and activated.

2.3.2.2 CLOCK GENERATOR. The clock generator contains a 20-MHz, crystal oscillator that serves as the time base for the CPU. The 20-MHz signal is divided by four, and various phases of the resulting 5-MHz clock are produced for the clock gates (SCK6), the I/O devices (CPH1 and CPH2/), the Front Panel Control Circuits (CPH2/), the Processor Section (FLCK/), the Core Memory Section (MNCK/), and the Control Unit (FLCK/).

2.3.2.3 CLOCK GATES. The clock gates derive various internal clocks from the 33-nanosecond wide, 5-MHz clock (SCK6) received from the clock generator. In some cases, clocks are produced when only specific microcommands are being executed; in other cases, clocks are inhibited when a HALT or logical pause is initiated by the run and inhibit control, or by the Core Memory Section during a memory read or write cycle. The complete clock distribution is shown in Figure 2-4.

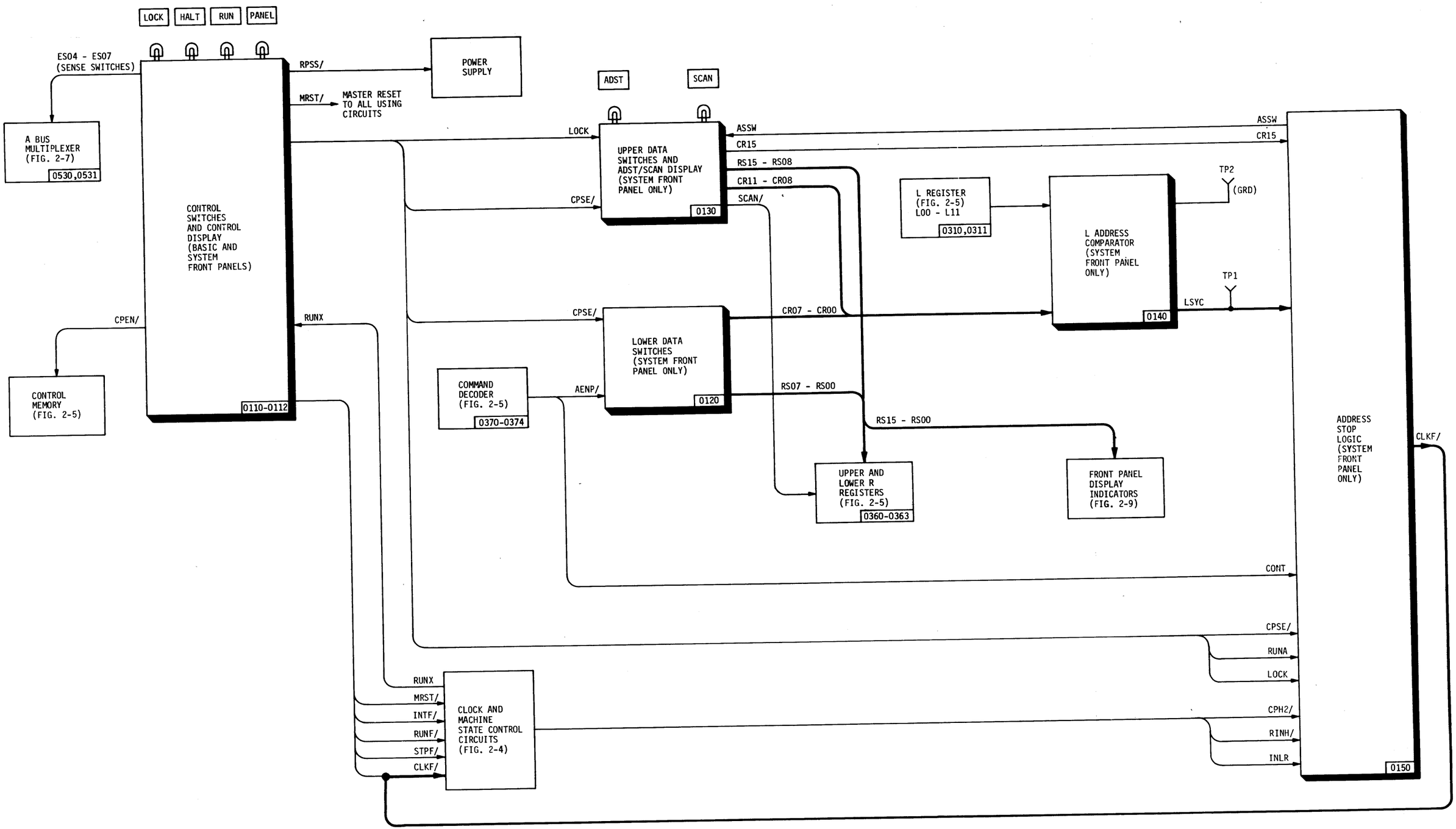


Figure 2-3. Front Panel Control Circuits

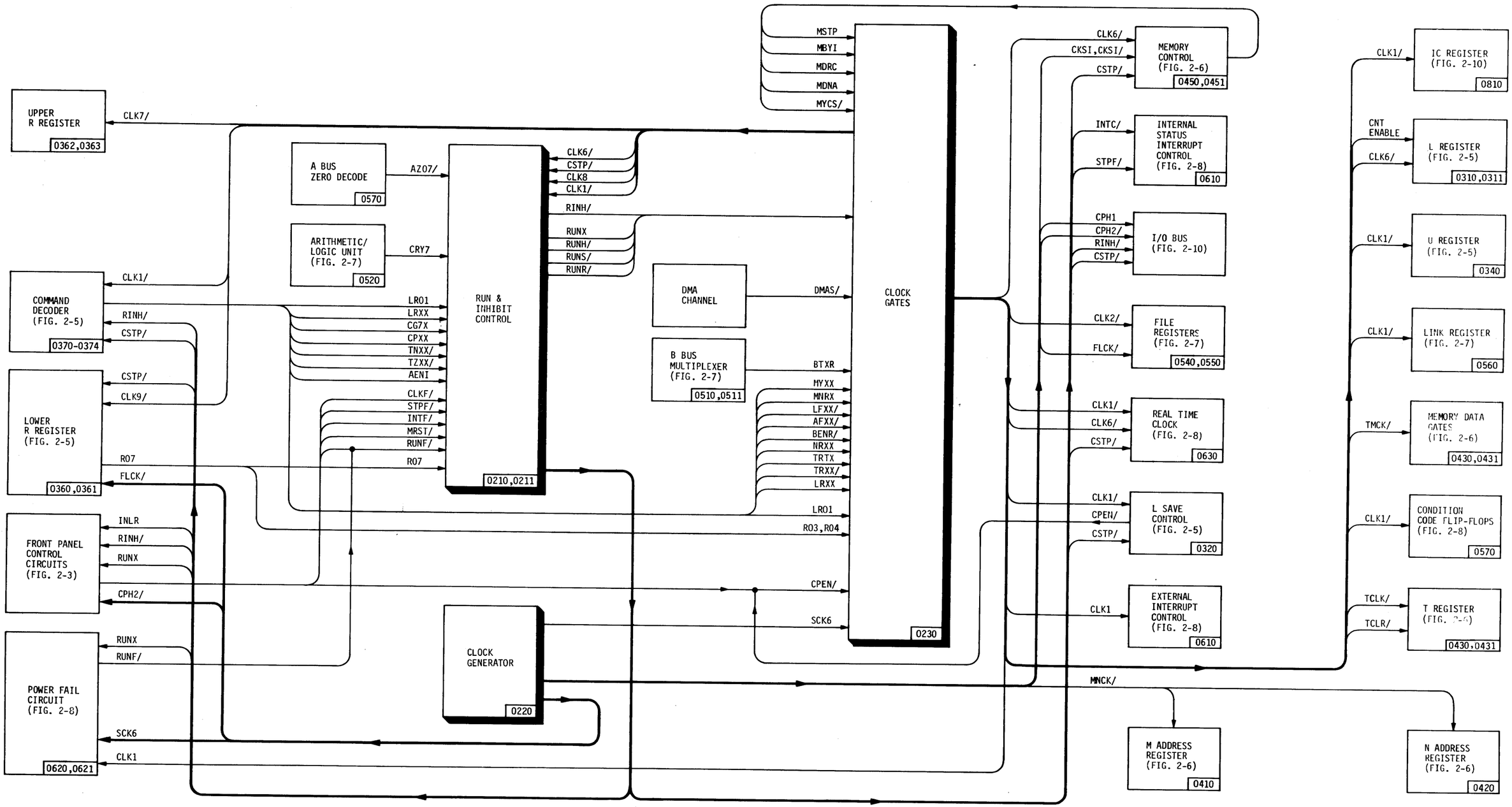


Figure 2-4. Clock and Machine State Control Circuits

2.3.3 CONTROL UNIT (Figure 2-5)

The Control Unit includes circuits contained on the control (C) and data (D) boards, and also includes the control memory. The purpose of the Control Unit is to access each microcommand in control memory, and to store and decode the microcommand. The resulting command decode signals are used to execute the microcommand. Major elements of the Control Unit are discussed in the following subparagraphs.

2.3.3.1 CONTROL MEMORY. The control memory is normally implemented with semiconductor read only memory (ROM) devices. Alternatively, a read-write memory providing an alterable control memory (ACM) can be utilized. The standard control memory contains up to 4096 16-bit words (microcommands). Accessing of control memory locations is accomplished by the L Register which supplies a 12-bit address code to the control memory. The 16-bit word from the accessed location is placed on the RS bus to the upper and lower R Registers.

2.3.3.2 L REGISTER. The L Register provides the means for accessing microcommands stored in the control memory. During the execution of jump and return jump type microcommands, the L Register is parallel loaded with a jump address received from the A bus and/or the upper R Register. Starting from the jump address, the L Register then steps sequentially through control memory addresses to access the microcommands until another jump or return command is executed.

2.3.3.3 L SAVE REGISTER. When executing an extended jump or return jump microcommand, a later return to the control memory address immediately following the address containing the extended jump or return jump microcommand may be desired. In this case the L Register address must be saved. The address to be saved is transferred from the L Register to the L Save Register. Subsequently, when a return microcommand is being executed, the address contained in the L Save Register is returned to the L Register, permitting the CPU to resume microcommand execution at the point where the jump or return occurred.

2.3.3.4 UPPER AND LOWER R REGISTERS. The upper and lower R Registers accept and store the data on the RS bus. The data on the RS bus can be a 16-bit microcommand received from either the control memory or the front panel data switches, or the 12-bit saved address from the L Save Register during the execution of a return command. Under certain circumstances, the upper and lower R Registers modify the data

on the RS bus before storage. The high-order modification bits (bits 8 through 15) are supplied from the U Register. The low-order modification bits (bits 0 through 7) are received from the output data (OD) register in the I/O Control Section. The data stored in the upper and lower R Registers are distributed over the R bus, as shown in Figure 2-5.

2.3.3.5 U REGISTER. The U Register is used to modify the high-order bits of the microcommand on the RS bus. The modification bits stored in the U Register are received from the A bus. When enabled, the contents of the U Register are gated to the upper R Register, where they are OR'ed with the high-order bits on the RS bus.

2.3.3.6 COMMAND DECODER. The command decoder receives the 16-bit microcommands presented on the R bus, and decodes the words to produce the command decode signals required to execute each microcommand. The command decodes are distributed throughout the CPU, as shown in Figure 2-5.

2.3.4 CORE MEMORY SECTION (Figure 2-6)

The Core Memory Section includes circuits on both the control (C) and data (D) boards, and also includes the core memory and the optional DMA channel. The purpose of the Core Memory Section is to access locations in core memory for storage and retrieval, either by the CPU or by the DMA.

When the DMA is being serviced for a memory operation, the DMA supplies the address to the core memory, and data to be stored or retrieved from core memory is supplied over the MD bus connections to the DMA channel. When the CPU is being serviced for a read or write operation, the M and N Registers supply the core memory address. Data retrieved from core memory is supplied via the memory data gates to the T Register. Data to be written into memory is supplied to the core memory from the MD Register in the Core Memory Section. The memory control supplies the timing signals for the memory read or write operation when either the CPU or the DMA is being serviced.

Data read from memory to the CPU is routed from the Core Memory Section to either the B bus multiplexer in the Processor Section, or to the output data (OD) register in the I/O Control Section. Data to be written into memory by the CPU is routed from the A bus in the Processor Section to the T Register in the Core Memory Section.

Descriptions of the main elements of the Core Memory Section are provided in the following subparagraphs.

2.3.4.1 CORE MEMORY. The core memory contains up to 65,536 8-bit storage locations, which are accessed by 16-bit address codes received from the M and N Registers, or from the DMA channel. Data to be written into the accessed locations is presented on the bi-directional memory data (MD) bus from either the Core Memory Section memory data (MD) register or the DMA channel. Data read from the accessed location is presented on the MD bus to the memory data gates in the processor and the DMA channel. Memory cycle timing is controlled by the memory control.

2.3.4.2 MEMORY CONTROL. The timing of memory read and write cycles is controlled by the memory control. The type of cycle initiated is determined by the microcommand being executed, or by the optional DMA channel when the DMA is being serviced. The four types of memory cycles are as follows:

a. Full-Cycle Read. This cycle is used for a non-destructive readout of the accessed memory location. The contents of the accessed location are read out and supplied to the DMA, or to the memory data gates, which supply the T Register with the memory read data. Then, during the write portion of the cycle, the original contents of the accessed location are written back into memory.

b. Half-Cycle Read. This cycle is used for a destructive readout of the accessed memory location. A half-cycle read operation cannot be initiated by the DMA. The contents of the accessed location are read out and supplied via the memory data gates to the T Register. No write action is performed, and the accessed memory location is left in an all 1 state.

c. Full-Cycle Write. This cycle is used to replace the current contents of the accessed location with new data. First, a memory read is performed, leaving the accessed location in an all 1 state. Then, the new data to be written (supplied to the MD Register from the T Register, or supplied from the DMA) is supplied over the MD bus and written into the accessed location.

d. Half-Cycle Write. This cycle can be used to write data into memory when the accessed location has been left in an all 1 state by a previously performed half-cycle read operation. A half-cycle write operation cannot be initiated by the DMA. The data to be written (supplied to the MD Register from the T Register) is supplied over the MD bus and written into the accessed location.

2.3.4.3 M AND N ADDRESS REGISTERS. The M and N Address Registers supply the 16-bit memory address to the core memory when the CPU is being serviced. The M Register contains the eight high-order bits of the address, and the N Register contains the

eight low-order bits. The M and N Registers are loaded with each address from the A bus in the Processor Section.

2.3.4.4 MEMORY DATA GATES. During a memory read operation, the memory data gates are used to gate the data from the MD bus to the T Register. The memory data gates are inhibited when the DMA is being serviced.

2.3.4.5 T REGISTER. The T Register serves as a general-purpose temporary storage register, both for the Processor Section and the Core Memory Section. During most arithmetical and logical operations, the operand is loaded into the T Register from the A bus. The operand in the T Register is then supplied to the B bus multiplexer. During memory read operations, the memory read data on the MD bus is supplied via the memory data gates to the T Register. During memory write operations, the data to be written into memory is supplied from the A bus to the T Register, and is subsequently loaded into the MD Register for entry into core memory.

2.3.4.6 MEMORY DATA (MD) REGISTER. The MD Register is loaded from the T Register with the data to be written into core memory during a memory write operation. The output from the MD Register is supplied via the MD bus to the core memory. When the DMA is being serviced, the MD Register is inhibited from supplying data to the MD bus.

2.3.5 PROCESSOR SECTION (Figure 2-7)

The circuits in this functional section are contained on the control (C), data (D), and front panel (FP) boards. The purpose of the Processor Section is to route data throughout the CPU, and to perform all arithmetic and logical operations on the data. Data on the Processor Section A bus can be entered (directly or indirectly) into any of the data and address registers in the CPU, including the registers in the Control Unit and the Core Memory Section.

Data is entered into the Processor Section via either the A or B bus multiplexers, or the File Register output buffer. Data that can be entered into the Processor Section via the B bus multiplexer includes R bus data from the lower R Register in the Control Unit, data from I/O devices (supplied from the input data bus via the input data bus gate in the I/O Control Section), and data from the T Register in the Core Memory Section. Data that can be entered into the Processor Section via the A

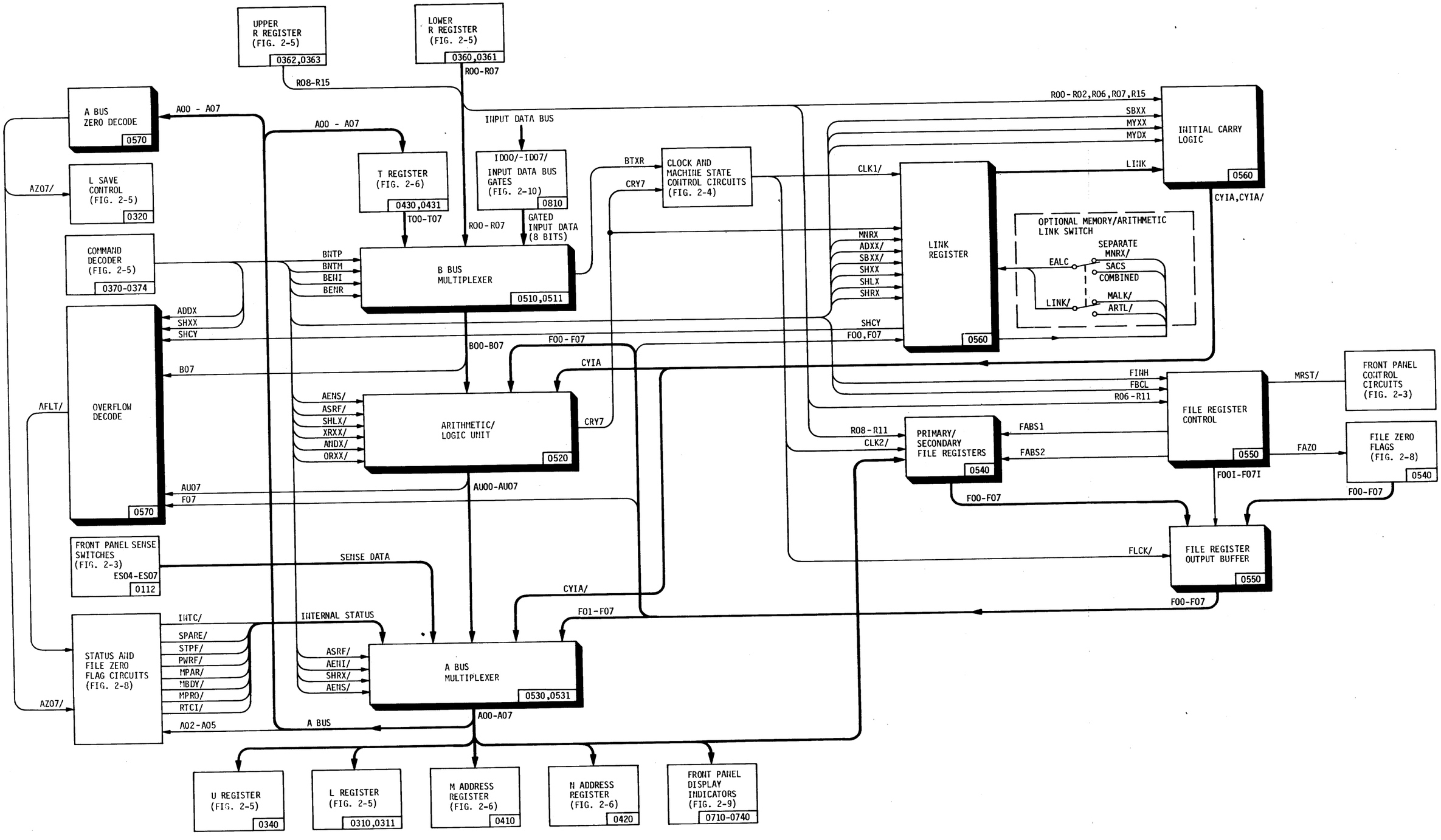


Figure 2-7. Processor Section

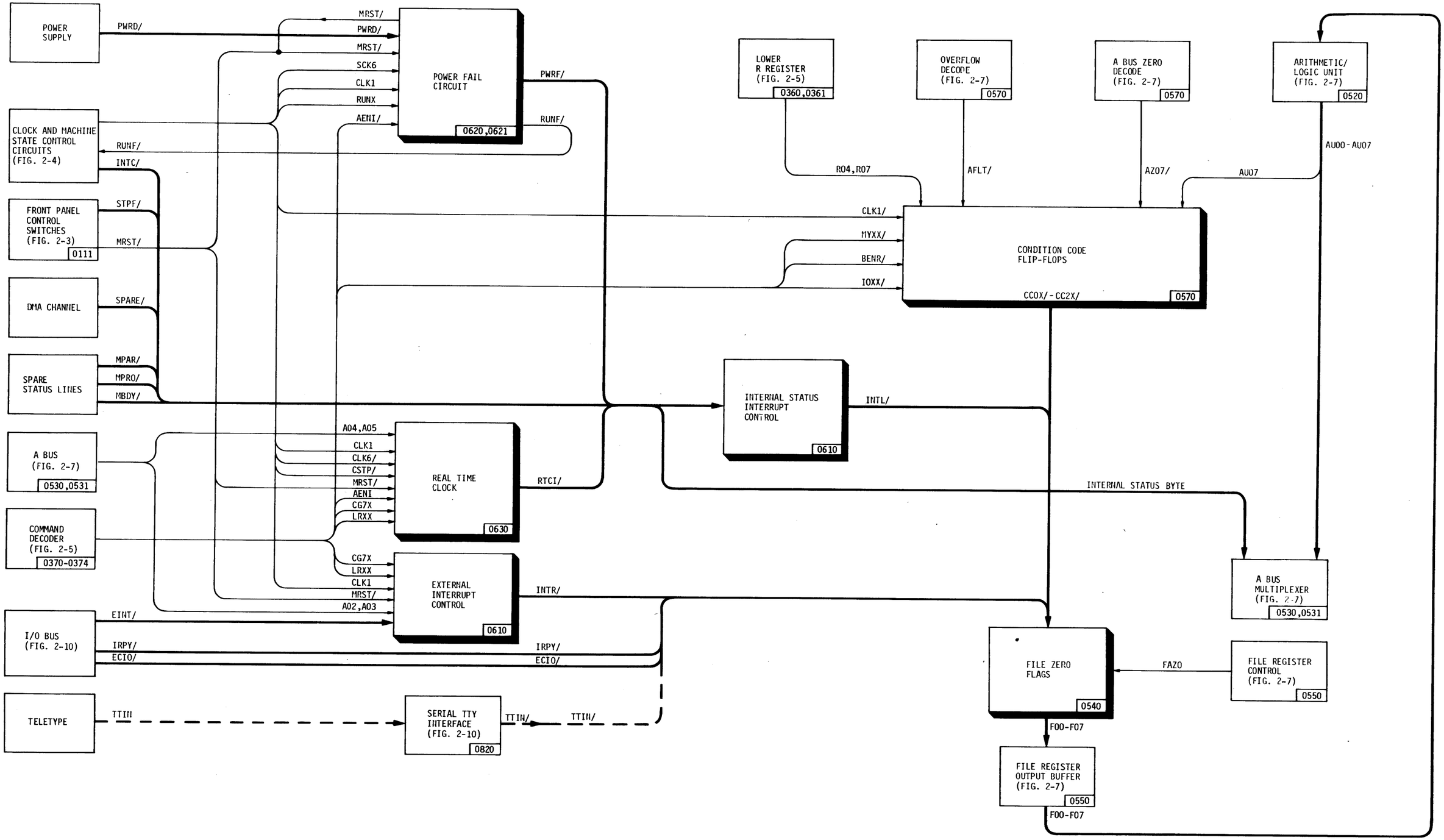


Figure 2-8. Status and File Zero Flag Circuits

bus multiplexer includes internal status data from the Status and File Zero Flag Circuits, and sense switch data from the front panel. File Zero Flag data from the Status and File Zero Flag Circuits is entered into the Processor Section via the File Register output buffer.

The following subparagraphs contain descriptions of the major elements of the Processor Section.

2.3.5.1 B BUS MULTIPLEXER. The B bus multiplexer contains selector gates that select the source of data supplied to the B input of the arithmetic/logic unit (ALU). Data selection depends upon the microcommand being executed, as established by the decode signals supplied from the command decoder in the Control Unit.

2.3.5.2 ARITHMETIC/LOGIC CIRCUITS. This circuit group includes the ALU, the Link Register, the initial carry logic, the A bus zero decode, and the overflow decode. These elements operate together to perform arithmetical and logical operations on the data from the B bus multiplexer and the File Registers. The arithmetical or logical operation performed depends upon the microcommand being executed, as established by decode signals supplied from the command decoder in the Control Unit. The following arithmetical and logical operations can be performed:

a. Transfer B Bus Data or File Register Data to A Bus Multiplexer. In this mode, the ALU serves as a port for transferring the unmodified data from the B bus or the File Register to the A bus multiplexer.

b. Add File and B Bus Data. In this mode, the data from the File Registers are summed with the selected data on the B bus.

c. Subtract B Bus Data From File. In this mode, the B bus multiplexer gates the complemented subtrahend to the ALU, where it is added to data from the File Registers. The result is a one's or two's complement subtraction (depending on the state of the initial carry bit). The carry-out of the high-order bit of the ALU is placed in the link register.

d. Shift File Left. In this mode, data from the File Registers are shifted one bit position to the left towards the high-order bit position. The high-order bit that is shifted out is placed in the Link Register.

e. Logical AND of B Bus and File Data. In this mode, a logical AND is performed between data on the B bus and data from the File Registers.

f. Logical OR of B Bus and File Data. In this mode, a logical OR is performed between data on the B bus and data from the File Registers.

g. Exclusive OR of B Bus and File Data. In this mode, a logical exclusive OR is performed between data on the B bus and data from the File Registers.

The initial carry logic determines the state of the initial carry input supplied to the ALU and the A bus multiplexer during arithmetical and logical operations. The state of the initial carry is determined by the microcommand being executed, and (in some cases) by the state of the Link Register bit.

The result produced by the ALU is gated through the A bus multiplexer to the A bus for distribution to the file or register designated in the microcommand being executed.

The A bus zero decode detects the presence of an all zero condition in the arithmetic/logic result on the A bus. The overflow decode detects the presence of an overflow condition during an arithmetic operation. An overflow condition exists when the carry-out of the high-order bit of the ALU differs from the carry-in to the high-order bit of the ALU.

2.3.5.3 FILE REGISTERS. The Processor Section contains thirty 8-bit File Registers to provide temporary storage of working data. When operate-type microcommands are being executed, the microcommand designates (in the f field) which of the File Registers supplies data (via the file register output buffer) to the ALU and to the A bus multiplexer. The result of the arithmetical or logical operation can then be returned (via the A bus) to the addressed File Register and/or the destination register. The File Registers are grouped into 15 primary registers and 15 secondary registers. Selection of the primary or secondary file register group or bank (designated by primary or secondary file select microcommands) is performed by the file register control.

In addition to file register data, the file register output buffer also supplies the file zero flag byte (from the Status and File Zero Flag Circuits) to the ALU.

2.3.5.4 A BUS MULTIPLEXER. The A bus serves as the main data distribution bus in the CPU. The A bus multiplexer selects which of the various data inputs (see Figure 2-7) is selected for output to the A bus. The A bus data selection depends upon the microcommand being executed, as established by decode signals from the command

decoder in the Control Unit. In addition to data selection, the A bus multiplexer also performs the shift right arithmetic operation on the file register data.

2.3.6 STATUS AND FILE ZERO FLAG CIRCUITS (Figure 2-8)
Circuits in this functional section are contained on the control (C) and data (D) boards. The purposes of the Status and File Zero Flag Circuits are as follows:

a. Detection and processing of internal and external interrupts.

b. Processing of Teletype data received via the serial TTY interface.

c. Formulation of the internal status byte supplied to the A bus when an Enter Internal Status (EIS) microcommand is being executed.

d. Formulation of a file zero flag byte for output (via the file register output buffer) to the arithmetic/logic unit in the Processor Section when the operate-type microcommand being executed addresses file zero.

Major elements of the Status and File Zero Flag Circuits are discussed in the following subparagraphs.

2.3.6.1 POWER-FAIL CIRCUIT. This circuit, in conjunction with the power supply, monitors the ac line for power drop-out or significant voltage transients. Upon detection of such a condition, the power-fail circuit generates the PWRF signal to the internal status interrupt control, and to the A bus multiplexer. In response to the power-fail signal, the internal status interrupt control initiates an interrupt that causes a jump to the power-is-failing subroutine. The purpose of this subroutine is to store in core memory all flags and register contents which are volatile during a power shutdown or a power turn-on. When the subroutine ends (within one millisecond after power-fail detection), the CPU is halted by program command. At this time, the machine is locked out in preparation for a loss of power, and a master reset (MRST/ = 0) occurs.

When normal ac power resumes, the power-fail circuit generates PWRF again, and the resulting interrupt produced by the internal status interrupt control is interpreted as a power-start interrupt. In addition, the CPU is placed in the RUN mode. The power-start interrupt initiates a recovery subroutine, during which all flags and registers are restored to their pre-failure condition, and normal operation is resumed.

2.3.6.2 REAL TIME CLOCK. When enabled under microcommand control, the real time clock counts repetitively through a 1-millisecond count cycle. Each time a count cycle is completed, a real time clock interrupt (RTCI) is supplied to the internal status interrupt control, and to the A bus multiplexer. The RTCI interrupt is terminated when the CPU executes an Enter Internal Status (EIS) microcommand.

2.3.6.3 EXTERNAL INTERRUPT CONTROL. When an external interrupt (EINT) is received from a device connected to the I/O bus, the external interrupt control supplies an interrupt request (INTR) to the file zero flags. The external interrupt terminates when the CPU executes an Interrupt Acknowledge (IAK) microcommand.

2.3.6.4 INTERNAL STATUS INTERRUPT CONTROL. The internal status interrupt control receives status inputs from the power-fail circuit, the real time clock, the front panel control switches, the Machine State Control and Clock Circuits, and four spare lines. When one or more of these status lines is active, an internal interrupt (INTL) is supplied to the file zero flags.

2.3.6.5 CONDITION CODE FLIP-FLOPS. When enabled by an I/O command (op code 7) or by an update condition code bit (bit 4) of an operate-type microcommand with op code 8, 9, B, C, D, E, or F, the condition code flip-flops store the presence of an overflow, negative, or zero condition resulting from an arithmetical or logical operation. The outputs from the three condition code flip-flops are supplied to the file zero flags.

2.3.6.6 FILE ZERO FLAGS. The file zero flags consist of gates that enter the file zero flag byte into the file register output buffer when enabled by the file register control. The 8-bit file zero flag byte is made up of the three condition code flip-flop bits, the INTR and INTL lines from the external and internal interrupt controls, the TTIN (teletype in) line from the serial TTY interface circuit, and the IRPY (I/O reply) and ECIO (concurrent I/O request) lines from the I/O bus. When the L Save Stack Option is included, the TTIN input is replaced by a stack overflow (STOV) input from the L Save Stack Option. The states of the eight flag lines are entered into the file register output buffer when the microcommand being executed contains the file zero address in the f field.

2.3.7 FRONT PANEL DISPLAY INDICATORS (Figure 2-9)

All Front Panel Display Indicator circuits are contained on the System Front Panel (FP) Board. The Front Panel Display Indicators are not included on the Basic Front Panel Board. The purpose of the Front Panel Display Indicators is to display operator-selected data on the front panel. Data selected for display is determined by a display multiplexer, which is controlled by the front panel display select pushbutton switches. Sources of display data include the RS bus, the L Register, the M and N Register, and the A bus.

2.3.8 I/O CONTROL SECTION (Figure 2-10)

Circuits in the I/O Control Section are contained on the data (D) and interface (IF) boards. The elements in this functional section provide the main data interface between the CPU and the I/O devices connected to the byte I/O bus. Also included is the serial teletype (TTY) interface with the Teletype. The following subparagraphs describe each of the major elements of the I/O Control Section.

2.3.8.1 OUTPUT DATA (OD) REGISTER. The OD Register receives data from either the Processor Section or the Core Memory Section via the T Register. The 8-bit byte in the OD Register is supplied over the output data bus to I/O devices, and to the lower R Register (MLC gating) in the Control Unit.

2.3.8.2 I/O CONTROL (IC) REGISTER. When any of the external I/O (EXIO) commands are being executed by the CPU, bits 4 through 6 of the microcommand in the lower R Register contain the I/O control code. These bits are loaded into the IC Register. The IC register outputs (IO1X/ through I03X/) are furnished to the serial TTY interface, and to I/O devices connected to the byte I/O bus. The I/O control code defines the operation to be performed by the selected I/O device.

2.3.8.3 INPUT DATA BUS GATES. These gates select either complemented or uncomplemented data from the input data bus for application to the B bus multiplexer. Complemented data is selected when a subtraction operation is being performed, or when a logical operation (op code C, D, or E) is in progress and bits 6 and 5 of the f field contain code 10.

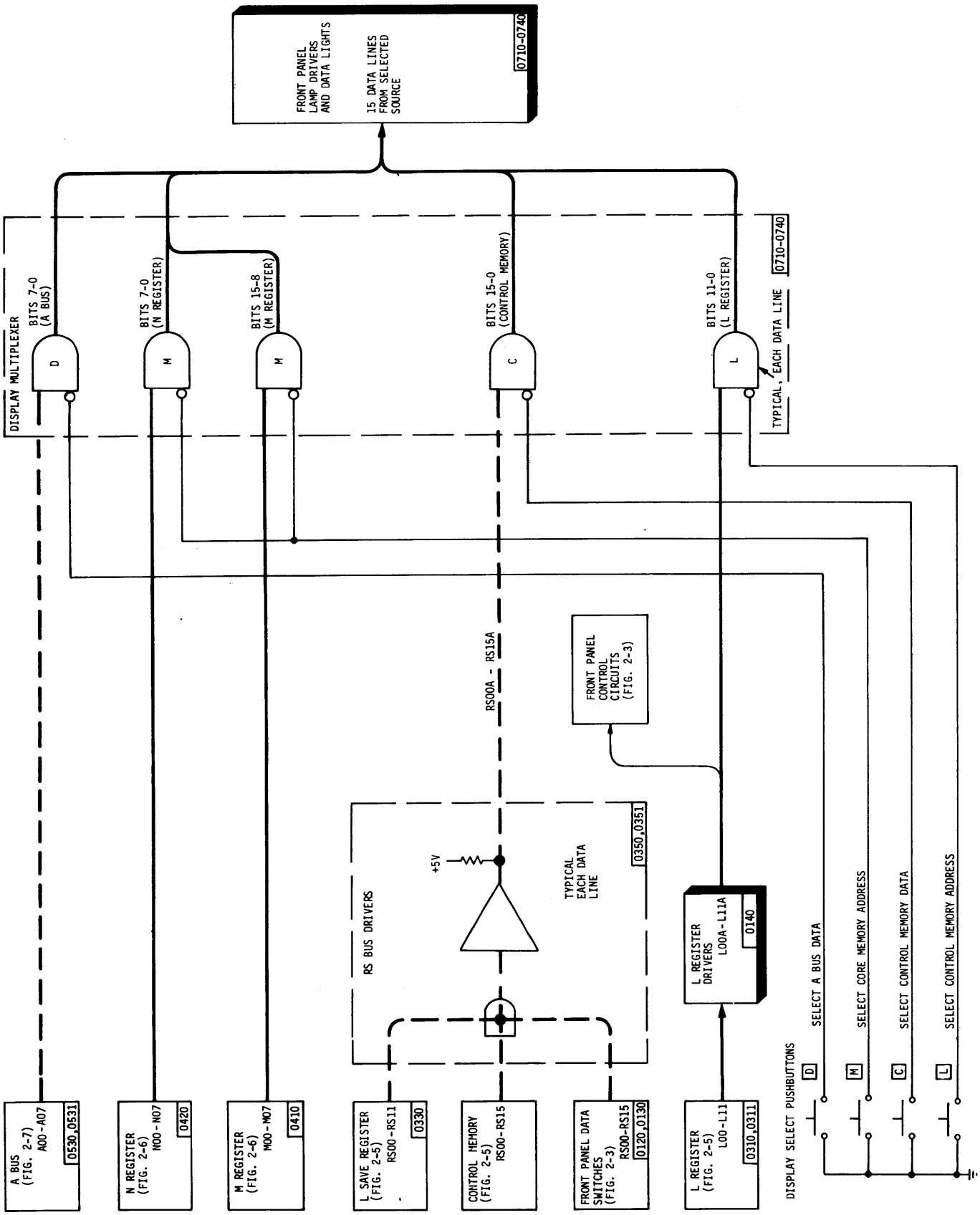


Figure 2-9. Front Panel Display Indicators

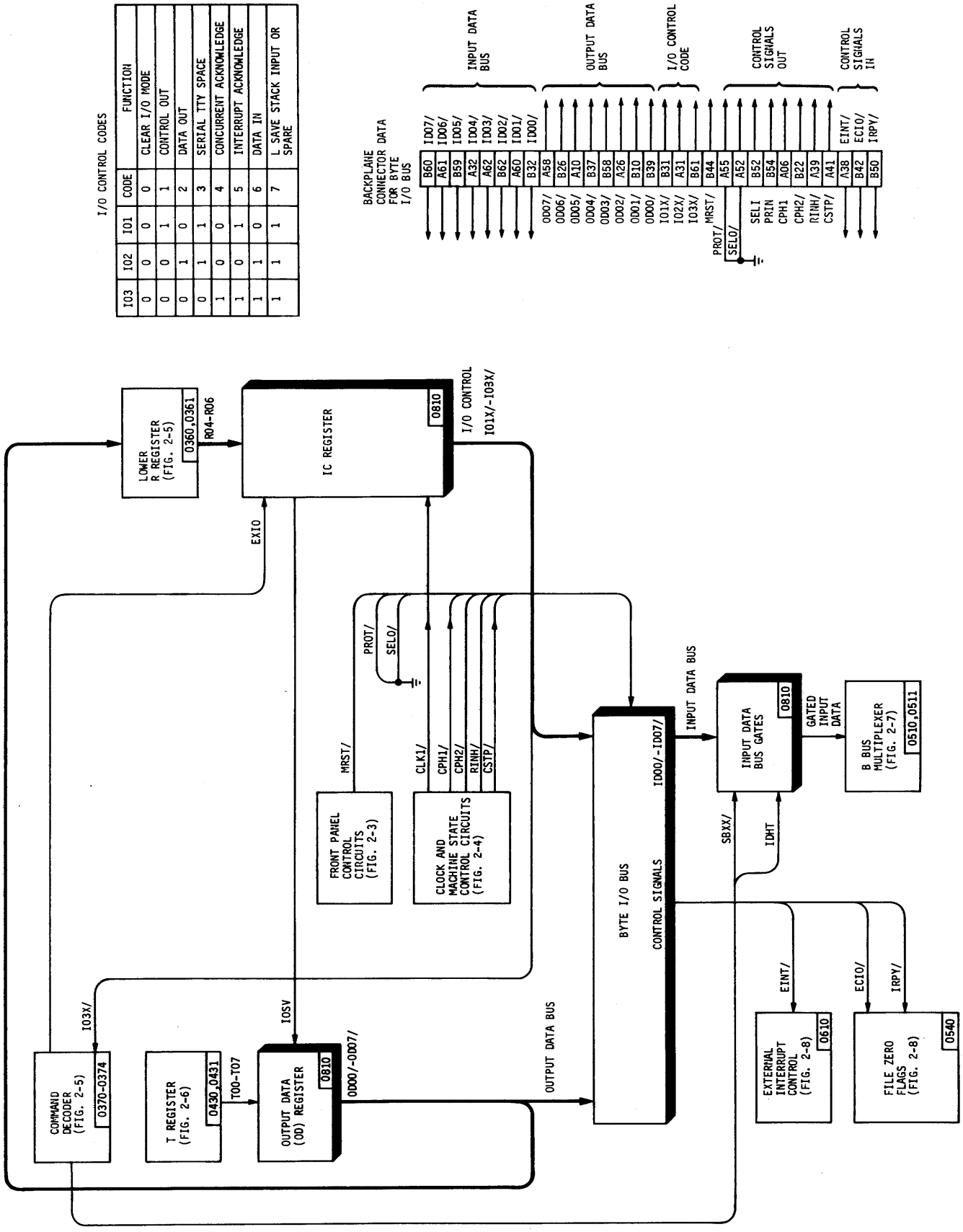


Figure 2-10. I/O Control Section

2.4 DATA FLOW DESCRIPTIONS (Figures 2-3 through 2-12)

The following subparagraphs contain data flow descriptions for each major data flow path in the CPU. Where applicable, pin-outs to the backplane connectors are also described.

2.4.1 CONTROL UNIT DATA FLOW (Figure 2-5)

Figure 2-5 depicts the data flow details for the Control Unit. Figure 2-5 also shows the backplane connector pin-outs for the control memory bus.

Microcommands to be executed are supplied over the RS bus, and are entered into the upper and lower R Registers. During normal operation, the source of the microcommand on the RS bus is the control memory. When the front panel is enabled, the control memory outputs are inhibited, and the source of microcommands on the RS bus is the data switches on the System Front Panel Board. Each microcommand contains 16 bits.

During normal operation, each microcommand is fetched from control memory by the 12-bit address from the L Register. Except when a jump or a return jump microcommand is being executed, the L Register steps sequentially through the control memory addresses. When a jump (jump extended or jump in 1K) microcommand is executed, the jump address (part of the microcommand in the R Register) is transferred into the L Register to produce an address jump.

When a return jump microcommand is executed, the current contents of the 12-bit L Save Register are transferred to the L Register to produce another type of address jump. If L Save is not inhibited by an earlier microcommand, the address immediately following the address that contained the jump or return jump microcommand is transferred from the L Register to the L Save Register and stored for a subsequent return jump. The data path from the L Save Register to the L Register is as follows:

a. The four high-order bits of the address in the L Save Register are supplied over the RS bus (outputs from control memory are inhibited), and are clocked into bit positions 8 through 11 of the upper R Register. At the next clock, these bits are clocked into the four high-order bit positions of the L Register.

b. The eight low-order bits of the address in the L Save Register are supplied over the RS bus (outputs from control memory are inhibited), and are clocked into the lower R Register. These bits are supplied from the lower R Register via the B bus multiplexer, the arithmetic/logic unit, and the A bus multiplexer to the L Register, and are clocked into the eight low-order bit positions of the L Register at the next clock.

A third type of address jump is provided for an operate-type microcommand (op codes 7 through 0). In this case, R register bit 0 contains the most significant bit (MSB) of the jump address which is entered into L register bit 8. The eight least significant bits (LSB) of the jump address are supplied from the A bus to the L register stages 7 through 0. L register stages 9 through 11 remain unchanged.

The microcommands on the RS bus may be modified before storage in the R Registers by modification data received from the U Register, the output data (OD) register, or the eight low-order data switches. For operate-type microcommands with op codes 8 through F, the contents of the 8-bit U Register are OR'ed with the eight high-order bits from control memory when the r field of the microcommand contains code 7. When a Modify Lower Command (MLC) microcommand is being executed, the contents of the 8-bit output data (OD) register are AND'ed with the eight low-order bits from control memory. When an Enter Console Switches (ECS) microcommand is being executed, the states of the eight low-order data switches on the front panel are AND'ed with the eight low-order bits from control memory.

The 16-bit microcommand stored in the R Register is distributed via the R bus to a number of points in the CPU, as shown in Figure 2-5. Major distribution points include the command decoder, which receives all 16 bits, the B bus multiplexer, which receives the eight low-order bits, and the primary/secondary file registers, which receive bits 6 through 11 to select the file register that stores the current data on the A bus. Bits 6 and 7 of a select primary or secondary file microcommand select either the primary or secondary file. Bits 8 through 11 of an operate-type command select one of 15 registers in the selected file.

When the microcommand in the R Register is for an external I/O operation (EXIO), bits 4 through 6 of the lower R Register are transferred to the IC Register for transmittal to I/O devices connected to the byte I/O bus.

2.4.2 CORE MEMORY SECTION DATA FLOW (Figure 2-6)

Figure 2-6 shows the detailed data flow for the Core Memory Section and the backplane connector pin-outs for the core memory bus and the DMA bus. The core memory bus includes the 16-bit memory address lines, the bi-directional 8-bit memory data bus, and memory cycle control signals.

Any one of up to 65,536 8-bit core memory locations may be accessed by the 16-bit address formed by the M and N Registers, or by the DMA channel. The M Register contains the eight high-order bits of the CPU address, and the N Register contains the eight low-order bits. The M and N Registers receive each control memory address from the A bus.

The core memory location addressed by the memory address lines (from the DMA or the M and N Registers) is made available for read or write operations by either the CPU or the DMA channel. If both the DMA channel and the CPU are requesting service simultaneously, the DMA channel has priority. The memory control controls the memory cycle for either the DMA or the CPU. When the DMA is being serviced, the memory data gates, the M and N Register outputs, and the MD Register in the Core Memory Section are all inhibited, disconnecting the CPU from the core memory.

When the CPU is being serviced during a memory read operation, the eight bits from the accessed core memory location are placed on the memory data bus, and are supplied via the memory data gates to the T Register. The output from the T Register is made available to either the B bus multiplexer or to the output data (OD) register.

When the CPU is being serviced during a memory write operation, the data to be written into memory is supplied from the A bus to the T Register, and is subsequently transferred to the MD Register, which supplied the data to core memory.

Control signals on the DMA and memory buses include four lines from the DMA channel to the memory control, one line from the memory control to the DMA channel, and three lines from the memory control to the core memory. Descriptions of these signals follow:

Signal	Source	Destination	Description
DMAR/	DMA	Memory Control	Signal is low (active) when DMA is requesting a memory read or write operation.
DMAS/	DMA	Memory Control	Signal is low (active) when the DMA is selected.
DMAW/	DMA	Memory Control	Signal is low (active) when DMA is requesting a memory write operation.

Signal	Source	Destination	Description
MSBY	Memory Control	DMA	Signal is low (active) while memory is busy with read or write operation.
READ	Memory Control	Core Control	Signal is high when a read/restore operation is in progress. Signal is low when a clear/write operation is in progress.
RTXX/	Memory Control	Core Memory	Negative-going 400-ns read pulse. The leading edge of this pulse defines the beginning of the read portion of a memory read or write cycle.
SPARE/	DMA	Internal Interrupt Control and A Bus	Spare status signal from DMA channel.
WTXX/	Memory Control	Core Memory	Negative-going 350-ns write pulse. The leading edge of this pulse defines the beginning of the write portion of a memory read or write cycle.

2.4.3 PROCESSOR SECTION (Figure 2-7)

The A and B buses are the two main data buses in the Processor Section. All registers in the CPU are accessible (directly and indirectly) for data entry from the A bus. The ALU operates on data from the B bus and the File Registers.

The B bus multiplexer receives 8-bit data from the input data bus (via the input data bus gates), the lower R Register, and the T Register. Either complemented or uncomplemented data from the T Register and the input data bus may be selected. The selected data is supplied from the B bus to the ALU. Depending upon the microcommand being executed, the ALU may pass data from the B bus or the File Registers directly to the A bus multiplexer without modification, or may perform logical or arithmetic operations on the B bus and file register data. Any of the following arithmetic/logic operations may be performed:

a. Logical 1. All bits from the ALU are forced to the 1 state.

b. Logical AND. The data on the B bus is AND'ed with the data from the File Registers.

c. Logical OR. The data on the B bus is OR'ed with the data from the File Registers.

d. Exclusive OR. An exclusive OR operation is performed on the data from the B bus and the data from the File Registers.

e. Add. A binary addition is performed on the B bus and file register data. Subtraction is accomplished by complementing the subtrahend (complementing is performed before gating the data to the B bus), and then adding. The state of the initial carry bit (CYIA) determines whether a one's complement or two's complement subtraction is performed. When initial carry is a 1, a two's complement subtraction is performed. When initial carry is a 0, a one's complement subtraction is performed.

f. Shift Left. The data from the B bus or the File Registers is shifted one bit position towards the most significant bit of the ALU output.

The A bus multiplexer receives the 8-bit output from the ALU, and receives the following additional data:

a. Internal Status Signals. The eight internal status line inputs include four spare lines, the real time clock interrupt line, the power-fail line, and two lines for front panel step and interrupt commands.

b. Sense Switch Signals. The four lines from the four front panel sense switches are supplied to the A bus multiplexer.

c. File Register Data. Bits 1 through 7 from the File Registers are supplied to the A bus multiplexer. The eighth bit supplied to the A bus multiplexer is formed by the initial carry logic (CYIA).

The 8-bit byte gated to the A bus may be any one of the following, depending upon the microcommand being executed:

a. Processed Data From the ALU. Arithmetic/logic unit data is placed on the A bus except when an EIS, ESS, SRF, SRI, or SFR command is being executed.

b. Right-Shifted File Register Data. This action is performed in response to the SFR or SRI (shift file right) commands. File register bits 1 through 7 occupy bit positions 0 through 6 of the A bus output, and bit 7 is signal CYIA from the initial carry logic. The state of CYIA is 0 for an SFR microcommand, and is 1 for an SRI microcommand.

c. 4 MSB of File Register Data. This action is performed in response to the SRF (shift right four) command. File Register bits 4 through 7 occupy bit positions 0 through 3 of the A bus output. Bits 4 through 7 from the ALU are forced to 1, and occupy bit positions 4 through 7 of the R bus output.

d. Sense Byte. This action is performed in response to the ESS center sense switches) command. Bits 0 through 3 from the ALU are forced to 1, and occupy positions 0 through 3 of the A bus output. The current states of sense switches 4 through 7 occupy bit positions 4 through 7 of the A bus output.

e. Internal Status Byte. This action is performed in response to the EIS (enter internal status) command. The states of the eight A bus output lines are determined by the states of the eight internal status signal inputs.

The selected data byte presented on the A bus is distributed to the L, M, N, T, and U Registers, to the File Registers, and to a number of miscellaneous points shown in Figure 2-7. If the File Registers are enabled, code bits 8 through 11 from the R bus select one of the 8-bit file register locations in which the data from the A bus is stored.

2.4.4 STATUS AND FILE ZERO FLAG DATA FLOW (Figure 2-8)

These data flow paths are involved with the processing of internal and external interrupts, I/O device requests, condition codes, and status signals. Two major data flow paths are involved; the internal status signal data path, and the file zero flag data path.

2.4.4.1 INTERNAL STATUS SIGNAL DATA FLOW. Internal status signals are received from the real time clock, the power fail circuits, the front panel control circuits, the Machine State Control and Clock Circuits, and four spare inputs. The eight lines from these units are supplied to the A bus multiplexer and to the internal status interrupt control. When one or more of the eight lines is active (low), signal INTL/ from the internal status interrupt control goes low. Subsequently, in response to an Enter Internal Status (EIS) command, the internal status signals are gated to the A bus as an internal status byte. The format of the internal status byte is as follows:

A Bus Bit Position	Description
0	This bit assumes the state of panel interrupt signal INTC/ from the Machine State Control and Clock Circuits. When INTC/ is active (low), indicating that the front panel INT switch has been activated, bit 0 of the internal status byte is 1.
1	Signal SPARE/ from the DMA channel is a DMA termination, and is a spare status line in the basic system.
2	This bit assumes the state of real time clock interrupt RTCI/ from the real time clock. When RTCI/ is active (low), indicating the completion of a real time clock counting cycle, bit 2 of the internal status byte is 1.

A Bus Bit Position	Description
3	Signal MPRO/ is a spare status line in the basic system.
4	Signal MPAR/ is a spare status line in the basic system.
5	Signal MBDY/ is a spare status line in the basic system.
6	This bit assumes the state of power-fail interrupt PWRF/ from the power-fail circuits. When PWRF/ is active (low), indicating a power failure or power-start, bit 6 of the internal status byte is 1.
7	This bit assumes the state of panel step switch signal STPF/ from the Front Panel Control Circuits. When signal STPF/ is active (low), indicating the front panel STEP switch has been activated, bit 7 of the internal status byte is 1.

2.4.4.2 FILE ZERO FLAG DATA FLOW. An 8-bit file zero flag byte is presented to the ALU by the file register output buffer when enabled by signal FAZO from the file register control. The byte from the ALU is supplied to the A bus multiplexer. Signal FAZO is active when the f field of an operate-type command contains code 0000, selecting file zero. When FAZO is active, the eight file zero flag lines from the internal status interrupt control, the external interrupt control, the I/O bus, the Teletype, and the condition code flip-flops are gated into the file register output buffer. The format of the file zero flag byte is as follows:

File Register Bit Position	Description
0	This bit assumes the state of condition code flip-flop CC0. Flip-flop CC0 assumes the state of signal AFLT/ from the overflow decode. When AFLT/ is active (low), indicating an overflow occurred during an add, shift, increment, subtract, decrement, or copy operation, bit 0 is 1. An overflow occurs when a one is shifted out of the end bit during a shift operation, or when the carry-out of the high order bit differs from the carry-in to the high-order bit during an arithmetic operation.
1	This bit assumes the state of condition code flip-flop CC1. Flip-flop CC1 is active when the sign for the result of an arithmetic operation is negative, which occurs when the most significant bit (bit 7) of the result on the A bus is a 1. When flip-flop CC1 is active, bit 1 is 1.

File Register Bit Position	Description
2	This bit assumes the state of condition code flip-flop CC2. Flip-flop CC2 is active when a zero condition is present in the result of an operate-type command execution. The zero condition exists when AZ07/ from the A bus zero decode is low (active), indicating the A bus contains an all zero byte, provided that additional conditions imposed by the command in process are satisfied. (No additional conditions are imposed for op code 7 commands. For op code 8, 9, B, C, D, E, or F commands, the link bit (Bit 7) of the command must be a zero or flip-flop CC2 must already be in the set condition).
3	This bit is a 1 when the concurrent I/O request line (ECIO/) from the I/O bus is active (low), indicating an external I/O device is requesting an input or output data transfer in the concurrent (block transfer) mode.
4	This bit is a 1 when signal INTL/ is active (low), indicating one or more of the internal status lines is active.
5	Signal IRPY (I/O reply) from the I/O bus is not implemented in the basic system.
6	When the L Save Stack Option is not included, this bit is a 1 when the Teletype is sending a space, and is in the 0 state when the Teletype is sending a mark. When the L Save Stack Option is included, the teletype input is replaced by stack overflow indicator STOV/, which is active (low) when a stack overflow exists.
7	This bit is a 1 when output INTR/ from the external interrupt control is active (low). Signal INTR/ is active when an external interrupt request is received from one of the I/O devices on the I/O bus (EINT/ low).

2.4.5 DATA DISPLAY INDICATOR DATA FLOW (Figure 2-9)

Data displayed in the 16 front panel data display lamps are selected by the display select pushbutton switches on the front panel, as follows:

a. D Pushbutton. When this switch is pressed, the eight bits from the A bus are displayed in lamps 7 through 0.

b. M Pushbutton. When this switch is pressed, the 16-bit core memory address (or other data) contained in the M and N Registers, is displayed. Lamps 15 through 7 display the contents of the M Register, and lamps 7 through 0 display the contents of the N Register.

c. L Pushbutton. When this switch is pressed, the 12-bit control memory address, contained in the L Register, is displayed in lamps 11 through 0.

d. C Pushbutton. When this switch is pressed, the 16 bits on the RS bus are displayed. The source of the data on the RS bus may be the control memory, or the front panel data switches. Although the L Save Register is also connected to the RS bus, the contents of this register cannot be observed on the display.

2.4.6 BYTE I/O BUS DATA FLOW (Figure 2-10)

Figure 2-10 shows the detailed data flow, as well as the backplane connector pin-outs, for the byte I/O bus that interfaces the CPU with various types of I/O devices. Eight-bit output data bytes to the byte I/O bus are supplied from the output data (OD) register, which receives inputs from the T Register. Eight-bit input data bytes from the byte I/O bus are supplied via the input data bus gates to the B bus in the Processor Section. The IC Register receives I/O control codes from the lower R Register, and supplies control codes to the devices connected to the byte I/O bus. The definition of the I/O control codes is given in Figure 2-10.

Control signals supplied from the CPU to the byte I/O bus include the following:

a. MRST/. Master reset signal MRST/ is active (low) when the operator activates the front panel RESET switch, or when a power failure occurs.

b. RINH/. Run inhibit RINH/ is active (low) when the CPU is executing a skip or jump.

c. CSTP/. The clock stop line is active (low) when the CPU is in the HALT state or when a skip, jump, or memory stop condition is present.

d. CPH1, CPH2/. These two 5-MHz clock phases are used to synchronize I/O devices with the CPU.

Control signals supplied from the I/O bus to the CPU include the following:

a. EINT/. An I/O device makes the external interrupt line low to request an interrupt.

b. ECIO/. An I/O device makes the concurrent I/O request line low in order to request a concurrent mode (block transfer) input or output data transfer.

c. IRPY/. An I/O device makes the I/O reply line low in response to an I/O control code from the CPU when closed loop operation is required. The signal is not used in the basic system.

An I/O operation is initiated by placing an 8-bit control byte, containing a 5-bit device address and a 3-bit function code, in the OD Register. Also, the I/O control

code (1) for control out is loaded into the IC Register and sent to the I/O devices. The device whose address corresponds to the address code in the control byte accepts and decodes the function code, and executes the specified action.

Each output data transfer must be preceded by a Data Out (code 2) I/O control code from the IC Register. Each input data transfer must be preceded by a Data In (code 6) I/O control code from the IC Register.

In the concurrent (block transfer) mode of operation, the I/O device issues a low ECIO/ output when the device is ready to transfer a data byte to or from the CPU. The CPU responds by issuing I/O control code 4 (concurrent acknowledge) from the IC Register. The I/O device responds to the concurrent acknowledge by placing its device address on the input bus. After receipt of the device address, the CPU executes the data transfer by issuing the I/O control code for Data In or Data Out.

When an I/O device issues a low EINT/ output to the CPU to request an interrupt, the CPU responds by issuing I/O control code 5 (interrupt acknowledge) from the IC Register. The I/O device responds to the interrupt acknowledge by placing its device address on the input bus to identify the source of the interrupt.

The SELO/ and PROT/ lines from the CPU to the byte I/O bus are hard-wired to ground. All devices on the byte I/O bus are connected in daisy-chain fashion to the SELO/ and PROT/ lines. The SELI/ and PRIN/ lines of the highest-priority device are connected directly to the SELO/ and PROT/ lines from the CPU. The SELO/ and PROT/ lines from the highest-priority device are connected to the SELI/ and PRIN/ lines of the second-highest-priority device, and so on, to the lowest-priority device in the daisy-chain. The SELO/ and PROT/ ground signals are propagated through the daisy-chain until they reach a device that is requesting priority. The highest-priority device requesting priority captures SELO/ or PROT/, preventing lower-priority devices from the requesting service until the device that captured PROT/ or SELO/ has been serviced.

2.5 COMMAND DECODING (Tables 2-1 through 2-4)

Table 2-1 is a glossary that lists and describes the command decode signals. Tables 2-2 through 2-4 list the microcommands in order by their hexadecimal operation code, and indicate the active command decode signals for each command. Tables 2-2 through

2-4 also indicate the type of fetch/execute cycle (refer to Paragraph 2.6.1) performed for each microcommand. Table 2-2 lists the commands with operation codes 0 through 6. Table 2-3 lists the commands with operation code 7. Table 2-4 lists the commands with operation codes 8 through F. The majority of the command decode signals listed in Table 2-1 originate in the command decoder (Figure 2-5). The remainder of the command decodes are produced by file register control (FBS1 and FBS2), initial carry logic (CYIA), L Save Control (RTNX), and U Register enable (RENU). The activated command decode signals listed in Tables 2-2 through 2-4 initiate and control the sequence of actions performed for each microcommand. All decoding action is performed on the contents of the upper and lower R Registers, which contain the current microcommand received from the control memory or the front panel switches. Figures 2-3 through 2-12 show the distribution of the command decode signals to the various functional elements of the CPU.

TABLE 2-1. COMMAND DECODE SIGNAL GLOSSARY

COMMAND DECODE	LOGIC DIAGRAM ORIGIN	DEFINITION
ADDX	0372	Enable overflow decode for op codes 1, 2, 3, 6, 8, 9, A, and B
ADXX	0370	ADD (op code 8) command
AENI	0373	Gate status byte to A bus
AENP	0373	Gate the eight low-order console data switches to lower R Register
AENS	0373	Gate sense switches onto A bus
AFXX	0370	Add file (AF) command (op code 3)
ANDX	0372	Perform logical AND operation in arithmetic/logic unit
ANXX	0370	AND With File (AND) command (op code E)
ASRF	0373	Shift File Right Four (SRF) command (op code 7X2X)
BENI	0374	Gate data from input bus to B bus
BENR	0374	Gate low-order R Register bits to B bus
BNTM	0374	Gate complemented T Register bits to B bus

TABLE 2-1. COMMAND DECODE SIGNAL GLOSSARY (cont)

COMMAND DECODE	LOGIC DIAGRAM ORIGIN	DEFINITION
BNTP	0374	Gate uncomplemented T Register bits to B bus
BTXR	0374	Transfer in progress from T Register to B bus
CG1B	0371	Op code 1B (L Save command group)
CG10	0371	Op code 10 (Load Zero command group)
CG7X	0371	Op code 17 (Load Seven command group)
CONT	0371	Return L Save to L Register
CPXX	0370	Compare File (CP) command (op code 6)
CYIA	0560	Initial carry
CYXX	0370	Copy (CPY) command (op code B)
EXIO	0372	External I/O operation (op codes 7X8X thru 7XFX)
FBCL	0371	File register control clock
FBS1	0550	Select primary file
FBS2	0550	Select secondary file
FINH	0372	File inhibit
IDHT	0372	Select uncomplemented bits from input data bus
IOES	0372	Op codes 7X1X, 7X2X, 7X4X, 7X7X (ESS, SRF, EIS, or ECS)
IOXX	0370	Op code 7
JPXX	0370	Jump Extended (JE) command (op code 0)
JUXX	0371	Modify Lower (MLC) command (op code 1A)
LFXX	0370	Load File With Literal (LF) command (op code 2)
LRLL	0372	Load L Register with jump or return jump address
LRXX	0370	Load literal into selected register (op code 1)
LROX	0371	Load L Register with jump address between 000 and OFF, or between 200 and 2FF
LRO1	0372	Load L Register with address

TABLE 2-1. COMMAND DECODE SIGNAL GLOSSARY (cont)

COMMAND DECODE	LOGIC DIAGRAM ORIGIN	DEFINITION
LR1X	0371	Load L Register with jump address between 100 and 1FF, or between 300 and 3FF
LR2X	0371	Op code 1C (jump in 1K)
LR3X	0371	Op code 1D (jump in 1K)
MNRX	0372	Load M or N Register
MRXX	0371	Load M Register
MYDX	0374	Code 01 (decrement file) in bits 7 and 6 of memory read or write command (op code A)
MYXX	0370	Memory read or write command (op code A)
NRXX	0371	Load N Register
ORXX	0370	Logical OR with file (LOR) command (op code C)
RENU	0362	OR U Register bits with high-order bits on RS bus
RTJX	0371	Return to address in L Save Register (op codes 1020, 1060, 10A0)
RTNX	0320	Return to address in L Save Register (op codes 0, 1020, 1060, 10A0, 19)
SBXX	0370	Subtract File (SBT or SBO) command (op code 9)
SHLX	0372	Shift file left (SFL or SLI) command (op code F)
SHRX	0373	Shift file right (SFR or SRI) command (op code F)
SHXX	0370	Shift file left or right command (op code F)
TNXX	0370	Test Not Zero (TN) command (op code 5)
TONY	0372	Enable BENI when T select bit (bit 5) = 1 for op codes 8, 9, A and B
TRTX	0371	Return, Load T (RLT) command (op code 19)
TRXX	0371	Load T Register
TZXX	0370	Test If Zero (TZ) command (op code 4)
URXX	0371	Load U Register
XRXX	0370	Exclusive OR with File (XOR) command (op code D)

TABLE 2-2. OP CODES 0 THROUGH 6 COMMAND DECODES

	COMMAND	MNE-MONIC	CYCLE TYPE	R REGISTER CODE (HEXADECIMAL) BITS 15-12	BITS 11-8	BITS 7-4	BITS 3-0
	JUMP EXTENDED	JE	I	0	**X	**X	**X
LOAD ZERO (LZ)	NO OPERATION	NOP	VI	1	0	0	0
LOAD ZERO (LZ)	ENABLE COMM RATES	ECR	VI	1	0	0	1
LOAD ZERO (LZ)	DISABLE COMM RATES	DCR	VI	1	0	0	2
LOAD ZERO (LZ)	INPUT COMM RATES	ICR	VI	1	0	0	4
LOAD ZERO (LZ)	UNASSIGNED		VI	1	0	0	8
LOAD ZERO (LZ)	UNASSIGNED		VI	1	0	1	0
LOAD ZERO (LZ)	RETURN	RTN	II	1	0	2	0
LOAD ZERO (LZ)	SELECT PRIMARY FILE	SPF	VI	1	0	4	0
LOAD ZERO (LZ)	RETURN, SELECT PRIMARY FILE	RSP	II	1	0	6	0
LOAD ZERO (LZ)	SELECT SECONDARY FILE	SSF	VI	1	0	8	0
LOAD ZERO (LZ)	RETURN, SELECT SECONDARY FILE	RSS	II	1	0	A	0
	LOAD T	LT	VI	1	1	*0-F	*0-F
	LOAD M	LM	VI	1	2	*0-F	*0-F
	LOAD N	LN	VI	1	3	*0-F	*0-F
	JUMP IN 1K	JP	I	1	**4	**0-F	**0-F
					**5	**0-F	**0-F
					**C	**0-F	**0-F
					**D	**0-F	**0-F
	LOAD U	LU	VI	1	6	*0-F	*0-F
LOAD SEVEN (LS)	DISABLE EXTERNAL INTERRUPTS	DEI	VI	1	7	0	4
LOAD SEVEN (LS)	ENABLE EXTERNAL INTERRUPTS	EEI	VI	1	7	0	8
LOAD SEVEN (LS)	DISABLE REAL TIME CLOCK	DRT	VI	1	7	1	0
LOAD SEVEN (LS)	ENABLE REAL TIME CLOCK	ERT	VI	1	7	2	0
LOAD SEVEN (LS)	HALT	HLT	VI	1	7	8	0
	LOAD EIGHT	LE	VI	1	8	*0-F	*0-F
	RETURN, LOAD T	RLT	II	1	9	*0-F	0-F
	MODIFY LOWER COMMAND	MLC	II	1	A	0	0
	INHIBIT L SAVE	ILS	VI	1	B	0	0
	L SAVE STACK OPTION COMMANDS		VI	1	B	0,9,A,C	0-F
	LOAD FILE WITH LITERAL	LF	VI	2	***0-F	*0-F	*0-F
	ADD FILE WITH LITERAL	AF	VI	3	***0-F	*0-F	*0-F
	TEST IF ZERO	TZ	III	4	***0-F	*0-F	*0-F
	TEST NOT ZERO	TN	III	5	***0-F	*0-F	*0-F
	COMPARE FILE	CP	III	6	***0-F	*0-F	*0-F

ACTIVE COMMAND DECODES: ADDX, AFXX, ANDX, BENR, CG1B, CG1O, CG7, CONT, CPXX, FBS1, FBS2, FBCL, FINT, JPXX, JUXX, LFXX, LRLL, LRXX, LROX, LRO1, LR1X, LR2X, LR3X, MMRX, MRXX, NRXX, RENU, RTJX, RTNX, TMXX, TRTX, TRXX, TZXX, URXX

* LITERAL
** JUMP ADDRESS
*** ADDRESS FILE

■ SIGNAL IS ACTIVE (XXXX = HIGH; XXXX/ = LOW)
▨ SIGNAL IS ACTIVE DURING SECOND MICROSTEP OF A RETURN EXECUTION

TABLE 2-3. OP CODE 7 COMMAND DECODES

COMMAND	MNE-MONIC	CYCLE TYPE NOTE	"R" REGISTER CODE (HEXADECIMAL) OP CODE (BITS 15-12)	f FIELD (BITS 11-8)	c FIELD (BITS 7-4)	r FIELD* (BITS 3-0)
ENTER SENSE SWITCHES	ESS	I,VI	7	0-F	1	0-7,8-F
SHIFT FILE RIGHT 4	SRF	I,VI	7	0-F	2	0-7,8-F
ENTER INTERNAL STATUS	EIS	I,VI	7	0-F	4	0-7,8-F
ENTER CONSOLE SWITCHES	ECS	I,VI	7	0	7	0
CLEAR I/O MODE	CIO	I,VI	7	0-F	8	0-7,8-F
CONTROL OUTPUT	COX	I,VI	7	0-F	9	0-7,8-F
DATA OUTPUT	DOX	I,VI	7	0-F	A	0-7,8-F
SERIAL TTY SPACE	SOX	I,VI	7	0-F	B	0-7,8-F
CONCURRENT ACK	CAK	I,VI	7	0-F	D	0-7,8-F
INTERRUPT ACK	IAK	I,VI	7	0-F	C	0-7,8-F
DATA INPUT	DIX	I,VI	7	0-F	E	0-7,8-F
STACK/SPARE INPUT	SIX	I,VI	7	0-F	F	0-7,8-F

ACTIVE COMMAND DECODES: AENI, AENP, AENS, AMDX, ASNF, BENI, EXIO, IOMT, IOES, IOXX

* BIT 3 OF r FIELD IS FILE INHIBIT. WHEN BIT 3 = 1, STORAGE IN FILE DESIGNATED BY F FIELD IS INHIBITED.

■ SIGNAL IS ACTIVE (XXXX = HIGH; XXXX/ = LOW)

NOTE: FOR JUMP IN 1/2K (CODE 4_8 OR 5_8 IN r FIELD), A TYPE I CYCLE IS PERFORMED; OTHERWISE, A TYPE VI CYCLE IS PERFORMED.

TABLE 2-4. OP CODES 8 THROUGH F COMMAND DECODES

COMMAND	MNEMONIC	CYCLE TYPE (NOTE 7)	OP CODE (BITS 15-12)	f FIELD** (BITS 11-8)	c FIELD (BITS 7-4)	r FIELD* (BITS 3-0)	ACTIVE COMMAND DECODES: ADDX ADDX ANDX ANDX BENI BNTM BNTP BTXR CYIA CYAX FINI IDMT MYDX MYXX ORXX SBXX SHLX SHRX SHXX TOMY XRXX	REMARKS
MOVE FILE	MOV	I,VI	8	0-F	000C	0-F	■ ■	
ADD LINK TO FILE	ALF	I,VI	8	0-F	100C	0-F	■ ■ ▨	NOTE 5
ADD FILE	ADD	I,VI	8	0-F	L01C	0-F	■ ▨ ▨ ▨ ■	NOTES 1,3,4,5
					011C	0-F	■ ▨ ▨ ▨ ■	NOTES 1,3,4
INCREMENT FILE	INC	I,VI	8	0-F	010C	0-F	■ ■	
SUBTRACT FILE TWO'S COMPLEMENT	SBT	I,VI	9	0-F	100C	0-F	■ ▨ ▨ ■	NOTE 5
					101C	0-F	■ ▨ ▨ ▨ ■	NOTES 1,2,4,5
					000C	0-F	■ ■	
					001C	0-F	■ ▨ ▨ ■	NOTES 1,2,4
SUBTRACT FILE ONE'S COMPLEMENT	SBO	I,VI	9	0-F	110C	0-F	■ ▨ ▨ ■	NOTE 5
					111C	0-F	■ ▨ ▨ ▨ ■	NOTES 1,2,4,5
					010C	0-F	■ ■	
					011C	0-F	■ ▨ ▨ ■	NOTES 1,2,4
READ MEMORY FULL CYCLE	RMF	I,VI	A	0-F	4	2	■ ■ ■	NOTE 5
					4	0,1,3-F	■ ■ ■	
					0	0-F	■ ■ ■	
					8	0-F	■ ■ ■	NOTE 5
					C	0-F	■ ■ ■	
READ MEMORY HALF CYCLE	RMH	I,VI	A	0-F	6	2	■ ■ ■	NOTE 5
					6	0,1,3-F	■ ■ ■	
					2	0-F	■ ■ ■	
					A	0-F	■ ■ ■	NOTE 5
					E	0-F	■ ■ ■	
WRITE MEMORY FULL CYCLE	WMF	I,VI	A	0-F	5	2	■ ■ ■	NOTE 5
					5	0,1,3-F	■ ■ ■	
					1	0-F	■ ■ ■	
					9	0-F	■ ■ ■	NOTE 5
					D	0-F	■ ■ ■	
WRITE MEMORY HALF CYCLE	WMH	I,VI	A	0-F	7	2	■ ■ ■	NOTE 5
					7	0,1,3-F	■ ■ ■	
					3	0-F	■ ■ ■	
					B	0-F	■ ■ ■	NOTE 5
					F	0-F	■ ■ ■	
COPY	CPY	I,VI	B	0-F	100C	0-F	■ ■ ■	NOTE 5
					110C	0-F	■ ■ ■	
					101C	0-F	■ ▨ ▨ ▨ ■	NOTES 1,3,4,5
					011C	0-F	■ ▨ ▨ ▨ ■	NOTES 1,3,4
					111C	0-F	■ ▨ ▨ ▨ ■	NOTES 1,3,4
					001C	0-F	■ ▨ ▨ ▨ ■	NOTES 1,3,4
ZERO FILE/REG	ZOF	I,VI	B	0-F	000C	0-F	■ ■	
+1 TO FILE/REG	POF	I,VI	B	0-F	010C	0-F	■ ■	
LOGICAL OR WITH FILE	LOR	I,VI	C	0-F	L11C	0-F	▨ ▨ ■	NOTE 6
					L10C	0-F	▨ ▨ ▨ ■	NOTES 1,2,4,6
					L01C	0-F	▨ ▨ ▨ ■	NOTES 1,3,4
MOVE FILE	MOV	I,VI	C	0-F	L00C	0-F	▨ ▨ ■	
EXCLUSIVE OR WITH FILE	XOR	I,VI	D	0-F	L11C	0-F	▨ ▨ ■	NOTE 6
					L01C	0-F	▨ ▨ ▨ ■	NOTES 1,3,4
					L10C	0-F	▨ ▨ ▨ ■	NOTES 1,2,4,6
					L00C	0-F	▨ ▨ ■	
AND WITH FILE	AND	I,VI	E	0-F	L11C	0-F	■ ■ ■	NOTE 6
					L01C	0-F	■ ▨ ▨ ■	NOTES 1,3,4
					L10C	0-F	■ ▨ ▨ ■	NOTES 1,2,4,6
					L00C	0-F	■ ■ ■	
SHIFT FILE LEFT	SFL	I,VI	F	0-F	000C	0-F	■ ■	
					100C	0-F	■ ■ ■	NOTE 5
SHIFT FILE LEFT & INSERT	SLI	I,VI	F	0-F	010C	0-F	■ ▨ ▨ ■	NOTES 1,2,4
SHIFT FILE RIGHT	SFR	I,VI	F	0-F	001C	0-F	■ ▨ ▨ ■	NOTES 1,3,4
					101C	0-F	■ ▨ ▨ ▨ ■	NOTES 1,3,4,5
SHIFT FILE RIGHT & INSERT	SRI	I,VI	F	0-F	011C	0-F	■ ▨ ▨ ▨ ■	

c FIELD FORMAT FOR OP CODES 8-F

BIT NO.	APPLICABLE OP CODES	DEFINITION
7	8,9,B,F	1 = Add link, and disable update of zero condition flag if flag is not already set.
	C,D,E	1 = disable update of zero condition flag if flag is not already set.
	F	1 = Insert link in vacated bit position of shifted file data, and disable update of zero condition flag if flag is not already set.
6	8,B	1 = Add 1
	9	1 = One's Complement Subtraction; 0 = Two's Complement Subtraction
	C,D,E	1 = Select complemented T Register bits or complemented input data bus bits for presentation on B Bus (Notes 1 and 2).
	F	1 = Insert 1 in vacated bit position of shifted file data.
7 6		
0 0	A	Copy file
0 1	A	Decrement file and add memory link if r field is selecting M or N Register
1 0	A	Add link to file
1 1	A	Increment file
5	8,B,C,D,E	1 = Select uncomplemented T Register bits or uncomplemented input data bus bits for presentation on B Bus (Notes 1 and 3).
	8,B	0 = Place all zero's on B Bus.
	C,D,E	0 = Place all zero's on B Bus if bit 6 = 0. 1 = Place all one's or B Bus if bit 6 = 1.
	9	1 = Select complemented T Register bits or complemented input data bus bits for presentation on B Bus (Notes 1 and 3). 0 = Place all one's on B Bus.
	A	1 = Half cycle; 0 = Full cycle
	F	1 = Shift right; 0 = Shift left
4	8,9,B,C,E,F	1 = Update Condition Code Flags
	A	1 = Write; 0 = Read

r FIELD DECODES FOR OPERATE-TYPE COMMANDS

CODE (BITS 2-0)	SELECTED REGISTER	DESTINATION DECODES
0_8	None	None
1_8	T	TRXX
2_8	M	MRXX, MNRX
3_8	N	NRXX, MNRX
4_8	L (jump in lower 1/2K)	LROX, LRO1, LRLL
5_8	L (jump in upper 1/2K)	LR1X, LRO1, LRLL
6_8	U	URXX
7_8	Upper R (or U Register with 8 high-order bits on RS Bus)	CG7, RENU

NOTES

1. BENI IS ACTIVE IF IC REGISTER BIT 3 IS A "1", SELECTING DATA FROM THE INPUT DATA BUS
2. BNTM IS ACTIVE IF BENI IS INACTIVE, SELECTING COMPLEMENTED T REGISTER DATA
3. BNTP IS ACTIVE IF BENI IS INACTIVE, SELECTING UNCOMPLEMENTED T REGISTER DATA
4. BTXR IS ACTIVE IF EITHER (BUT NOT BOTH) BNTM OR BNTP IS ACTIVE
5. CYIA IS ACTIVE IF LINK CONTROL (c FIELD BIT 7) AND LINK BIT ARE BOTH "1".
6. MYDX IS ACTIVE IF LINK CONTROL (c FIELD BIT 7) IS A "0".
7. FOR JUMP IN 1/2 K (CODE 4_8 OR 5_8 IN r FIELD), A TYPE I CYCLE IS PERFORMED; OTHERWISE A TYPE VI CYCLE IS PERFORMED

■ SIGNAL IS ACTIVE (XXXX = H; XXXX/ = L)

▨ SIGNAL IS ACTIVE (XXXX = H; XXXX/ = L) WHEN ADDITIONAL CONDITIONS IN REMARKS COLUMN ARE SATISFIED

*BIT 3 OF r FIELD IS FILE WRITE INHIBIT. WHEN BIT 3 IS A "1", STORAGE IN THE FILE DESIGNATED BY THE f FIELD IS INHIBITED

**f FIELD SELECTS THE FILE THAT CONTAINS THE OPERAND. DATA FROM THE A BUS IS WRITTEN BACK INTO THE FILE IF BIT 3 IS A "0"

2.6 CPU TIMING CONSIDERATIONS

All major timing considerations for the CPU, including fetch/execute cycles, internal clock timing, L Register timing, and memory cycle timing, are discussed under this paragraph.

2.6.1 FETCH/EXECUTE CYCLES (Figure 2-11)

The fetch/execute cycle consists of the sequence of actions performed by the Control Unit to fetch and execute microcommands from control memory. Six types of fetch/execute cycles are employed. The type of cycle used for each command is indicated in Tables 2-2 through 2-4. The six cycle types are illustrated in Figure 2-13. As shown in this illustration, the sequence of actions performed during a fetch/execute cycle is divided into 200-nanosecond microsteps, defined by the basic 5-MHz CPU clock (SCK6). The principal elements involved in the fetch/execute cycle are the R Register, the L Register, the command decoder, and the Machine State Control and Clock Circuits.

During microstep 0, the command to be executed is fetched (accessed) from control memory. Also, the command loaded into the R Register at the microstep preceding microstep 0 is executed. In microstep 1, the command fetched in microstep 0 is loaded into the R Register and executed. For cycle types I, II, III, IV, and V, microsteps 2 and up are also required for execution.

The following subparagraphs describe each type of fetch/execute cycle.

2.6.1.1 CYCLE TYPE I. This cycle is performed for the jump extended (op code 0) and jump in 1K (op code 14, 15, 1C, or 1D) commands. For these commands, the low-order bits (bits 0 through 11 for jump extended, bits 0 through 8 and 11 for jump in 1K) contain a jump address, which is entered into the R Register and subsequently transferred to the L Register. The U Register must be cleared for proper execution of a jump extended microcommand. The L Register then accesses the command at the jump address. Total execution time is 400 nanoseconds.

During microstep 1, the jump command is loaded into the R Register and decoded, and the L Register advances to the next address.

During microstep 2, a run inhibit/clock stop commences and the L Register is inhibited from advancing to the next address stored in the R Register. If a previously executed Inhibit L Save (ILS) command has not disabled the L Save Register, the address in the L Register is transferred to the L Save Register. Finally, the jump address in the R Register is transferred to the L Register. Bits 8 through 11 of the R Register are supplied directly to the L Register. Bits 0 through 7 of the R Register are supplied via the B bus, the arithmetic/logic unit, and the A bus to the L Register.

During microstep 3, the microcommand from the jump address in control memory is loaded into the R Register and executed, and the L Register is advanced to fetch the next command from control memory.

2.6.1.2 CYCLE TYPE II. This cycle is performed for return jump commands (op codes 1020, 1060, 10A0, and 19), in which the address stored in the L Save Register is entered back into the L Register. The microcommand at the L Save address in control memory is then fetched and executed. Total execution time is 600 nanoseconds.

During microstep 1, the return jump command is loaded into the R Register and decoded, and the L Register is advanced to the next address. The control memory is inhibited, preventing the addressed command from being supplied to the R Register.

During microstep 2, op code 0 is forced into R Register stages R15 through R12, and the address from the L Save Register is loaded into R Register stages R11 through R00. Also, the L Register is inhibited from advancing to the next address, and the address in the L Register is transferred to the L Save Register.

During microstep 3, the L Save address from R Register stages R11 through R00 is transferred to the L Register. Bits R11 through R08 are transferred directly to the L Register. Bits R07 through R00 are transferred via the B bus, the arithmetic/logic unit, and the A bus to the L Register. Also, a run inhibit/clock stop commences, and the L Register is inhibited from advancing to the next address.

During microstep 4, the microcommand from the L Save address in control memory is loaded into the R Register and executed, and the L Register is advanced to fetch the next command from control memory.

CYCLE TYPE	MICROSTEP NO. (EACH STEP IS 200 NANOSECONDS)						
	FETCH	EXECUTE					
	0	1	2	3	4	5	6
I (JUMP) (OP CODE 0, 14, 15, 1C, 1D)	L REGISTER FETCHES JUMP COMMAND FROM CONTROL MEMORY	JUMP COMMAND → R REGISTER	L REGISTER → L SAVE REGISTER RUN INHIBIT/CLOCK STOP	LOAD COMMAND FROM JUMP ADDRESS INTO R REGISTER AND EXECUTE			
	CPU EXECUTES PRECEDING COMMAND	+1 → L REGISTER	INHIBIT L REGISTER ADVANCE JUMP ADDRESS → L REGISTER	+1 → L REGISTER (JUMP ADDRESS)+1			
		EXECUTE JUMP COMMAND					
II (RETURN) (OP CODE 1020, 1060, 10A0, 19	L REGISTER FETCHES RETURN COMMAND FROM CONTROL MEMORY	RETURN COMMAND → R REGISTER	OP CODE 0 → R15 - R12	R11 - R00 → L REGISTER (L SAVE ADDRESS)	LOAD COMMAND FROM L SAVE ADDRESS INTO R REGISTER AND EXECUTE		
	CPU EXECUTES PRECEDING COMMAND	+1 → L REGISTER	L SAVE → R11 - R00 INHIBIT L REGISTER ADVANCE	RUN INHIBIT/CLOCK STOP			
		INHIBIT CONTROL MEMORY OUTPUT	L REGISTER → L SAVE REGISTER	INHIBIT L REGISTER ADVANCE	+1 → L REGISTER (L SAVE ADDRESS +1)		
		EXECUTE RETURN COMMAND					
III (SKIP-TYPE) (OP CODE 4, 5, OR 6)	L REGISTER FETCHES TYPE III COMMAND FROM CONTROL MEMORY	TYPE III COMMAND → R REGISTER	IF COMPARE OR TEST WAS AFFIRMATIVE (SKIP):	LOAD COMMAND AT SKIP ADDRESS +1 INTO R REGISTER AND EXECUTE			
		+1 → L REGISTER (SKIP ADDRESS)	+1 → L REGISTER (SKIP ADDRESS +1) RUN INHIBIT/CLOCK STOP PREVENTS EXECUTION OF COMMAND AT SKIP ADDRESS				
	CPU EXECUTES PRECEDING COMMAND	EXECUTE COMPARE OR TEST	IF COMPARE OR TEST WAS NOT AFFIRMATIVE (NO SKIP): EXECUTE COMMAND AT SKIP ADDRESS	+1 → L REGISTER (SKIP ADDRESS + 2)			
			+1 → L REGISTER (SKIP ADDRESS +1)				
		EXECUTE TYPE III COMMAND					
IV (HALF CYCLE READ OR WRITE) (OP CODE AX2, AX3, AX6, AX7, AXA, AXB, AXE, AXF)	L REGISTER FETCHES HALF CYCLE READ OR WRITE COMMAND FROM CONTROL MEMORY	LOAD HALF CYCLE READ OR WRITE COMMAND INTO R REGISTER	LOAD NEXT COMMAND INTO R REGISTER (EXECUTE IF NO CLOCK STOP)	LOAD NEXT COMMAND INTO R REGISTER AND EXECUTE IF NO CLOCK STOP	IF CLOCK STOP OCCURRED, EXECUTE COMMAND PENDING IN R REGISTER		
			+1 → L REGISTER	+1 → L REGISTER IF NO CLOCK STOP	LOAD NEXT COMMAND INTO R REGISTER		
	CPU EXECUTES PRECEDING COMMAND	+1 → L REGISTER	MEMORY BUSY	MEMORY BUSY			
			MEMORY STOP ENABLE	MEMORY STOP ENABLE	+1 → L REGISTER		
		EXECUTE HALF CYCLE READ OR WRITE COMMAND					
V (FULL CYCLE READ OR WRITE) (OP CODE AX0, AX1, AX4, AX5, AX8, AX9, AXC, AXD)	L REGISTER FETCHES FULL CYCLE READ OR WRITE COMMAND FROM CONTROL MEMORY	LOAD FULL CYCLE READ OR WRITE COMMAND INTO R REGISTER	LOAD NEXT COMMAND INTO R REGISTER (EXECUTE IF NO CLOCK STOP)	LOAD NEXT COMMAND INTO R REGISTER AND EXECUTE IF NO CLOCK STOP	LOAD NEXT COMMAND INTO R REGISTER AND EXECUTE IF NO CLOCK STOP	LOAD NEXT COMMAND INTO R REGISTER AND EXECUTE IF NO CLOCK STOP	IF CLOCK STOP OCCURRED, EXECUTE COMMAND PENDING IN R REGISTER
			+1 → L REGISTER	+1 → L REGISTER IF NO CLOCK STOP	+1 → L REGISTER IF NO CLOCK STOP	+1 → L REGISTER IF NO CLOCK STOP	LOAD NEXT COMMAND INTO R REGISTER
	CPU EXECUTES PRECEDING COMMAND	+1 → L REGISTER	MEMORY BUSY	MEMORY BUSY	MEMORY BUSY	MEMORY BUSY	
			MEMORY STOP ENABLE	MEMORY STOP ENABLE	MEMORY STOP ENABLE	MEMORY STOP ENABLE	+1 → L REGISTER
		EXECUTE FULL CYCLE READ OR WRITE COMMAND					
VI (STANDARD COMMANDS) (OP CODES 1001, 1002, 1004, 1008, 1010, 1040, 1080, 11, 12, 13, 17, 18, 1A, 1B, 2, 3, 7, 8, 9, B, C, D, E, F)	L REGISTER FETCHES TYPE VI COMMAND FROM CONTROL MEMORY	TYPE VI COMMAND → R REGISTER					
		+1 → L REGISTER (FETCH NEXT COMMAND)					
	CPU EXECUTES PRECEDING COMMAND	EXECUTE TYPE VI COMMAND					
	FETCH	EXECUTE					

Figure 2-11. Types of Fetch/Execute Cycles

2.6.1.3 CYCLE TYPE III. This cycle is performed for skip-type commands having op codes 4, 5, and 6. For these commands, a compare or test is made between the literal in the command and the contents of a selected file register. If the compare or test is affirmative (A bus byte is all zero's for op code 4; A bus byte is not all zero's for op code 5; sum of literal and file is greater than 255 for op code 6), execution of the next command is skipped. If the compare or test is not affirmative, the next command is not skipped. Total execution time is 400 nanoseconds if a skip is required, or 200 nanoseconds if no skip is required.

During microstep 1, the Type III command is loaded into the R Register, decoded, and the compare or test is performed. Also, the L Register advances to the skip address.

During microstep 2, the command located at the skip address is loaded into the R Register, and the L Register advances to the address beyond the skip address. If the result of the compare or test performed in microstep 1 was affirmative, the run inhibit/clock stop prevents execution of the command at the skip address. If the result was not affirmative, the command at the skip address is executed.

During microstep 3, the command at the address beyond the skip address is loaded into the R Register, decoded, and executed. Also, the L Register is advanced to access the next command.

2.6.1.4 CYCLE TYPE IV. This cycle is performed for half-cycle memory read or write commands. Execution requires 600 nanoseconds. While the read or write operation is in progress, commands at the addresses following the address that contained the memory command are fetched and executed, provided that no clock stop occurs. A clock stop is initiated at microstep 2 or 3 when any of the following conditions occurs:

a. Another memory read or write command is loaded into the R Register at microstep 2 or 3 (clock stop initiated by memory stop enable).

b. The command loaded into the R Register at microstep 2 or 3 selects the M or N Register (clock stop initiated by memory busy).

c. The command loaded into the R Register at microstep 2 or 3 of a half-cycle read operation selects the T Register as the operand on the B bus (clock stop initiated by memory data not available).

When a clock stop occurs, loading of the next command into the R Register is inhibited, execution of the current command in the R Register is delayed, and the L Register is prevented from advancing to the next address until the clock stop is lifted. When the clock stop terminates, the command that is pending in the R Register is executed, and the L Register advances to the next address. If no clock stop occurs, commands are fetched and executed at microsteps 2 and 3 while the read or write operation is in progress.

During microstep 1, the read or write command is loaded into the R Register and decoded, and the L Register advances to the next address.

During microstep 2, the memory read or write operation commences. Also, the command at the next address is loaded into the R Register. If the command does not initiate a clock stop, the command is executed, and the L Register advances to the next address.

During microstep 3, the memory read or write operation continues. If the command in microstep 2 was executed, the next command is loaded into the R Register, and the L Register advances to the next address. The command is not executed if a clock stop is produced. Execution of the read or write operation terminates at microstep 4. If a clock stop occurred, the pending command in the R Register is executed. Also, the L Register advances to the next address.

2.6.1.5 CYCLE TYPE V. This cycle is identical to the type IV cycle, except that the memory read or write operation extends for two additional microsteps. Execution time is 1 microsecond. The description in Paragraph 2.6.1.4 (cycle type IV) is also applicable to cycle type V. The actions in microsteps 3, 4, and 5 of cycle type V are identical to those in microstep 3 of the type IV cycle, and the actions in microstep 6 are identical to those in microstep 4 of the type IV cycle. A clock stop is initiated in microsteps 2, 3, 4, or 5 when any of the following occurs:

a. Another memory read or write command is loaded into the R Register at microstep 2, 3, 4, or 5 (clock stop initiated by memory stop enable).

b. The command loaded into the R Register at microstep 2, 3, 4, or 5 selects the M or N Register (clock stop initiated by memory busy).

c. The command loaded into the R Register at microstep 2 or 3 of a full-cycle read operation selects the T Register as the operand on the B bus (clock stop initiated by memory data not available).

If a clock stop does not occur, commands are fetched and executed at microsteps 2, 3, 4, and 5 while the read or write operation is in progress.

2.6.1.6 FETCH/EXECUTE CYCLE FOR OPERATE-TYPE COMMANDS THAT SELECT THE L REGISTER. Figure 2-11 classifies all operate-type microcommands (op codes 7 through F) as having type VI fetch/execute cycles. A special case exists when the r field (bits 2 through 0) of an operate-type microcommand contains code 4_8 or 5_8, which causes the L Register to be loaded with a 9-bit jump address (the 3 MSB of the L Register remain unchanged), producing a jump in 1/2K. In this case, a type I fetch/execute cycle is performed in the following sequence:

a. In microstep 1, the code in the r field is decoded, selecting the L Register for entry of a jump address. Also, the operate-type microcommand is executed.

b. In microstep 2, A bus bits 7 through 0 (8 LSB of the jump address produced by the operate-type microcommand) are loaded into L Register stages 7 through 0. Also, the state of R Register bit R00 is loaded into L Register stage 8. L Register bits 9, 10, and 11 remain unchanged. A run inhibit/clock stop inhibits execution of the microcommand loaded into the R Register at microstep 2.

c. In microstep 3, the microcommand at the jump address (entered into the L Register in microstep 2) is stored in the R Register and executed, and the L Register advances to the jump address plus one to fetch the next microcommand.

In effect, an operate-type microcommand with code 4_8 or 5_8 in the r field causes execution of the operate-type microcommand, followed by execution of a jump microcommand. Total execution time for the double command is 400 nanoseconds.

2.6.1.7 CYCLE TYPE VI. This cycle is performed for all standard commands (op codes 1001, 1002, 1004, 1008, 1010, 1040, 1080, 11, 12, 13, 17, 18, 1A, 1B, 2, 3, 7, 8, 9, B, C, D, E, and F). During microstep 0, the command is fetched from the control memory, and the preceding command, stored in the R Register at the microstep preceding microstep 0, is executed. During microstep 1, the command fetched in microstep 0 is executed, and the L Register advances to the next address to fetch the next

command. The execute time for all standard commands is 200 nanoseconds. During each microstep, a command is executed, and the next command is fetched from control memory.

2.6.2 BASIC CLOCK TIMING (Figure 2-12)

The basic timing source for all clocks is a 20-MHz, crystal-controlled, square-wave. Clock phase 1 (CPH1) is derived by dividing the 20-MHz clock by four. Clock phase 2 (CPH2/) is derived by delaying clock phase 1 by 33 nanoseconds and inverting. Once every 200 nanoseconds, when clock phases 1 and 2 are both high, clock SCK6 is produced. Signal SCK6 is a 33-nanosecond-wide, 5-MHz clock.

File register clock FLCK/ is a symmetrical 5-MHz square wave that leads CPH1 by 90 degrees. If clock stop signal CSTP/ is not low, the 50-nanosecond-wide M-N register clock (MNCK/) is produced on the positive-going edge of each FLCK/ pulse.

Memory clock CKS1/ is the same as clock phase 1. Memory clock CKS1 is the same as clock phase 2. During the 33-nanosecond-wide interval when both CKS1/ and CKS1 are high, a clock is produced (if enabled) by the memory control.

Clock 6 (CLK6/) is the inverted phase of 5-MHz clock SCK6. Data clock 1 (CLK1) is identical to SCK6, however, CLK1 is inhibited when CSTP/ = 0 (clock stop) exists.

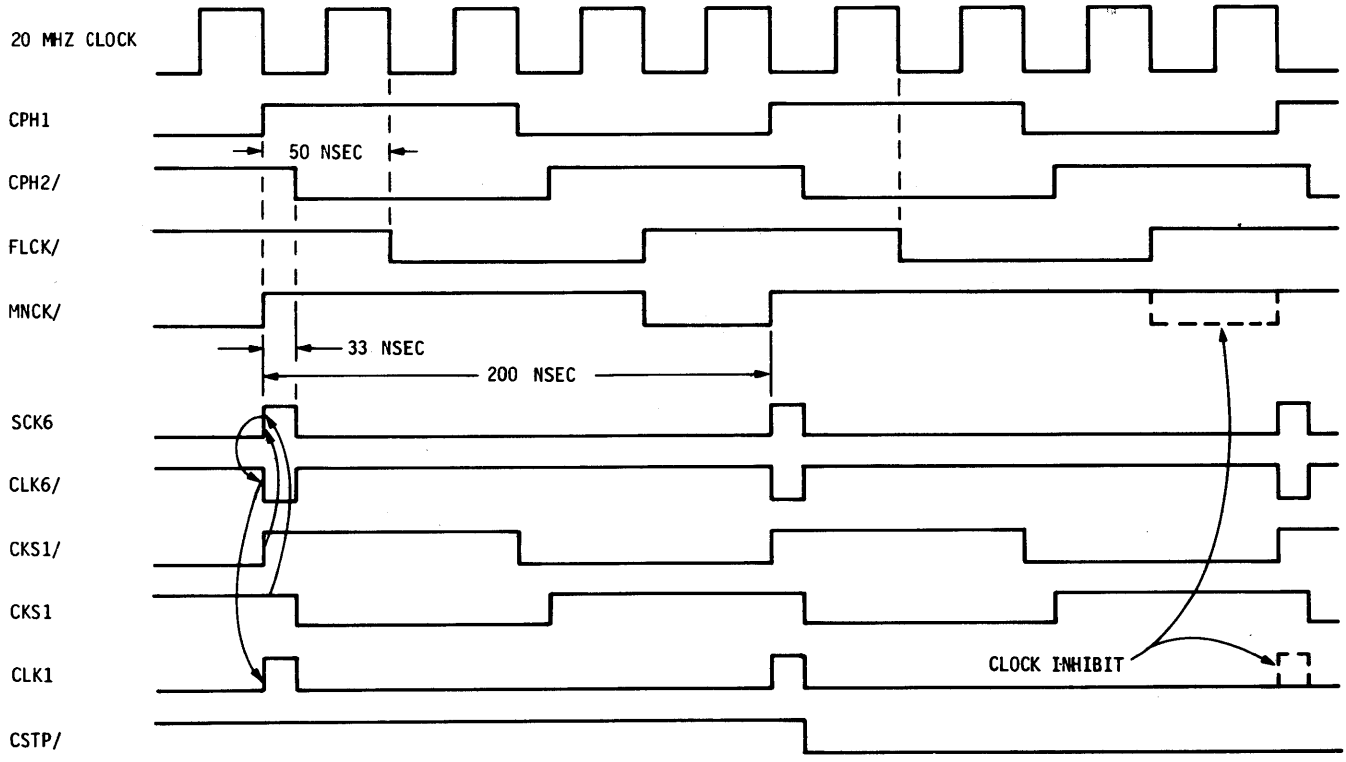


Figure 2-12. Basic Clock Timing

2.6.3 RUN/HALT TIMING (Figure 2-13)

The run/halt circuit in the run and inhibit control initiates a clock stop when a Halt command (op code 1780) is executed (run halt signal RUNH/ = 0), or when either the CLOCK switch or the address stop logic (in the Front Panel Control Circuits) produces CLKF/ = 0. When CLKF/ = 0 exists, run reset occurs (RUNR/ = 0), provided that the microcommand in the R Register is not a jump, or a skip-type microcommand for which a skip is taken (INLR = 1). If a skip or jump is in progress, INLR = 0 delays the occurrence of RUNR/ = 0 for one clock time. When either RUNH/ = 0 or RUNR/ = 0 occurs, RUNX goes low at the next clock to produce a clock stop (CSTP/ = 0). RUNX remains low until the clock following the occurrence of a run set (RUNS/ = 0). Signal RUNS/ goes low for one clock period when the RUN, STEP, CLOCK, or INT switch is activated, or when a power coming up condition exists in the power fail circuit (RUNF/ = 0). As shown in Figure 2-15, operation of the CLOCK switch while the CPU is in a HALT state causes the CPU to shift to the RUN state for one clock interval (two clock intervals for a jump microcommand, or for a skip-type microcommand when the skip is taken).

2.6.4 RUN INHIBIT TIMING (Figure 2-14)

The run inhibit circuit in the run inhibit control initiates a clock stop when a jump, return jump, or skip-type command (fetch/execute cycle type I, II, or III) is being executed. Signal RINK goes high during microstep 1 of atype I cycle, during microstep 1 of a type III cycle when a skip is performed, and during microstep 2 of a type II cycle. Then, on the trailing edge of the next clock, run inhibit RINH/ goes low for one clock interval, producing a clock stop (CSTP/ = 0).

2.6.5 PANEL INTERRUPT TIMING (Figure 2-15)

The panel interrupt circuit in the run inhibit control initiates a panel interrupt when the front panel INT switch is activated. When the INT switch is activated, signal INTF/ = 0 produces INTCK/ = 0 at the next clock. One clock later, panel interrupt INTC/ = 0 is produced, and continues until an enter internal status command (op code 7X4X) is executed, producing AENI = 1. One clock after AENI = 1 occurs, INTC/ is reset.

2.6.6 CLOCK STOP AND CLOCK 7, 8, AND 9 TIMING (Figure 2-16)

The clock stop signal (CSTP/) is controlled by the RUNX, RINH/, and memory stop conditions. (Refer to Paragraphs 2.6.1.4 and 2.6.1.5 for a description of memory

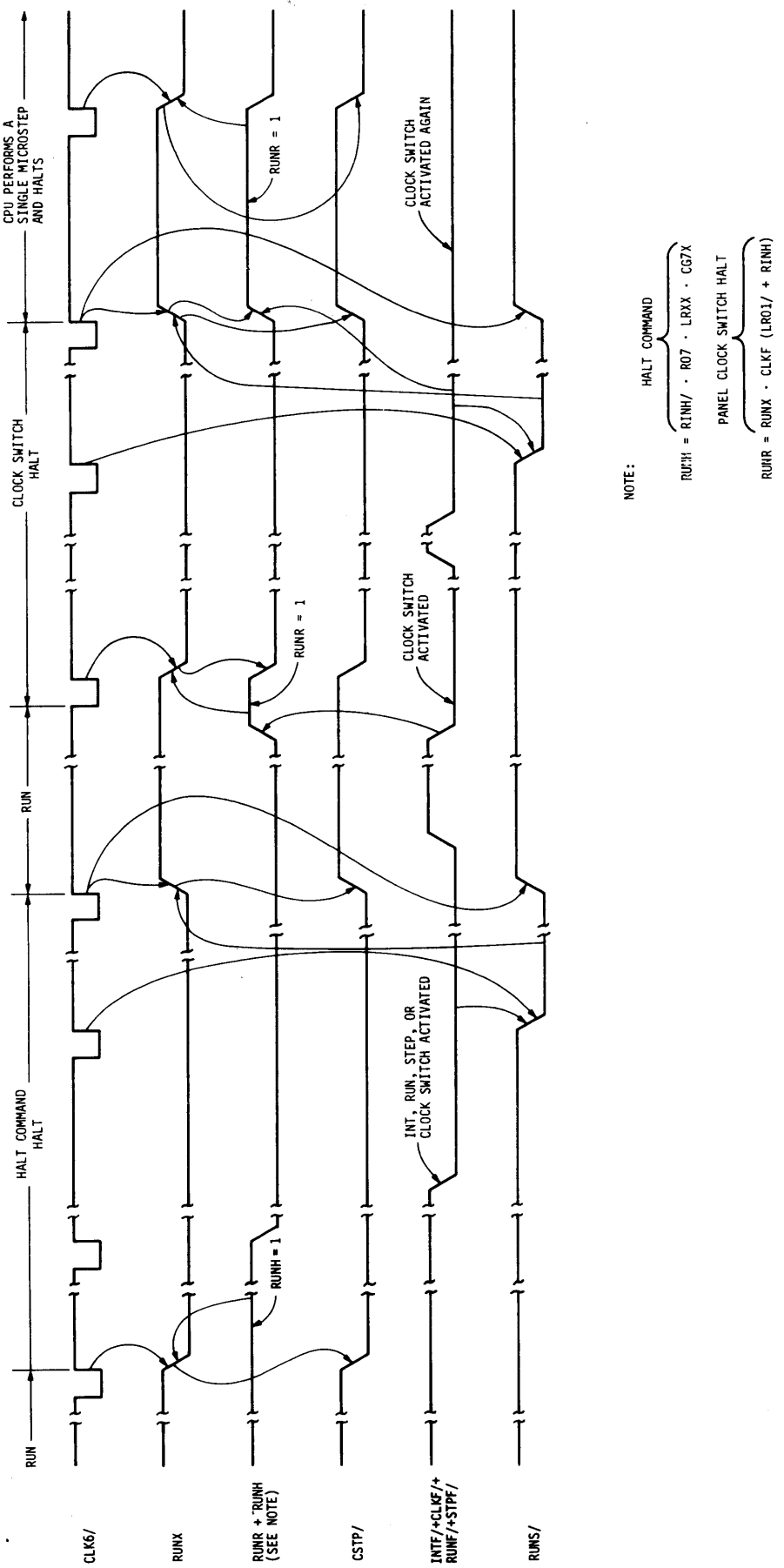


Figure 2-13. Run/Halt Timing

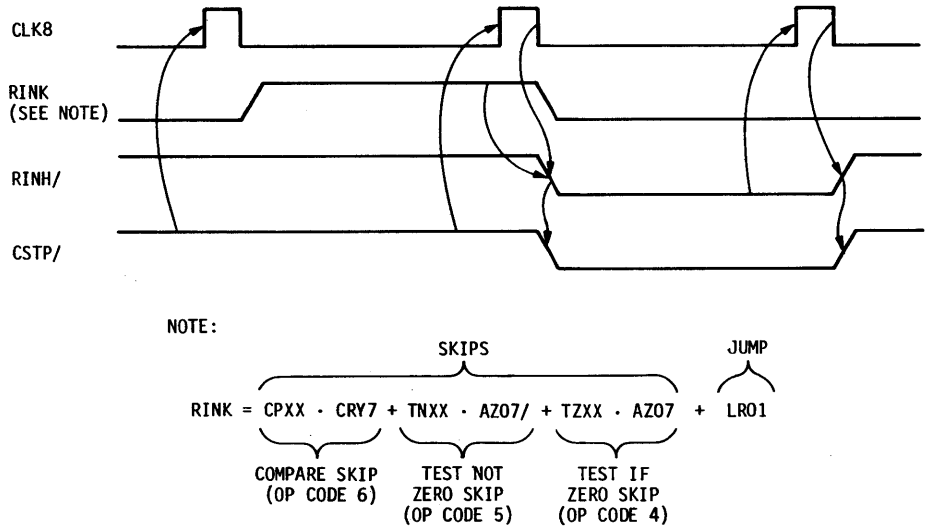


Figure 2-14. Run Inhibit Timing

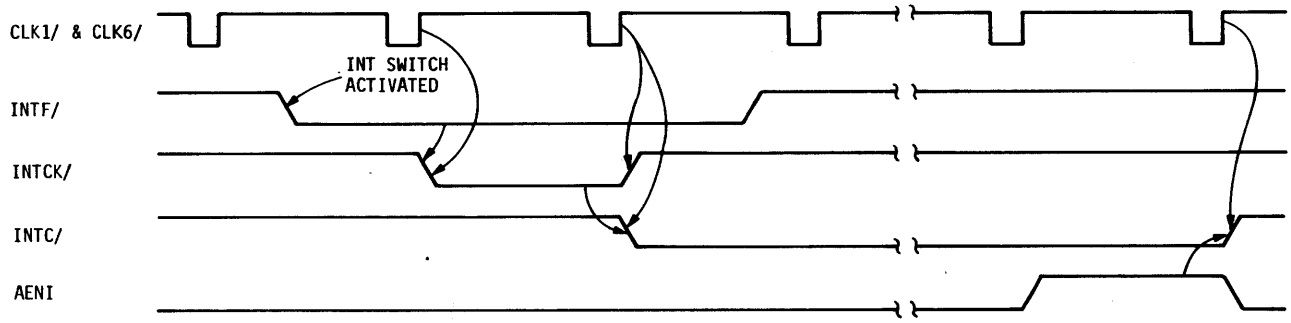


Figure 2-15. Panel Interrupt Timing

stop conditions.) When RUNX = 1, RINH/ = 0, or a memory stop occurs, CSTP/ = 0 is produced, inhibiting the clocks controlled by CSTP/.

Clocks 7 and 9 (CLK7/ and CLK9/) are used to load the upper (CLK7/) and lower (CLK9/) R Registers. Clocks 7 and 9 are produced at each occurrence of 5-MHz clock SCK6 (refer to Paragraph 2.6.2), provided that at least one of the following exists:

a. Run inhibit RINH/ is low (run inhibit in progress).
b. Clock stop CSTP is low (no clock stop in progress).
c. Run control RUNX is low (Halt in progress).

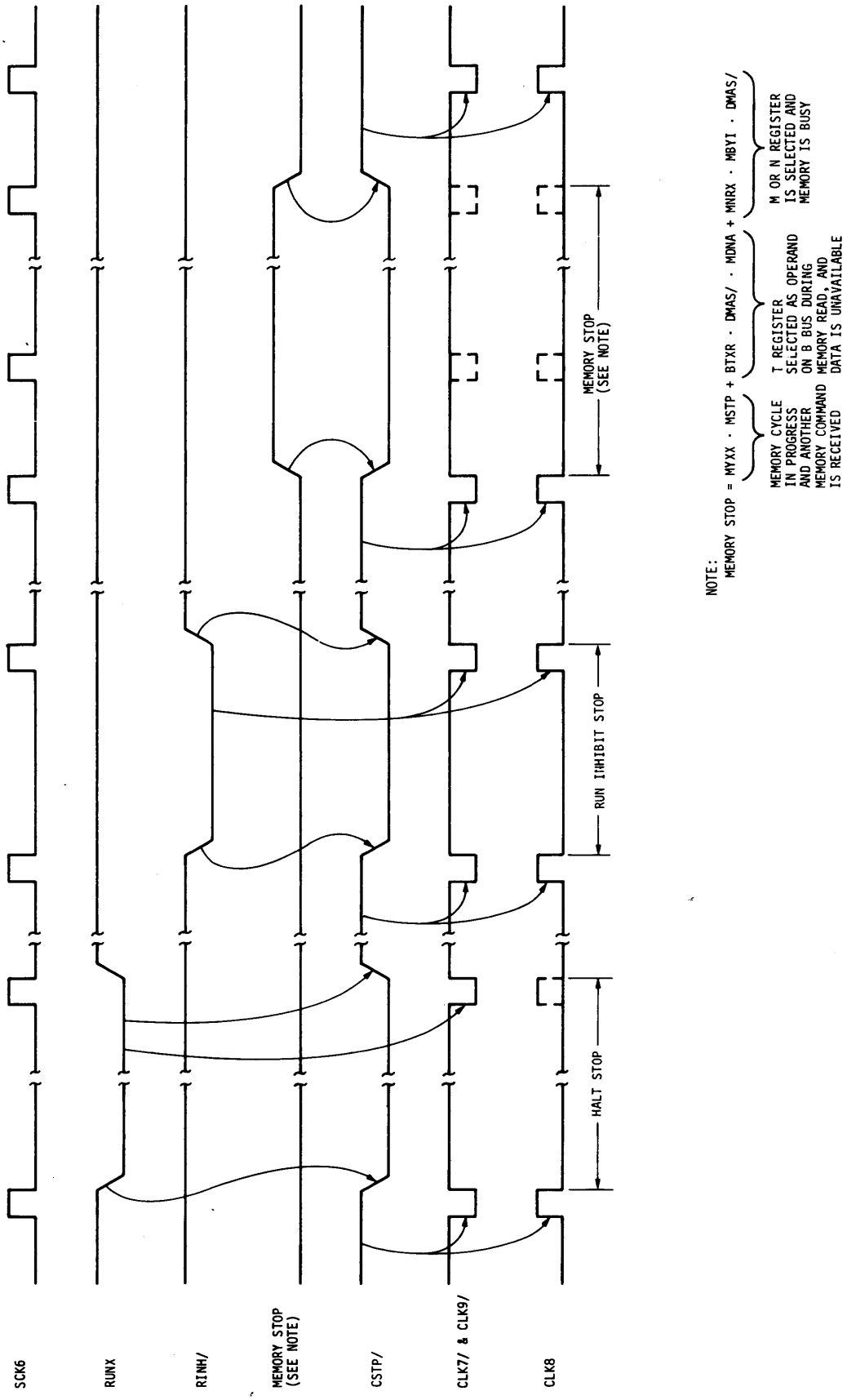


Figure 2-16. Timing for Clock Stop and Clocks 7, 8, and 9

Consequently, clocks 7 and 9 are inhibited only when a clock stop has been initiated by a memory stop.

Clock 8 (CLK8) is produced at each occurrence of 5-MHz clock SCK6, provided that either CSTP/ = 1 or RINH/ = 0 exists. Consequently, clock 8 is inhibited only when a clock stop has been initiated by a halt (RUNX = 0), or by a memory stop. Clock 8 is used in the run inhibit control to clock the run inhibit and switch sync flip-flops.

2.6.7 CLOCK 2 TIMING (Figure 2-17)

Clock 2 (CLK2/) is used to clock the File Registers. Clock 2 is inhibited whenever CSTP/ = 0 (clock stop) exists. If CSTP/ = 1 exists, clock 2 is produced at the 5-MHz clock (SCK6) following the occurrence of any of the following command decodes:

a. LFXX = 1 (load file with literal - op code 2)

b. AFXX = 1 (add file with literal - op code 3)

c. Bit 3 (R03) of the microcommand is a 0 (not file inhibit), for op codes 7, 8, 9, A, B, C, D, E, and F (BENR/ = 1)

2.6.8 M REGISTER CLEAR CLOCK TIMING (Figure 2-18)

The M Register clear (MCLR/) clock is used to clear the M Register. Clock MCLR/ is inhibited whenever CSTP/ = 0 (clock stop) exists. If CSTP/ = 1, MCLR/ is produced at the 5-MHz clock (SCK6) produced during the execution of a Load N (op code 13) command (LRXX and NRXX both true).

2.6.9 T REGISTER LOAD CLOCK TIMING (Figure 2-19)

The T Register load clock (TCLK/) is used for a synchronous load of the T Register from the A bus. Clock TCLK/ is inhibited whenever CSTP/ = 0 (clock stop) exists. If CSTP/ = 1, TCLK/ is produced at the 5-MHz clock (SCK6) following the occurrence of a load T Register command decode (TRTX = 1 or TRXX = 1). Command decode TRTX = 1 is produced when a return, load T command (op code 19) is being executed. Command decode TRXX = 1 is produced when a load T command (op code 11) is being executed, or when the r field selects the T Register during the execution of a command with op code 7, 8, 9, A, B, C, D, E, or F.

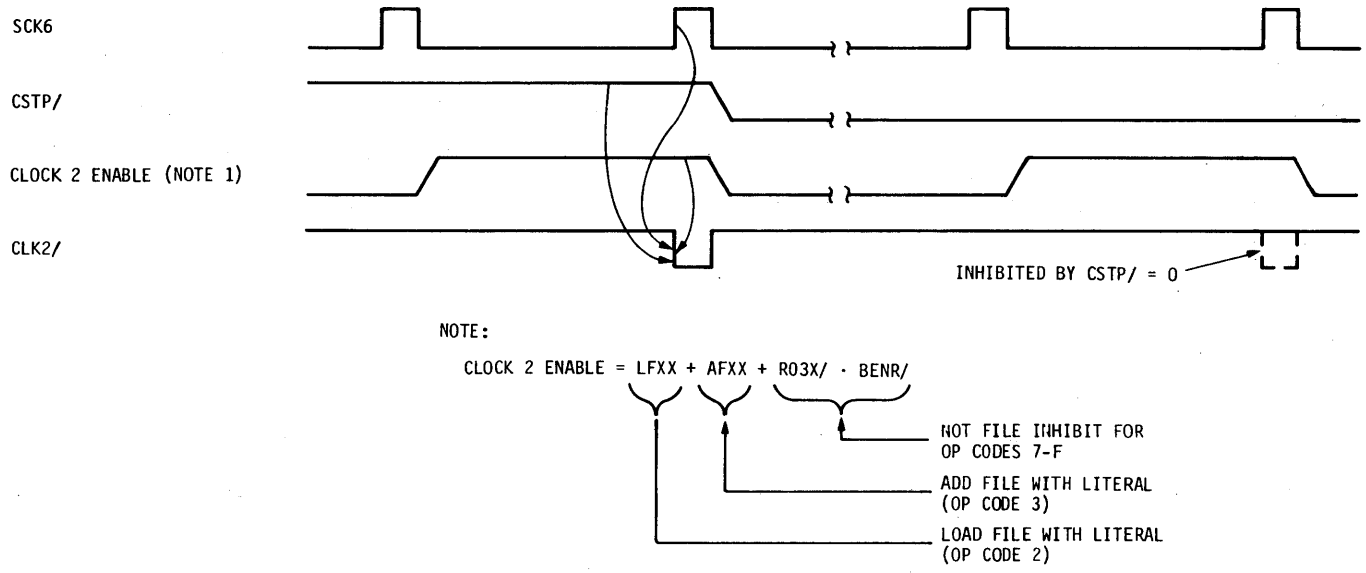


Figure 2-17. Clock 2 (File Register Write Clock) Timing

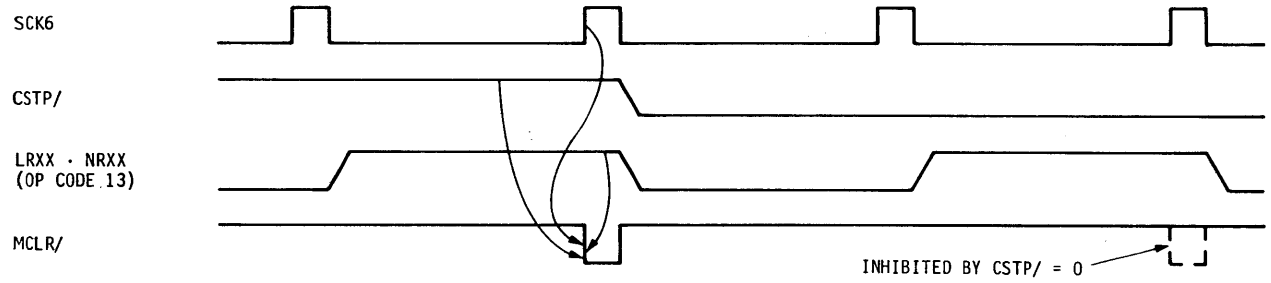


Figure 2-18. MCLR/ (M Register Clear) Timing

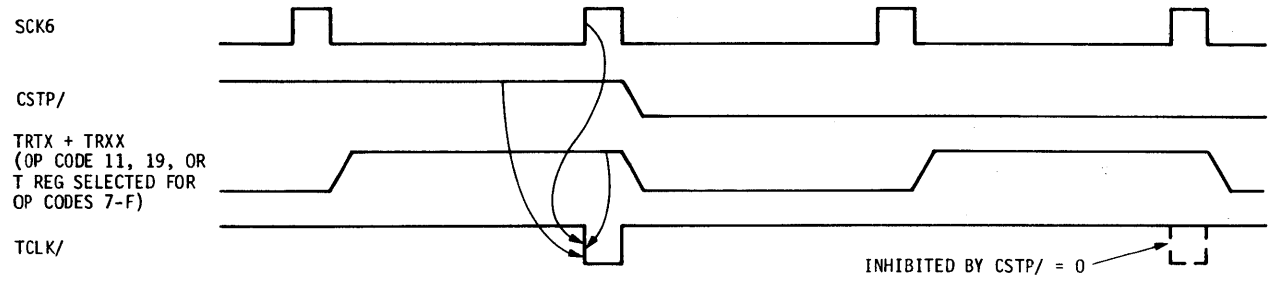


Figure 2-19. TCLK/ (T Register Load) Timing

2.6.10 L REGISTER TIMING (Figure 2-20)

The operation of the L Register depends upon the type of fetch/execute cycle being performed. When CNT ENABLE = 1 (count enable) and LRLL/ = 1 (not L Register load) both exist, the L Register functions as a straight modulo-4096 binary counter that advances one count on the trailing edge of each 5-MHz clock (SCK6). When CNT ENABLE = 0 (not count enable) and LRLL/ = 0 (not L register load) both exist, the L Register stops counting, and holds the current count value (address). When LRLL/ = 0 (L register load) and JPXX/ = 0 (execute jump extended or return jump command) both exist, a jump (jump extended) or L Save (return) address is loaded into the L Register. The eight low-order address bits are received (via the B bus, the arithmetic/logic unit, and the A bus) from the lower R Register. The four high-order address bits are received directly from the upper R Register.

When LRLL/ = 0 and JPXX/ = 1 both exist, a jump in 1K command (op code 14, 15, 1C, or 1D) or jump in 1/2K (op code 7 through F with code 4_8 or 5_8 in r field) is being executed. The eight low-order address bits (bits 0 through 7) are received from the A bus. Bit 8 is received from stage R08 of the upper R Register. For a jump in 1K, bit 9 is received from stage R11 of the upper R Register and bits 10 and 11 remain unchanged. For a jump in 1/2K, bits 9 through 11 remain unchanged.

The various modes and related timing signals for the L Register are shown in Figure 2-20. This illustration depicts a typical sequence that involves all of the various L Register modes. First, a jump extended (op code 0) command is executed, causing the L Register to store the jump address contained in the command. Following the execution of the jump extended load, the L Register resumes counting. Then, a return command is executed (RTNX = 1), stopping the counting action in the L Register. The occurrence of CPEN = 0 inhibits the control memory. During microstep 2 of the return command execution, JPXX/ = 0 causes the L Save address to be loaded into the L Register. Following the execution of the return command, the L Register resumes counting. Next, a jump in 1K command is executed, causing the L Register to store the jump in 1K address contained in the command. After execution of the jump in 1K command, the L Register again resumes counting.

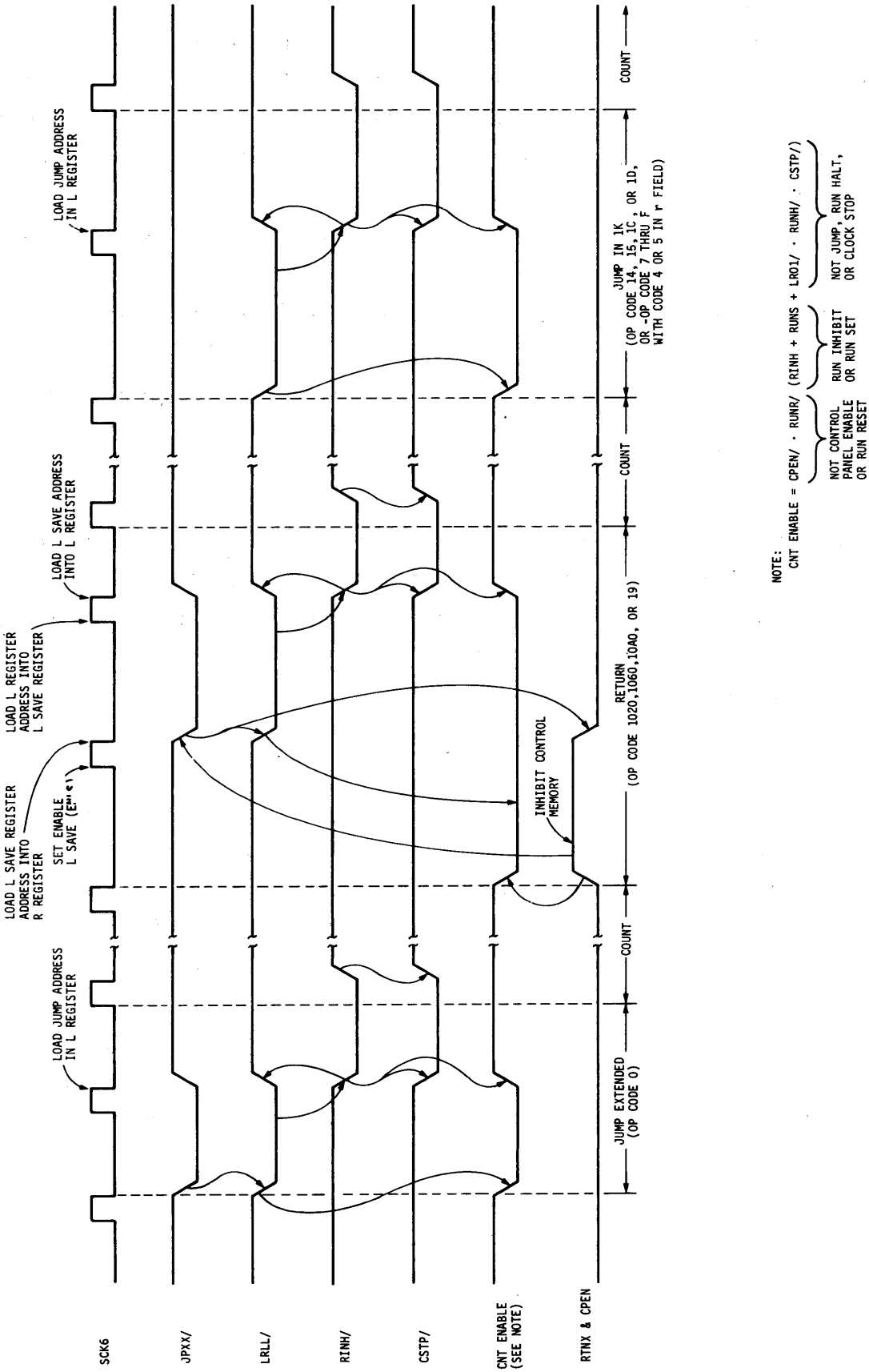


Figure 2-20. L Register Timing

2.6.11 CORE MEMORY READ/WRITE TIMING (Figures 2-21 and 2-22)
Timing during core memory read/write cycles is controlled by the memory control, in conjunction with the Machine State Control and Clock Circuits. Four types of cycles can be initiated by the CPU, depending upon the microcommand:

Microcommand Op Code	Type of Memory Cycle	Execution Time	Type of Fetch/ Execute Cycle
AX0, AX4, AX8, AXC	Full-Cycle Memory Read	1 microsecond	V
AX2, AX6, AXA, AXE	Half-Cycle Memory Read	600 nanoseconds	IV
AX1, AX5, AX9, AXD	Full-Cycle Memory Write	1 microsecond	V
AX3, AX7, AXB, AXF	Half-Cycle Memory Write	600 nanoseconds	IV

It should also be noted that different types of fetch/execute cycles are required for full-cycle and half-cycle memory operations (see Figure 2-11).

During a half-cycle operation, only a memory read (half-cycle memory read) or a memory write (half-cycle memory write) action is performed. During a full-cycle read or write operation, a read operation followed by a write operation is performed. Bit R04 of the command in the R Register determines whether a read (bit 4 = 0) or write (bit 4 = 1) is being executed. Bit R05 of the command in the R Register determines whether a half-cycle (bit 5 = 1) or a full-cycle (bit 5 = 0) operation is being executed. Memory read or write operations cannot be initiated when a clock stop exists (CSTP/ = 0). Timing differences for memory cycles initiated by the DMA channel are discussed in Paragraph 2.6.11.3.

2.6.11.1 MEMORY WRITE CYCLE TIMING (Figure 2-21). Figure 2-21 shows the timing for both full- and half-cycle memory write operations. The full- and half-cycle operations are discussed separately in the following subparagraphs.

2.6.11.1.1 Half-Cycle Memory Write. At clock time t_0, the half-cycle write command is decoded, producing MYCS/ = 0 (memory cycle start), MRCY/ = 1 (not read cycle), and MFCY/ = 1 (not full-cycle). Then, at clock time t_1, the Write flip-flop (WRITE FF) is set, producing MSTP = 1 (memory stop), MBSY = 1 and MBYI = 1 (memory busy) and MTSV = 1 (inhibit memory data register load). The occurrence of MTSV = 1 produces 350-nanosecond memory write pulse WTXX/. Write pulse WTXX/ terminates between clocks t_2 and t_3. At clock time t_3, the WRITE FF is reset.

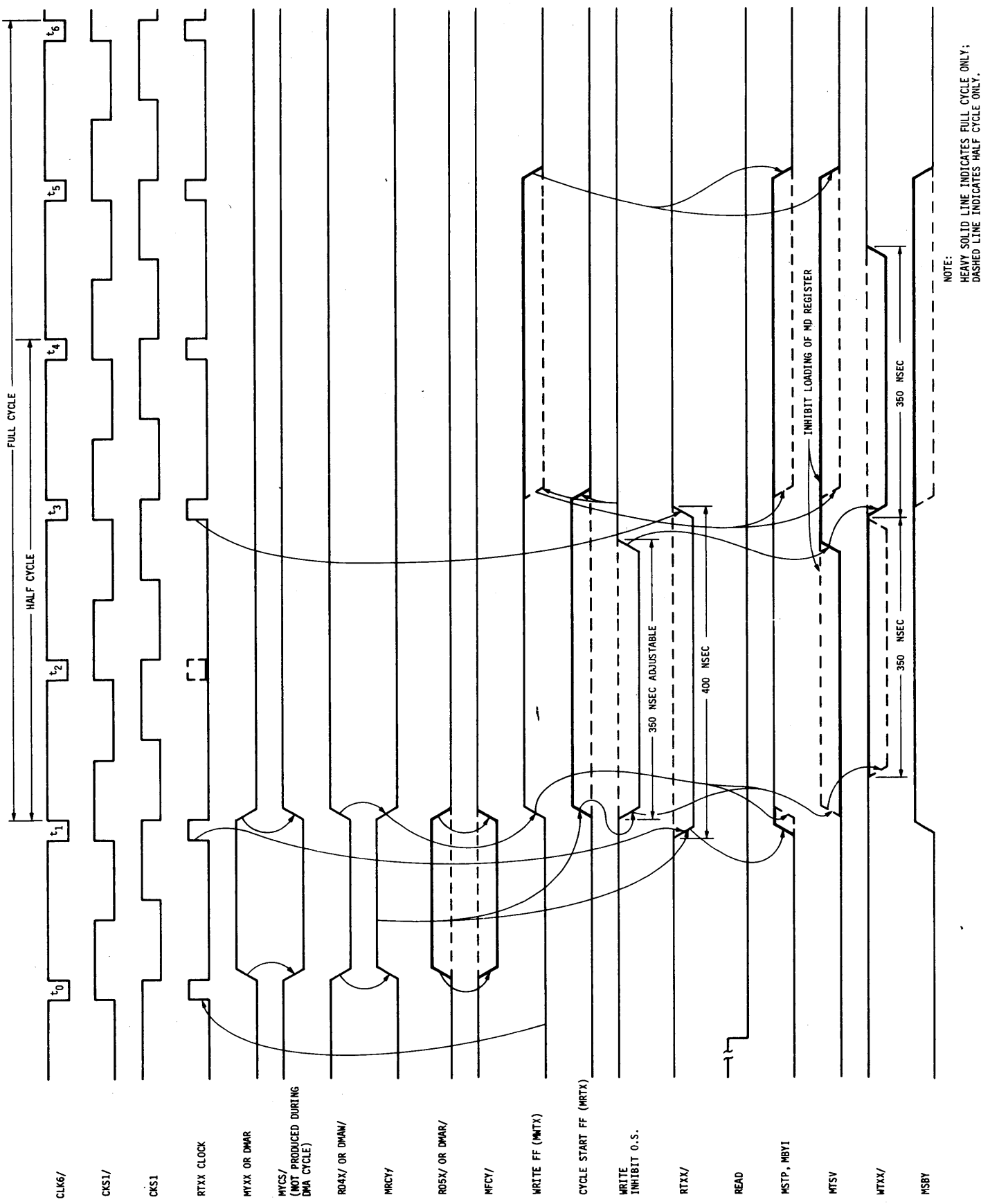


Figure 2-21. Memory Write Cycle Timing

2.6.11.1.2 Full-Cycle Memory Write. At clock time t_0, the full-cycle write command is decoded, producing MYCS/ = 0 (memory cycle start), MRCY/ = 1 (not read cycle), and MFCY/ = 0 (full cycle). Then, at clock time t_1, the Write (WRITE FF) and Read (READ FF) flip-flops are both set, producing MSTP = 1 (memory stop), MBSY = 1 and MBYI = 1 (memory busy), and RTXX/ = 0 (memory read pulse). The occurrence of READ FF = 1 triggers the Write Inhibit One-Shot (WRITE INHIBIT O.S.), producing an adjustable 350-nanosecond negative pulse. The 350-nanosecond pulse inhibits the RTXX clock at clock time t_2, enabling the RTXX/ pulse to remain low until clock time t_3. The WRITE INHIBIT O.S. pulse terminates between clocks t_2 and t_3. On the trailing edge, MTSV = 1 occurs (inhibit memory data register load), producing 350-nanosecond memory write pulse WTXX/. Write pulse WTXX/ terminates between clocks t_4 and t_5. At clock time t_5, WTXX/ = 1 resets WRITE FF.

2.6.11.2 MEMORY READ CYCLE TIMING (Figure 2-22). Figure 2-22 shows the timing for both full- and half-cycle memory read operations. The full- and half-cycle operations are discussed separately in the following subparagraphs.

2.6.11.2.1 Half-Cycle Memory Read. At clock time t_0, the half-cycle memory read command is decoded, producing MYCS/ = 0 (memory cycle start), MRCY/ = 0 (memory read cycle), and MFCY/ = 1 (not full-cycle). Then, at clock time t_1, the TCLR/ clock is produced, clearing the T Register in preparation for the memory read data load at clock time t_3. Also, at clock time t_1, the READ FF is set, memory read pulse RTXX/ is produced, and MSTP = 1 (memory stop), MBSY = 1 and MBYI = 1 (memory busy), MDNA = 1 (memory data not available), READ = 1 (memory read enable), and TDBC/ = 1 (inhibit memory data register output to MD bus) all occur. When the READ FF sets, the Write Inhibit One-Shot (WRITE INHIBIT O.S.) is triggered, producing an adjustable 350-nanosecond negative pulse. The 350-nanosecond pulse inhibits the RTXX clock at clock time t_2, permitting the RTXX/ pulse to continue until clock time t_3.

At clock time t_2 MDRC = 1 (memory data read control) occurs, producing the TMCK/ clock at clock time t_3. The occurrence of TMCK/ gates and read data from the memory data bus to the T Register. The WRITE INHIBIT O.S. pulse terminates between clocks t_2 and t_4. Finally, at clock time t_3, the READ FF is reset, RTXX/ = 1 occurs, and the half-cycle read operation terminates. At clock time t_4, MDRC = 0, TDBC/ = 0, and READ = 0 occur.

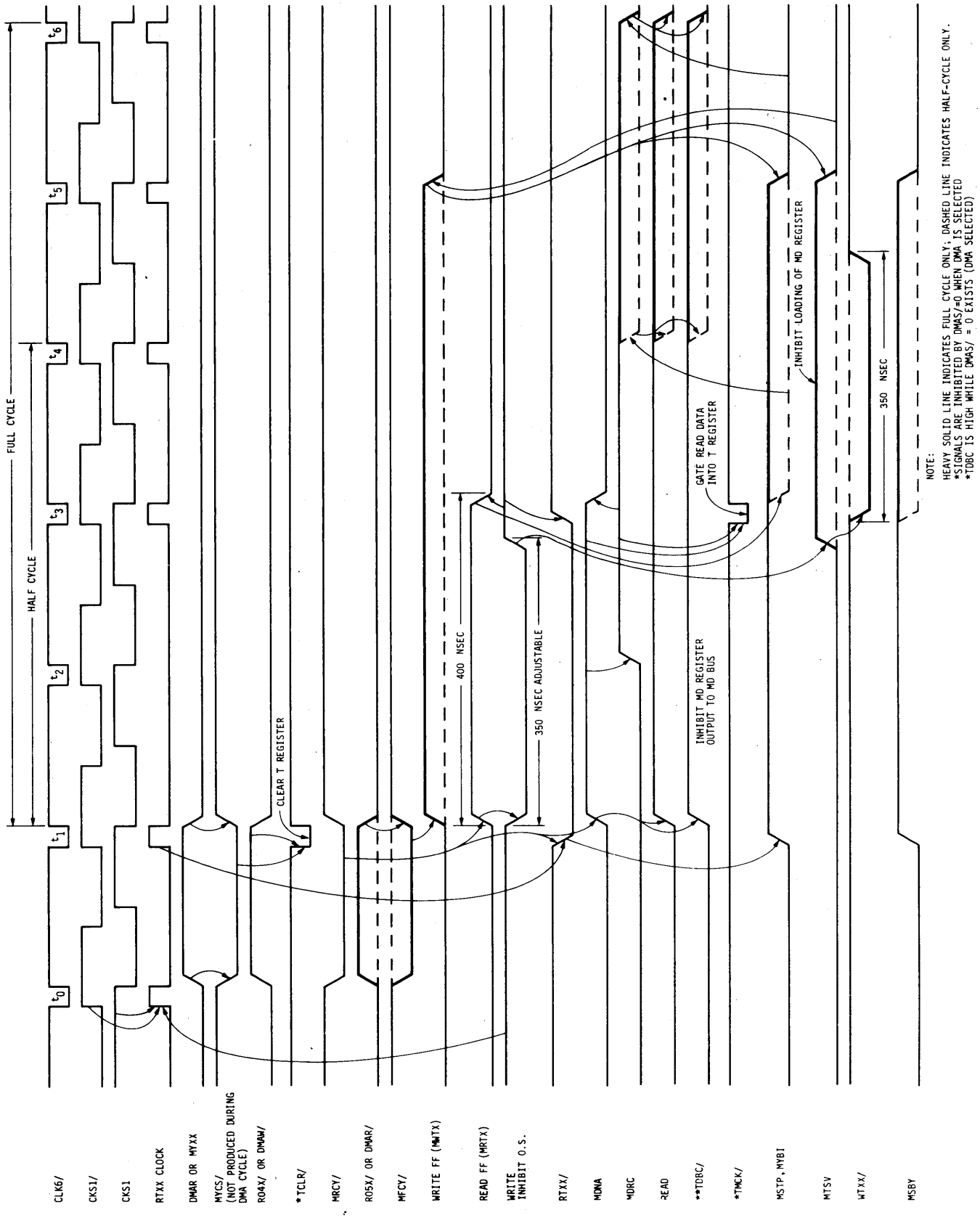


Figure 2-22. Memory Read Cycle Timing

2.6.11.2.2 Full-Cycle Memory Read. At clock time t_0, the full-cycle memory read command is decoded, producing MYCS/ = 0 (memory cycle start), MRCY/ = 0 (memory read cycle), and MFCY/ = 0 (full-cycle). Then, at clock time t_1, the TCLR/ clock is produced, clearing the T Register in preparation for the memory read data load at clock time t_3. Also at clock time t_1, the READ FF and WRITE FF are both set, memory read pulse RTXX/ is produced, and MSTP = 1 (memory stop), MBSY = 1 and MBYI = 1 (memory busy), MDNA = 1 (memory data not available), READ = 1 (memory read enable), and TDBC/ = 1 (inhibit memory data register output to MD bus) all occur. When the READ FF sets, the Write Inhibit One-Shot (WRITE INHIBIT O.S.) is triggered, producing a 350-nanosecond negative pulse. The 350-nanosecond pulse inhibits the RTXX clock at clock time t_2, permitting the RTXX/ pulse to continue until clock time t_3.

At clock time t_2, MDRC = 1 (memory data read control) occurs, producing the TMCK/ clock at clock time t_3. The occurrence of TMCK/ gates the read data from the memory data bus to the T Register.

The WRITE INHIBIT O.S. pulse terminates between clocks t_2 and t_3. When the pulse termines, MTSV = 1 (inhibit loading of MD Register) occurs, producing 350-nanosecond write pulse WTXX/. At clock time t_3, the READ FF is reset, the RTXX/ pulse terminates, and MDNA = 0 occurs.

Write pulse WTXX/ terminates between clocks t_4 and t_5. At clock time t_5 the WRITE FF is reset, terminating the full-cycle memory read operation. Finally, at clock time t_6, MDRC = 0, TDBC/ = 0, and READ = 0 occur.

2.6.11.3 MODIFIED TIMING FOR DMA MEMORY CYCLES. The DMA channel has priority over the CPU in accessing core memory. However, the DMA can only initiate full-cycle read or write operations. The memory read or write operation is initiated by DMAR/ = 0 (DMA request) instead of by MYXX = 1. DMA requests are inhibited while the memory is busy (MBSY = 1). When a DMA request is being serviced (DMAR/ = 0), signal DMAW/ replaces R04X/ as the read or write operation select control, and MYCS/ = 0 is not produced. When DMAW/ = 0 exists, a full-cycle write operation is performed. When DMAW/ = 1 exists, a full-cycle read operation is performed. Other than these changes, timing for the full-cycle write operation is identical to that described in Paragraph 2.6.11.1.2.

For a full-cycle read operation, timing is identical to that described in Paragraph 2.6.11.2.2, with the changes just described, and following exceptions:

a. The occurrence of DMAS/ = 0 (DMA selected) inhibits the generation of a TCLR/ clock at clock time t_1 (no T Register clear).

b. The occurrence of DMAS/ = 0 inhibits the generation of a TMCK/ clock at clock time t_3 (no memory read data loaded into T Register).

c. The occurrence of DMAS/ = 0 forces TDBC/ to the 1 state for the duration of the DMA memory read cycle (output from MD Register is inhibited).

During a DMA memory cycle, a clock stop is initiated only when a memory read or write command is in the R Register.

2.6.12 MASTER RESET

The master-reset pulse (MRST/ = 0) is produced when the front panel RESET button is activated, and also occurs during the power-fail sequence, immediately following a CPU HALT. When the master-reset pulse occurs, the following actions are produced in the CPU:

a. The following registers are cleared:

   1. L Register (control memory location 000 is addressed)

   2. IC Register

   3. U Register

b. The following interrupts are cleared:

   1. Panel Interrupt (INTC/)

   2. Real Time clock interrupt (RTCI/). Also, the real time clock is disabled.

   3. External Interrupt (INTR/). Also, the external interrupt is disabled.

   4. Power Fail Interrupt (PWRF/).

c. All control flip-flops in the memory control are reset.

d. All machine state control flip-flops (RUNX, RINH, SS1, and SS2) in the run and inhibit control are reset. The CPU enters the HALT state, and a clock stop occurs (CSTP/ = 0).

e. The following registers are enabled:

1. L Save Register

2. Primary file registers

f. The address stop logic is reset.

2.7 LOGIC DESCRIPTIONS

Figures 2-3 through 2-10 identify the logic diagram figure number(s) for the elements mentioned in each logic description. These logic diagrams (four-digit numbers) are also referenced in the text.

2.7.1 FRONT PANEL CONTROL CIRCUITS (Figure 2-3)

2.7.1.1 CONTROL SWITCHES AND CONTROL DISPLAY. When key-lock switch S2 is in the ON position, CONE/ = 0 exists, enabling the RUN (S17), STEP (S18), INT (S19), CLOCK (S20), and RESET (S21) machine state control switches (logic diagrams 0110, 0111, and 0112). Associated with the RUN, STEP, INT, and CLOCK switches are individual latches that are set when the switch is activated and reset when the switch is deactivated. When none of the four switches is activated, all four latches are reset. Paragraph 2.6.13 describes the actions produced when the RESET switch is activated.

Activation of PANEL switch S22 when CONE/ = 0, produces CPSE/ = 0, CPSE = 1, and CPEN/ = 0. The occurrence of CPSE/ = 0 causes the PANEL indicator (DS22) to light, and enables the upper and lower data switches. The occurrence of CPEN/ = 0 forces all the control memory outputs to the deactivated high level condition. CONE/ = 0 also enables the PANEL switch.

When key-lock switch S2 is in the LOCK position, LOCK = 1 occurs, and LOCK indicator DS21 is lit, indicating that the front panel is disabled.

Signal RUNX from the Machine State Control and Clock Circuits controls the RUN and HALT indicators. When RUNX = 1 exists, RUN indicator DS19 is lit. When RUNX = 0 exists, HALT indicator DS20 is lit.

Sense switches S23 through S26 are always enabled. Sense switch outputs ES04 through ES07 are supplied to the A bus multiplexer in the Processor Section.

2.7.1.2 UPPER DATA SWITCHES AND ADST/SCAN DISPLAY. The states of data switches 15 through 8 (S1 through S8) are gated onto the RS bus when CPSE/ = 0 exists (logic diagram 0130). The RS bus supplies the switch signals to the R Register in the Control Unit, and to the Front Panel Display Indicators. The output from data switch CR15 (S1) is supplied to the address stop logic to enable the ADDRESS STOP mode. Outputs CR11 through CR08 from data switches S2 through S8 are supplied to the L address comparator.

The ASSW latch output will be high (1) when data switch 15 (S1) is activated (CR15 = 1), and LOCK = 0 also exists. If CPSE/ = 1 (not panel mode) and ASSW = 1, the ADST indicator will light, indicating the Address Stop mode has been selected. The activation of data switch 14 (S2) with CPSE/ = 1 present, produces SCAN/ = 0 and lights the SCAN indicator to indicate the SCAN mode has been selected.

2.7.1.3 LOWER DATA SWITCHES. The occurence of CPSE/ = 0 (panel mode selected) or AENP/ = 0 (Enable Console Switches microcommand being executed) gates the states of data switches 7 through 0 (S9 through S16) (logic diagram 0120) onto the RS bus. The RS bus supplies the switch signals to the R Register in the Control Unit, and to the Front Panel Display Indicators. Outputs CR07 through CR00 from data switches S9 through S16 are supplied to the L Address Comparator.

2.7.1.4 L ADDRESS COMPARATOR. The L Address Comparator (logic diagram 0140) consists of three 4-bit comparators that are chained together to form a 12-bit address match detector. A match is detected when the states of signals CR11 through CR00 from the upper and lower data switches match the states of bits L11 through L00 from the L Register in the Control Unit. When a match is detected, signal LSYC goes high for one clock interval. Signal LSYC is supplied to the address stop logic, and is made available for monitoring at test point LSYC of the front panel board.

2.7.1.5 ADDRESS STOP LOGIC. The address stop logic (Figure 2-25 and logic diagram 0150) is only enabled in the ADDRESS STOP mode, in which a CPU HALT is initiated at a stopping address selected by the front panel data switches. When the ADDRESS STOP mode is selected, the following conditions must exist to select the ADDRESS STOP mode:

a. LOCK = 0 is present, indicating that the front panel is not locked out.

b. CPSE/ = 1 is present, indicating that the PANEL mode is not selected.

c. CR15 = 1 exists, indicating that data switch 15 is activated to select the ADDRESS STOP mode.

When all three of these conditions are present, ASSW = 1 occurs, and the ADST indicator lights. Also, clock CPH2/ is applied to the two address stop flip-flops (designated FF1 and FF2 in Figure 2-25). Flip-flop FF1 is clocked on the negative-going leading edge of CPH2/, and FF2 is clocked on the positive-going trailing edge of CPH2/.

As shown in Figure 2-25, flip-flops FF1 and FF2 are normally in the reset state. An address stop (CLKF/ = 0) occurs when either flip-flop is set, provided that RUNA = 1 exists. Signal RUNA = 1 exists when the CPU is in the RUN state, provided that the R Register does not contain a return jump microcommand.

If the CPU is in the RUN state, an address stop sequence is initiated when LSYC = 1 occurs, indicating the stopping address has been reached. With LSYC = 1, flip-flop FF1 is set at the next CPH2/ leading edge, provided that both of the following conditions exist:

a. INLR = 1 exists. When INLR = 0 exists at the clock following the occurrence of LSYC = 1, no address stop is initiated. This condition occurs when the stopping address is the next address beyond a location containing a jump microcommand, or the next address beyond a location containing a skip-type microcommand for which a skip is taken.

b. RUNA = 1 exists. When RUNA = 0 exists at the clock following the occurrence of LSYC = 1, no address stop is initiated. This condition occurs when the stopping address is the next address beyond a location containing a return jump microcommand (op code 1020, 1060, 10A0, or 19).

The four different address stop cases shown in Figures 2-23, 2-24, and 2-25 are described separately in the following subparagraphs.

2.7.1.5.1 Case 1 (Figure 2-23). Case 1 is the general case, in which the stopping address does not contain a jump, return jump, or skip-type microcommand. Flip-flop FF1 sets on the next high to low edge of CPH2/ following the occurrence of LSYC = 1, and remains set for one clock interval. Flip-flop FF2 sets on the next CPH2/ low to high edge that follows the setting of FF1, and FF2 also remains set for one clock interval. CLKF/ = 0 occurs when FF1 sets, producing CNT ENABLE = 0 to inhibit

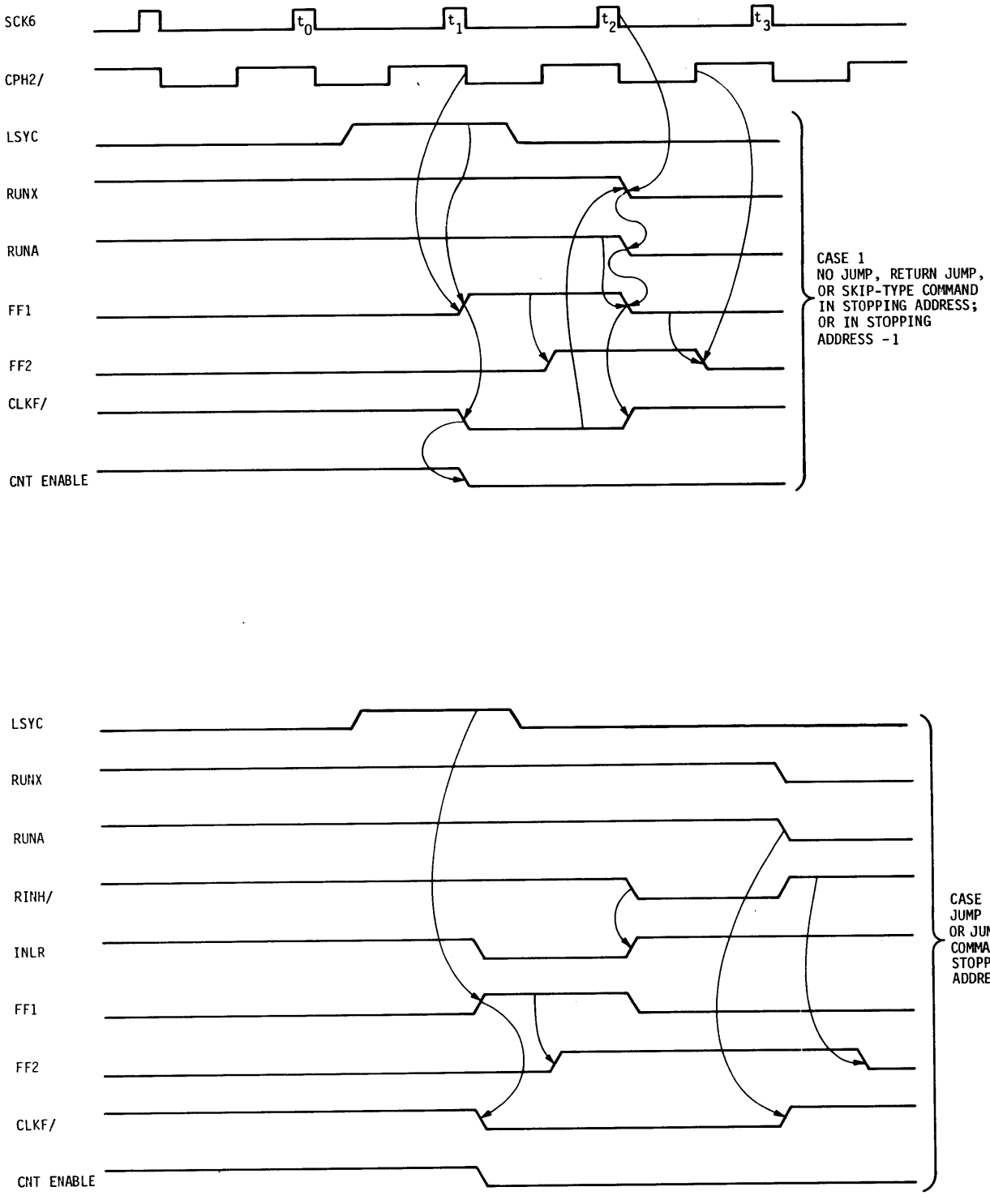


Figure 2-23. Address Stop Timing, Case 1 and Case 2

further counting in the L Register. Then, at the next clock, RUNX = 0 occurs to produce the CPU HALT. When the HALT occurs, the L Register contains the stopping address plus one.

2.7.1.5.2 Case 2 (Figure 2-23). Case 2 shows the timing when any jump type microcommand except the return jump type is located at the stopping address. Flip-flop FF1 sets on the next CPH2/ high-to-low edge following the occurrence of LSYC = 1. The setting of FF1 allows the setting of FF2 on the next CPH2/ low to high edge. FF1 is set for one clock period.

During the time FF1 is set, LR01 = 1 exists, producing RINH/ = 1. This produces CNT ENABLE = 0 and LRLL/ = 0 which inhibits the count mode of the L Register and enables the jump address to be loaded into the L Register on clock pulse t_2.

LR01 = 1 and RINH/ = 1 both exist between clock pulses t_1 and t_2 producing INLR = 0, which prevents the RUNX flip-flop from being reset on clock pulse t_2.

CLKF/ = 0 exists during the time FF1 = 1 or FF2 = 1 exists and RUNA = 1. RINH/ = 0 and FF1 = 0 both exist between clock pulses t_2 and t_3, inhibiting the reset of FF2 in the same clock period. Thus, CLKF/ = 0 exists at clock pulse t_3 which, since INLR is now equal to 1, allows the RUNX flip-flop to reset on clock pulse t_3.

The CPU then enters the HALT state with the L Register containing the jump address. Flip-flop FF2 then resets on the next CPH2/ low to high edge following clock time t_3 to terminate the Case 2 address stop sequence.

2.7.1.5.3 Case 3 (Figure 2-24). Case 3 shows the timing when a return jump type microcommand is located at the stopping address. Flip-flop FF1 sets on the next CPH2/ high to low edge following the occurrence of LSYC = 1. The setting of FF1 allows the setting of FF2 on the next CPH2/ low to high edge.

During clock intervals t_1 and t_2 CONT = 1 and RINH/ = 1 both exist during clock period which produces RUNA = 0 and CLKF/ = 1 during this clock period. RUNA = 0 also inhibits FF1 from setting at clock time t_2 in order to keep FF2 set until an extended jump (op code and the address contained in the L Save Register) has been forced into the R Register at clock time t_2 and is being decoded during clock period t_2 and t_3.

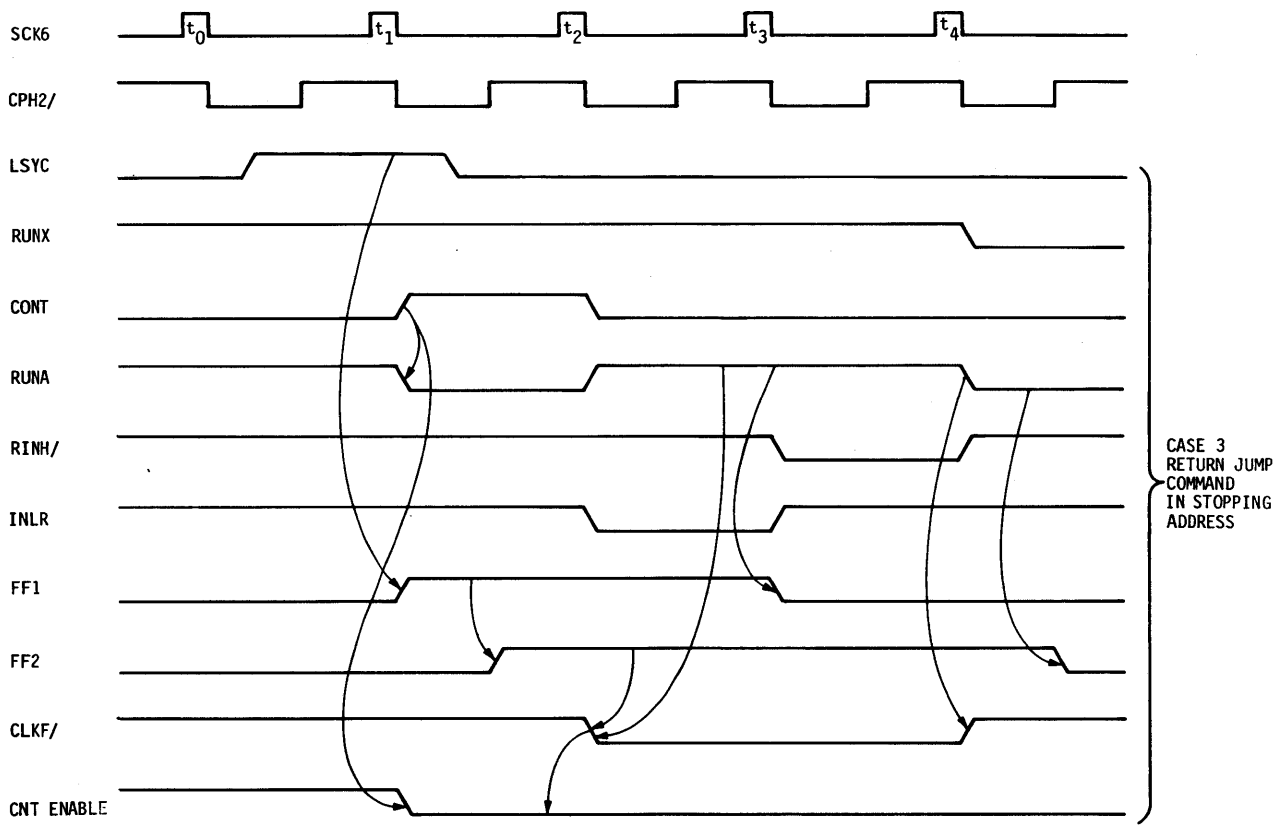


Figure 2-24. Address Stop Timing, Case 3

From clock time t_2 on, the timing is the same as the jump type microcommand described for Case 2 (refer to Paragraph 2.7.1.5.2).

2.7.1.5.4 Case 4 (Figure 2-25). Case 4 shows the timing when a skip-type command is in the stopping address and the skip is taken. If no skip is taken, timing is identical to Case 1, and the L Register stops at the skip address. If a skip occurs, INLR = 0 is produced at clock time t_1, delaying the occurrence of RUNX = 0 until clock time t_3. The occurrence of RINH/ = 0 delays the reset of flip-flop FF2 until the CPH2/ leading edge following clock time t_3. Consequently, when the HALT occurs, the L Register contains the stopping address plus two (skip address plus one) skip address plus one (skip).

2.7.2 MACHINE STATE CONTROL AND CLOCK CIRCUITS (Figure 2-4)

2.7.2.1 RUN AND INHIBIT CONTROL. The run and inhibit control (logic diagrams 0210 and 0211) includes the run/halt flip-flop (RUNX), the run inhibit flip-flop (RINH), the panel interrupt flip-flop (INTC), and the two switch sync flip-flops (SS1 and SS2). A complete description and timing diagram for the run/halt flip-flop is provided in Paragraph 2.6.3. A complete description and timing diagram for the run

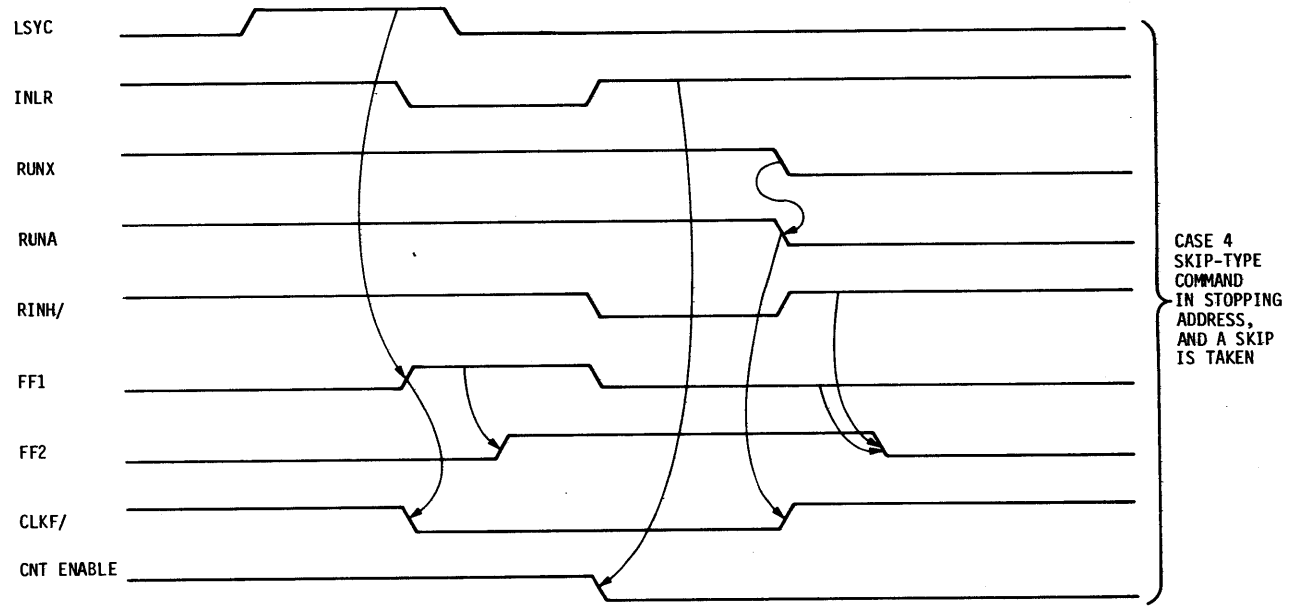


Figure 2-25. Address Stop Timing, Case 4

inhibit flip-flop is provided in Paragraph 2.6.4. A complete description and timing diagram for the panel interrupt flip-flop is provided in Paragraph 2.6.5. Switch sync flip-flops SS1 and SS2 operate as follows:

a. In the normal (quiescent) condition, SS1 is reset and SS2 is set.

b. When RUNF/ = 0, STPF/ = 0, CLKF/ = 0, or INTF/ = 0 occurs, RSHI = 1 sets flip-flop SS1 at the next occurrence of CLK6/. With SS1 set, setting of the RUNX and INTC flip-flops occurs.

c. When SS1 = 1 exists and a clock stop has not occurred (CSTP/ = 1), flip-flop SS2 resets at the next occurrence of CLK8.

d. With SS1 set and SS2 reset, setting of the RUNX and INTC flip-flops is inhibited.

e. When RUNF/, STPF/, CLKF/, and INTF/ are all in the 1 state, RSHI = 0 resets flip-flop SS1. The reset of SS1 sets SS2, restoring the switch sync flip-flops to their quiescent states.

2.7.2.2 CLOCK GENERATOR. The clock generator (logic diagram 0220) contains the 20-MHz crystal-controlled clock circuit and the following 5-MHz clock derivatives of the 20-MHz clock:

a. CPH1

b. CPH2

c. FLCK/

d. MNCK/
e. CKS1
f. CKS1/
g. SCK6

A complete description and timing diagram for these clocks is provided in Paragraph 2.6.2.

2.7.2.3 CLOCK GATES. The clock gates (logic diagram 0230) include the clock stop and count enable circuits, and the gating that derives the following internal CPU clocks from 33-nanosecond-wide, 5-MHz clock SCK6:

a. CLK1 and CLK1/
b. CLK2/
c. CLK6/
d. CLK7/
e. CLK8/
f. CLK9/
g. MCLR/
h. TCLK/
i. TCLR/
j. TMCK/

Descriptions and timing diagrams for clock stop (CSTP), CLK7/, CLK8, and CLK9/ are provided in Paragraph 2.6.6. Descriptions and timing diagrams for CLK1, CLK1/, and CLK6/ are provided in Paragraph 2.6.2. A description and timing diagram for CLK2/ is provided in Paragraph 2.6.7. A description and timing diagram for MCLR/ is provided in Paragraph 2.6.8. A description and timing diagram for TCLK/ is provided in Paragraph 2.6.9. Descriptions and timing diagrams for TCLR/ and TMCK/ are provided in Paragraph 2.6.11.2. A description and timing diagram for count enable signal CNT ENABLE is provided in Paragraph 2.6.10.

2.7.3 CONTROL UNIT (Figure 2-5)

2.7.3.1 L REGISTER. Complete timing for the L Register is described in Paragraph 2.6.10. When CNT ENABLE = 1 and LRLL/ = 1 both exist, the L Register (logic diagrams 0310 and 0311) functions as a straight modulo-4096 binary counter, which advances

one count on the positive-going trailing edge of each CLK6/ pulse. When CNT ENABLE = 0 and LRLL/ = 1 both exist, the L Register stops counting, and holds the current count. When CNT ENABLE = 0 and LRLL/ = 0 both exist, the L Register is parallel-loaded with a jump address on the positive-going trailing edge of CLK6/. If JPXX/ = 0 exists (jump extended or return jump) when LRLL/ = 0 occurs, bits 0 through 7 from the A bus are loaded into L Register stages 0 through 7, and R Register bits 8 through 11 are loaded into L Register stages 8 through 11. If JPXX/ = 1 exists when LRLL/ = 0 occurs (jump in 1K or jump in 1/2K command), bits 0 through 7 from the A bus are loaded into L Register stages 0 through 7. L register stage 8 assumes the state of DES0 (same as R08 from jump in 1K; same as R00 for jump in 1/2K). For a jump in 1K, L register stage 9 assumes the state of R11 and L register stages 10 and 11 remain unchanged. For a jump in 1/2K, L register stages 9 through 11 remain unchanged.

An optional jumper connection on the Data board (see note 1 on logic drawing 0311) disables counting in the four MSB of the L Register. With this jumper installed, the L Register functions as a modulo 256 binary counter. In this case the four MSB can only be changed by a jump or return jump microcommand.

2.7.3.2 L SAVE REGISTER AND L SAVE CONTROL. When an Inhibit L Save (ILS) microcommand is being executed (op code 1B00), AZ07/ = 0 and CG1B/ = 0 both occur, and the ENLS flip-flop is reset at the next occurrence of CLK1/. In this condition, ENLS = 0 exists, and data entry into the L Save Register (logic diagram 0330) is inhibited. The ENLS flip-flop is set by MRST/ = 0, or when a return jump command is decoded (CONT = 1), provided that CSTP/ = 1 exists (no clock stop), and that line SSIH/ is not jumpered to ground (L Save Stack Option not implemented). When ENLS = 1 exists, the L Save Register is loaded with L Register bits 11 through 0 each time JPXX/ = 0 occurs (extended jump or return jump). The occurrence of CONT = 1 (return jump) also produces RTNA = 1 and RTNB = 1 (gate contents of L Save Register onto RS bus), RTNX/ = 0 (force op code 0 into upper R Register stages 15 through 12 if U Register has previously been set to 0), and CPEN/ = 0 (inhibit) microcommand outputs from control memory). When RTNA = 0 and RTNB = 0 both exist (return jump not in progress), the L Save outputs to the RS bus are inhibited.

2.7.3.3 U REGISTER. When URXX/ = 0 occurs (load U Register) the 8-bit U Register (logic diagram 0340) is loaded with the data on the A bus when the negative-going

leading edge of the next CLK1/ pulse occurs. When MRST/ = 0 occurs, the U Register is reset to zero.

2.7.3.4 LOWER R REGISTER. Except when the SCAN mode is selected, the R Register (logic diagrams 0360 and 0361) is loaded with bits 0 through 7 of the RS bus on the positive-going trailing edge of each CLK9/ pulse. The bits on the RS bus may be any one of the following:

a. Microcommand bits 0 through 7 from control memory.

b. The states of front panel data switches 0 through 7 (if the front panel is enabled).

c. Microcommand bits 0 through 7 from control memory, AND'ed with the states of front panel data switches 0 through 7 (Enter Console Switches microcommand being executed).

d. Bits 0 through 7 from the L Save Register (return jump being executed).

e. Microcommand bits 0 through 7 from control memory AND'ed with the contents of the OD Register bits 0 through 7 (Modify Lower Commands MLC being executed).

When the SCAN mode is selected at the front panel, SCAN/ = 0 holds all lower R register stages in the reset condition, preventing the loading of microcommands on the RS bus.

When a Modify Lower Command (op code 1A00) is being executed, JUXX/ = 0 occurs. Then, while FLCK/ and CSTP/ are both high, the bits (0 through 7) from the output data (OD) register are AND'ed with the corresponding bits of the next microcommand placed on the RS bus. The resulting modified command bits are loaded into the lower R Register on the positive-going trailing edge of the next CLK9/ pulse.

2.7.3.5 UPPER R REGISTER AND U REGISTER ENABLE. The upper R Register (logic diagrams 0362 and 0363) is loaded with bits 8 through 15 of the RS bus on the positive-going trailing edge of each CLK7/ pulse, except when one of the following occurs: the SCAN mode is selected, an extended jump is being executed, or operate-type command with op code 8 through F contains code 7_8 in the r field.

When the SCAN mode is selected at the front panel, SCAN/ = 0 holds all upper R register stages in the reset state, preventing the loading of microcommands on the

RS bus. When a jump extended command (RS bus bits 12 through 15 are all zero's) is on the RS bus, or when the microcommand on the RS bus has op code 8 through F (RS15 = 1), and code 7 is contained in r field bits RS02 through RS00, RENU = 1 occurs. When RENU = 1 exists, the eight bits from the U Register are OR'ed with RS bus bits 8 through 15 of the jump extended or operate-type command. Thus, for proper operation of jump extended command, the U Register must have been previously loaded with zeros. The resulting modified command bits are loaded into the upper R Register on the positive-going trailing edge of the next CLK9/ pulse.

2.7.3.6 COMMAND DECODER. Tables 2-2 through 2-4 indicate which command decodes are active for each microcommand in the CPU repertoire. Table 2-1 is a glossary that defines each command decode signal, and indicates the logic diagram sheet of origin. First-level command decoding is performed by the op code decoder and the destination decoder on logic diagrams 0370 through 0374. The op code decoder decodes the code in upper R Register bits R15 through R12. The destination decoder is enabled for op codes 1, and 7 through F. When the op code is 1, bits R08 through R11 are routed to the destination decoder for decoding. When the op code is 7 through F, bits R00 through R02 (r field) are routed to the destination decoder. In either case, the decoder destination signal indicates which register is to be loaded (refer to Table 2-4).

Second- and third-level command decoding is performed by the gating elements on logic diagrams 0372 through 0374. All signals generated on these sheets are fully defined in Tables 2-1 through 2-4.

2.7.4 CORE MEMORY SECTION (Figure 2-6)

2.7.4.1 M REGISTER. The 8-bit M Register (logic diagram 0410) is loaded with bits 0 through 7 from the A bus. The M Register load is enabled when MRXX/ = 0 exists (load M Register). The output follows the input during the MNCK/ pulse and latches-up on the positive trailing edge of MNCK/. The M Register is cleared by an MCLR/ pulse, which occurs only when a Load N microcommand (op code 13) is being executed. The address stored in the M Register is gated to the memory address bus (connected to core memory) when DMAS/ = 1 exists (DMA not selected).

An optional jumper connection on the Data board (see logic drawing 0410, note 2) disables stage 7 of the M Register. If this jumper is installed, the memory addressing capability is reduced to 32K.

2.7.4.2 N REGISTER. The 8-bit N Register (logic diagram 0420) is loaded with bits 0 through 7 from the A bus. The N Register load is enabled only when NRXX/ = 0 exists (load N Register). The output follows the input during the MNCK/ pulse and latches-up on the positive trailing edge of MNCK/. The address stored in the N Register is gated to the memory address bus (connected to core memory) when DMAS/ = 1 exists (DMA not slected).

2.7.4.3 MEMORY CONTROL. The memory control (logic diagrams 0450 and 0451) develops the timing signals that control memory read and write cycles initiated by either the CPU or the DMA. A complete description of the timing signals developed by the memory control is provided in Paragraph 2.6.11.

2.7.4.4 MEMORY DATA GATES. The memory data gates (logic diagrams 0430 and 0431) control the transfer of memory read data from the MD bus to the T Register during a memory read operation. At clock time t_3 of a memory read cycle (see Figure 2-24), the TMCK/ pulse occurs, and the memory read data on the MD bus is gated to the T Register.

2.7.4.5 T REGISTER. The 8-bit T Register (logic diagrams 0430 and 0431) is synchronously loaded from the A bus, and is asynchronously loaded from the memory data gates. At clock time t_1 of a memory read cycle (see Figure 2-24), the TCLR/ pulse clears the T Register in preparation for loading memory read data. Then, at clock time t_3, the memory read data from the memory data gates are applied to the direct set inputs of the T register flip-flops.

Entry of A bus bits 0 through 7 into the T Register occurs on the positive-going trailing edge of the TCLK/ pulse (refer to Paragraph 2.6.10).

2.7.4.6 MEMORY DATA (MD) REGISTER. When signal MTSV = 0 exists (see Figures 2-23 and 2-24), the MD Register (logic diagram 0440) copies the contents of the T Register. The contents of the MD Register are supplied to the MD bus whenever TDBC/ = 0 exists (see Figure 2-24). When TDBC/ = 1 exists, all MD register output lines are forced to the 1 state.

2.7.5 PROCESSOR SECTION (Figure 2-7)

2.7.5.1 B BUS MULTIPLEXER. The B bus multiplexer (logic diagrams 0510 and 0511) receives either complemented or uncomplemented data from the input data bus gates, complemented and uncomplemented data from the T Register, and the eight bits from the lower R Register. Table 2-5 lists all possible state combinations of the command decodes that control B bus data selection. For each state combination, the data selected for presentation on the B bus is indicated in Table 2-5 (refer to Tables 2-2 through 2-4 for command decode states for each microcommand).

TABLE 2-5. B BUS DATA SELECTION

COMMAND DECODE STATES				DATA SELECTED FOR ENTRY ONTO B BUS
BNTP	BNTM	BENI	BENR	
1	0	0	0	Uncomplemented T Register bits 0 through 7
0	1	0	0	Complemented T Register bits 0 through 7
1	1	0	0	All zero's
0	0	1	0	Complemented or uncomplemented data from input data bus gates 0 through 7
0	0	0	1	Uncomplemented lower R Register bits 0 through 7
0	0	0	0	All one's

2.7.5.2 ARITHMETIC/LOGIC UNIT. The arithmetic/logic unit (ALU) performs logical and arithmetical operations on the data from the B bus and the selected File Register (logic diagram 0520). Table 2-6 lists all possible state combinations of the command decodes and the initial carry signal that control the type of operation performed by the ALU (refer to Tables 2-2 through 2-4 for command decode states for each microcommand). For each state combination, the type of arithmetical or logical operation is indicated. When command decode FINH = 1 (file inhibit) exists, the ALU serves as a port that passes the data on the B bus to the A bus multiplexer. When all zero's are placed on the B bus, the ALU serves as a port for passing File Register data to the A bus multiplexer.

TABLE 2-6. ARITHMETIC/LOGIC UNIT OPERATIONS

COMMAND DECODE STATES						INITIAL CARRY (CYIA/)	OPERATION PERFORMED
AENS/	ASRF/	SHLX/	XRXX/	ANDX/	ORXX/		
0	1	1	1	1	1	1	Force AU00 through AU07 to all 1 state.
1	0	1	1	1	1	1	Force AU00 through AU07 to all 1 state.
1	1	0	1	1	1	1	Shift contents of selected File Register one bit to the left towards the most significant bit, and enter 0 in vacated (LSB) bit (shift file left).
1	1	0	1	1	1	0	Shift contents of selected File Register one bit position to the left towards the most significant bit, and enter 1 in vacated (LSB) bit position (shift file left and insert).
1	1	1	0	1	1	0 or 1	Exclusive OR of B bus data and selected file register data.
1	1	1	1	0	1	0 or 1	Logical AND of B bus data and selected file register data.
1	1	1	1	1	0	0 or 1	Logical OR of B bus data and selected file register data.
1	1	1	1	1	1	1	Add B bus data to selected file register data.*
1	1	1	1	1	1	0	Add B bus data to selected file register data, and add 1 to the sum.**

*If B bus contains all zero's, unmodified file register data is passed to the A bus multiplexer. If FINH = 1 exists, unmodified B bus data is passed to the A bus multiplexer. If a subtraction operation is in progress, the result is a one's complement subtraction.

**If a subtraction operation is in progress, the result is a two's complement subtraction.

2.7.5.3 A BUS MULTIPLEXER. The A bus multiplexer (logic diagrams 0530 and 0531) receives data inputs from the Status and File Zero Flag Circuits (internal status byte), the front panel sense switches (sense switch states), the arithmetic/logic unit (AU00 through AU07), the file register output buffer (selected file register data), and the initial carry logic (initial carry CYIA). Table 2-7 lists all possible state combinations of the command decodes that control A bus data selection (refer to Tables 2-2 through 2-4 for command decode states for each microcommand). For each state combination, the data selected for presentation on the A bus is indicated.

TABLE 2-7. A BUS DATA SELECTION

COMMAND DECODE STATES				DATA SELECTED FOR ENTRY ONTO A BUS							
ASRF/	AENI/	SHRX/	AENS/	A07	A06	A05	A04	A03	A02	A01	A00
				SHIFT RIGHT FOUR							
0	1	1	1	1	1	1	1	F07	F06	F05	F04
				INTERNAL STATUS BYTE							
1	0	1	1	PWRF/	STPF/	MBDY/	MPAR	MPRO/	RTCI/	SPARE/	INTC/
				SHIFT RIGHT							
1	1	0	1	CYIA	F07	F06	F05	F04	F03	F02	F01
				SENSE SWITCHES							
1	1	1	0	ES07	ES06	ES05	ES04	1	1	1	1
				COPY ALU OUTPUT							
1	1	1	1	AU07	AU06	AU05	AU04	AU03	AU02	AU01	AU00

2.7.5.4 FILE REGISTER CONTROL. When a select primary or secondary file microcommand (op code 1040, 1060, 1080, or 10A0) is being executed, bits R07 and R06 determine whether the primary or secondary file is enabled (logic diagram 0550). When R07 = 1, file control flip-flop FBK2 is set by the negative-going trailing edge of FBCL.

With FBK2 set, the secondary file is enabled until another select file microcommand is executed. When R06 = 1, file control flip-flop FBK2 is reset by FBCL, and the primary file is enabled. When FAZO/ is low (File Zero Flags not selected) either FBS1/ = 0 (select primary file) or FBS2/ = 0 (select secondary file) is produced to select the primary or secondary file.

When bits R08 through R11 are 0, FAZO = 1 occurs, selecting the file zero flags. When the file inhibit command decode is active (FINH = 1) during the execution of a microcommand, signals F071 through F001 are all made false, entering an all-zero byte into the file register output buffer.

2.7.5.5 PRIMARY AND SECONDARY FILE REGISTERS. The primary and secondary files each contain 15 registers (logic diagram 0540). When FBS1/ = 0 exists, the primary file is selected; when FBS2/ = 0 exists, the secondary file is selected. The 8-bit output from the selected File Register is supplied to the file register output buffer. On the negative-going leading edge of file register clock CLK2/, A bus bits 0 through 7 are stored in the selected File Register.

2.7.5.6 FILE REGISTER OUTPUT BUFFER. The 8-bit outputs of the file zero flag, the primary and secondary files, and the file register control (F071 through F001) are AND'ed at the input of the file register output buffer (logic diagram 0550). Except when a file inhibit condition occurs (F071 through F001 are all zero), the selected data source for the file register output buffer is the primary file, the secondary file, or the file zero flags. When a file inhibit occurs, the F071 through F001 inputs override the data from the selected source and all zero's are entered into the file register output buffer. Data entry into the file register output buffer occurs during the FLCK/ pulse and data is latched to the file register output buffer on the positivegoing trailing edge of the FLCK/ pulse.

2.7.5.7 LINK REGISTER. The Link Register (logic diagram 0560) is active during the execution of Add (op code 8), subtract (op code 9), compare (op code 6), and shift (op code F) microcommands, all of which produce SACS = 1. The Link Register contains two flip-flops, which are selected as described later, to store the link bit resulting from an Add (ADXX/ = 0), Subtract (SBXX/ = 0), Compare (CPXX/ = 0), or shift (SHXX/ = 0) operation. For an Add, Compare, or Subtract operation, the link bit is the carry-out (CRY7) of the high-order bit of the ALU. For a shift left (SHLX = 1)

operation, the link bit is the most significant bit (F07) from the file register output buffer. For a shift right operation, the link bit is the least significant bit (F00) of the file register output buffer. Storage of the selected link bit in the selected link flip-flop occurs on the positive-going leading edge of the CLK1 pulse.

The jumper connections shown on logic diagram 0560 are for operation of the link register in the Separate Link mode. In the Separate Link mode, both the memory link (MLXX) flip-flop and the arithmetic flip-flop (ARTL) are utilized. When the r field of operate-type microcommands Add, Subtract, or Shift selects the M or N Registers (MNRX = 1), the link bit is stored in the memory link (MLXX) flip-flop. When the r field of these same microcommands do not select the M or N Register (MNRX = 0), the link bit is stored in the arithmetic link (ARTL) flip-flop.

Microcommands that utilize the link conditions previously stored in either the MLXX or ARTL flip-flops, must specify in their r field M or N to select the MLXX flip-flop output. If they do not specify M or N, the ARTL flip-flop output will be the Link bit.

The combined Link mode is selected by making the jumper connections described in the note on logic diagram 0560. In the combined Link mode, only the ARTL flip-flop is selected to store the link conditions and selected as the Link bit. However, as shown in Figure 2-7, the Link Register may also be controlled by a remotely located, optional memory/arithmetic link switch. When this switch is included, all jumper connections are removed to permit the state of the remotely located switch to select the separate or combined link mode as described previously.

2.7.5.8 INITIAL CARRY LOGIC. The initial carry logic (logic diagram 0560) determines whether a 1 or a 0 is entered as the carry in (CYIA/) to the least significant bit of the ALU. The initial carry bit (CYIA) is also used by the A bus multiplexer during shift right operations (refer to Table 2-7). As shown in Table 2-6, the state of the initial carry bit affects the operation performed by the ALU during arithmetic (add, subtract, and shift left operations. Table 2-4 shows the state of the initial carry bit for operate-type microcommands in the CPU repertoire. The initial carry logic is only active for operate-type microcommands with op codes 8 through F. For other microcommands CYIA = 0 always exists. As shown in Table 2-4,

the initial carry bit is forced to the 1 or 0 state by some microcommands. In other cases, the initial carry bit is a 1 only when the LINK output from the link register and the link control bit (bit 7 of c field) are both 1.

2.7.5.9 A BUS ZERO DECODE. The A bus zero decode (logic diagram 0570) detects the presence of an all-zero byte on the A bus. When an all-zero byte is present, AZ07/ = 0 occurs.

2.7.5.10 OVERFLOW DECODE. The overflow decode (logic diagram 0570) supplies an overflow status indication to the condition code flip-flops in the Status and File Zero Flag Circuits. An overflow condition occurs (AFLT/ = 0) when either of the following conditions exists:

a. For op codes 1, 2, 3, 6, 7, 8, 9, A, and B (ADDX = 1), an overflow condition occurs when either of the following conditions exists, indicating that the carry out from the high-order ALU bit (AU07) differs from the carry in to the high-order bit:

   1. AU07 = 0, and B bus bit B07 and file register bit F07 are both 1.
   2. AU07 = 1, and bits B07 and F07 are both 0.

b. During execution of an op code F shift command (SHXX = 1), the out-shifted bit (to be stored in the Link Register) is a 1 (SHCY = 1), indicating a shift overflow.

2.7.6 STATUS AND FILE ZERO FLAG CIRCUITS (Figure 2-8)

2.7.6.1 POWER FAIL CIRCUIT. In the quiescent state, the following conditions exist in the power fail circuit (logic diagrams 0620 and 0621):

a. Both PWRF/ and RUNF/ are high (inactive).

b. The PWRD/ input from the power supply is high (inactive), indicating no power failure is occurring.

c. The 8-bit shift register contains a 1 in bit 3, and zero's in all other bits. In this state (state 3), the gated SCK6 clock input to the shift register is inhibited by PWRD/ = 1, and no shifting action takes place.

The sequence of actions when a power failure occurs is as follows:

a. The power failure detection circuit in the power supply is a critically loaded dc-voltage supply with a threshold element. Whenever primary power is being lost, the threshold element triggers a latch in the power supply,

producing PWRD/ = 0. The normal dc voltages of the power supply are assured of remaining within their regulation bands for two milliseconds after PWRD/ = 0 occurs.

b. The occurrence of PWRD/ = 0 permits the SCK6 clock to be gated to the 8-bit shift register, and the shift register advances to state 4.

c. If, when state 4 occurs, the CPU is already in the HALT state (RUNX = 0), the master reset one-shot triggers, producing MRST/ = 0 and PFRS/ = 0. The occurrence of PFRS/ = 0 resets the shift register to state 0. With the shift register in state 0, PWRD/ = 0 holds MRST/ and PFRS/ in the 0 state, after the master reset one-shot pulse terminates, and the power fail sequence terminates.

d. If, when state 4 occurs, the CPU is in the RUN state (RUNX = 1), the shift register advances to state 5 at the next SCK6 clock. In state 5, the power fail flip-flop (PWRF) sets, producing RUNF/ = 0 and PWRF/ = 0. The occurrence of PWRF/ = 0 initiates an internal status interrupt.

e. At the occurrence of the next SCK6 clock, the shift register advances to state 6. In this state, RUNX = 1 inhibits the SCK6 clock to the shift register, and the shift register is held in state 6.

f. While the shift register is holding in state 6, an Enter Internal Status (EIS) microcommand is executed by the firmware in response to the internal status interrupt produced by PWRF/ = 0. When the EIS microcommand is executed, AENI = 1 gates the CLK1/ pulse to the PWRF flip-flop, and the flip-flop resets, producing PWRF/ = 1 and RUNF/ = 1.

g. Following execution of the EIS microcommand, the CPU should enter the power-is-failing firmware subroutine, during which the contents of all volatile registers and flags are stored in core memory. When the power-is-failing subroutine is completed, a halt microcommand is executed, placing the CPU in the HALT state (RUNX = 0).

h. The occurrence of RUNX = 0 causes the shift register to advance to state 7 at the next SCK6 clock. No further action takes place in state 7. Then, at the next SCK6 clock, the shift register advances to state 8.

i. When shift register state 8 occurs, the master reset one-shot triggers, producing MRST/ = 0 and PFRS/ = 0 (refer to Paragraph 2.6.12 for master reset actions). The occurrence of PFRS/ = 0 resets the shift register to state 0. With the shift register in state 0, PWRD/ = 0 holds MRST/ and PFRS/ in the 0 state after the master reset one-shot pulse terminates, and the power-fail sequence terminates.

At the end of the power-fail sequence, the CPU is in the HALT state, the master reset (MRST/) is active, and the shift register is in state 0. This condition continues until the power supply detection element detects that normal ac power has resumed. When power-up condition is detected, the PWRD/ latch in the power supply

resets, producing PWRD/ = 1. When PWRD/ = 1 occurs, the power fail circuit initiates the following automatic restart sequence:

a. When PWRD/ = 1 occurs, MRST/ and PFRS/ both return to the 1 state if the master reset one-shot has timed out. With PFRS/ = 1, the SCK6 clock to the shift register is again enabled.

b. At the first SCK6, the shift register advances from state 0 to state 1. In state 1, the PWRF flip-flop is set, producing PWRF/ = 0 and RUNF/ = 0. State 1 also triggers the power-on one-shot.

c. The occurrence of RUNF/ = 0 (as set from state 1) causes the CPU to return to the RUN state (RUNX = 1). The occurrence of PWRF/ = 0 (as set from state 1) produces another internal status interrupt.

d. At the next SCK6 clock, the shift register advances to state 2. The advance to state 3 is inhibited by the power-on one-shot pulse that was initiated in state 1. While the shift register pauses in state 2, an Enter Internal Status (EIS) microcommand is executed by the firmware in response to the internal status interrupt produced by PWRF/ = 0. When the EIS microcommand is executed, AENI = 1 gates the CLK1/ pulse to the PWRF flip-flop, and the flip-flop resets, producing PWRF/ = 1 and RUNF/ = 1.

e. Following the execution of the EIS microcommand, the CPU enters the recovery firmware subroutine. During this time all registers and flags are restored to their pre-fail state so the macro level program can resume at the point where it was interrupted by the power fail. The power-on one-shot is timed for approximately 200 microseconds to allow the recovery software time to complete before the power-fail sequencer again senses the PWRD/ signal in state 3.

f. When the power-on one-shot pulse terminates, the shift register advances to state 3. In state 3, the existence of PWRD/ = 1 inhibits the SCK6 clock to the shift register, and the shift register remains in state 3 (quiescent state), terminating the automatic restart sequence.

2.7.6.2 REAL TIME CLOCK. The real time clock (logic diagram 0630) is a 12-stage preset counter with jumper preset connections normally made so that the binary counting modulus is decimal 2500. Real time count control flip-flop RTCC receives the 5-MHz CLK6/ clock, and changes state at each CLK6/ pulse. At every other CLK6/ pulse, when RTCC is high, the real time clock counter advances one count. Consequently, the counter advances one count every 400 nanoseconds, and reaches a full count (RTC3 = 1) once each millisecond.

The real time clock enable/disable flip-flop is set (enable real time clock) by the CLK1/ clock when an enable real time clock (ERT) microcommand (op code 1720) is executed (CG7X = 1, LRXX = 1, and A05X = 1). The flip-flop is reset (disable real time

clock) by the CLK1/ clock when a disable real time clock (DRT) microcommand (op code 1710) is executed (CG7X = 1, LRXX = 1, and A04X = 1). When the real time clock is disabled, the counter is loaded with the preset starting count (decimal 1596), and held at this count. When the real time clock is enabled, the counter begins counting from the preset value at the 400-nanosecond (2.5-MHz) rate. After one millisecond, the full count is reached, RTC3 = 1 occurs, the counter is again loaded with the decimal 1596 preset value, and the RTCI flip-flop sets (RTCI/ = 0) at the next CLK6/ clock. After being loaded with the preset value, the counter resumes counting. The CPU responds to the real time clock interrupt (RTCI) by executing an enter internal status (EIS) microcommand, producing AENI = 1. If CSTP/ = 1 (not clock stop) exists when AENI occurs, the RTCI flip-flop is reset at the next CLK6/ clock, terminating the interrupt. This cycle repeats at one-millisecond intervals until a disable real time clock microcommand is executed.

2.7.6.3 INTERNAL STATUS INTERRUPT CONTROL. The internal status interrupt control (logic diagram 0610) receives signal PWRF/ from the power-fail circuit, signal RTCI/ from the real time clock, signal STPF/ from the Front Panel Control Circuits STEP switch, signal INTC/ from the run and inhibit control, and four spare status line inputs (MPRO/, SPARE/, MBDY/, and MPAR/). When any of these input lines goes low (interrupt active), signal INTL/ = 0 occurs, indicating that an internal status interrupt has occurred.

2.7.6.4 EXTERNAL INTERRUPT CONTROL. When an enable external interrupt (EEI) microcommand (op code 1708) is executed (CG7X = 1, LRXX = 1, and A03X = 1), interrupt enable flip-flop INTE is set (logic diagram 0610). In this condition, the occurrence of EINT/ = 0 produces INTR/ = 0, indicating that an external interrupt has occurred. When a disable external interrupt (DEI) microcommand (op code 1704) is executed (CG7X = 1, LRXX = 1, and A02X = 1), flip-flop INTE is reset, and the generation of INTR/ = 0 is inhibited.

2.7.6.5 CONDITION CODE FLIP-FLOPS. When an I/O microcommand (op code 7) is being executed (IOXX/ = 0), or when the update condition code bit (R04) is true in a microcommand with op code 8, 9, B, C, D, E, or F (BENR/ = 1 and MYXX/ = 1), CCOD = 1 occurs (logic diagram 0570). With CCOD = 1 present, flip-flop CCOX assumes the state of overflow status indication AFLT/ at the next CLK1/ clock, and flip-flop CC1X assumes the state of A bus bit A07 at the same CLK1/ clock.

With CCOD = 1 present, and if CC2D = 1 also exists, flip-flop CC2X assumes the state of A bus zero decode AZ07/ at the same CLK1/ clock. Signal CC2D = 1 is produced when any one of the following conditions exists:

a. IOXX/ = 0 is present (op code 7)
b. R07X = 0 is present (link bit of operate-type command is 0)
c. CC2X/ = 0 exists (flip-flop CC2X is already in the set state).

2.7.6.6 FILE ZERO FLAGS. The file zero flags consist of eight gates which supply the file zero flag inputs to the file register output buffer when FAZO = 1 exists (file zero selected in an operate-type microcommand). The eight file zero flag input lines are fully described in Paragraph 2.4.4.2.

2.7.7 FRONT PANEL DISPLAY INDICATORS (Figure 2-9)

2.7.7.1 RS BUS DRIVERS. The RS bus lines (logic diagrams 0350 and 0351), which are common to the front panel switches, the control memory, and the L Save Register are applied to the non-inverting RS bus drivers. Outputs RS15A through RS00A are supplied to the display multiplexer.

2.7.7.2 L REGISTER DRIVERS. L Register inputs L11X through L00X (logic diagram 0140) are applied to the non-inverting L Register drivers. The outputs (L11A through L00A) are supplied to the display multiplexer and to the L address comparator in the Front Panel Control Circuits.

2.7.7.3 DISPLAY MULTIPLEXER. The 16-bit display multiplexer (logic diagrams 0710 through 0740) selects the data supplied to the lamp drivers. Each of the 16 display multiplexer stages is a 4:1 selector. Signals RS0X and RS1X, whose states depend upon which display select pushbutton is pressed, determine which of the four inputs to each stage is gated to the lamp drivers. Table 2-8 summarizes the Display multiplexer selection actions.

2.7.7.4 LAMP DRIVERS AND DATA LIGHTS. Each of the 16 data lights (logic diagrams 0710 through 0740) are driven by an NPN transistor lamp driver. When the input to a lamp driver is high (1 state), the transistor is switched on, providing a current path to ground for the associated data indicator.

TABLE 2-8. DISPLAY MULTIPLEXER DATA SELECTION

ACTIVE DISPLAY SELECT PUSHBUTTON	RESULTING DISPLAY SELECT CODE		SELECT DATA	
	RS0X	RS1X	LAMPS 15 THRU 8	LAMPS 7 THRU 0
D	0	0	All Zero's	A07 thru A00
M	1	1	M07 thru M00	N07 thru N00
L	1	0	Zero's in bits 15 thru 12; L11 thru L08 in bits 11 thru 8	L07 thru L00
C	0	1	RS15 thru RS08	RS07 thru RS00

2.7.8 I/O CONTROL SECTION (Figure 2-10)

2.7.8.1 I/O CONTROL (IC) REGISTER. When an external I/O command is being executed (logic diagram 0810), the occurrence of EXIO = 1 enables the IC Register to be loaded at the next occurrence of the CLK1/ clock. When CLK1/ occurs, bits R04 through R06 from the lower R Register are stored in the IC Register. The inverted outputs (I01X/ through I03X/) from the IC Register are supplied to the byte I/O bus. If any of the three outputs are low (active), indicating that the IC Register contains a non-zero I/O control code, IOSV = 1 is produced.

2.7.8.2 OUTPUT DATA (OD) REGISTER. When IOSV = 0 exists (logic diagram 0810), indicating the IC Register contains an all-zero code, the OD Register copies the contents of the T Register. When IOSV = 1 occurs, OD Register loading is inhibited, and the OD Register retains the data that was present during the clock time before IOSV = 1 occurred.

2.7.8.3 INPUT DATA BUS GATES. The eight input data bus gates (logic diagram 0810) select either complemented or uncomplemented data from the input data bus for application to the B bus multiplexer. When both IDHT = 1 (op code 7 or bit R05 = 1) and SBXX/ = 1 (not subtract operation) both exist, uncomplemented data is selected for application to the B bus multiplexer. When either IDHT = 0 or SBXX/ = 0 occurs, complemented data is selected.

When BENI = 1 exists, complemented data from the input data bus is selected for any of the following:

a. Subtract Command (op code 9)

b. Logical OR (op code C), Exclusive OR (op code D), or AND (op code E) command in which c field bits 6 and 5 contain code 10_2.

When BENI = 1, uncomplemented data from the input data bus is selected for any of the following commands:

a. Add (op code 8) or copy (op code B) command.

b. Logical OR (op code C), Exclusive OR (op code D), or AND (op code E) command in which c field bits 6 and 5 contain code 01_2.

2.8 FLOW DIAGRAM DESCRIPTIONS

Figure 2-26 is a flow block diagram that provides an overall instruction flow diagram index. Data on Figure 2-26 is indexed to op code functions and the specific drawing (Figure 2-27 through 2-38) on which a detailed flow diagram is presented for each op code. Also included on Figure 2-26 is a description of the symbols used in Figures 2-26 through 2-38.

Initially, the effective microcommand is loaded into the R Register where command execution occurs. Overall flow for command execution, designating the appropriate figure and area in which this execution occurs, is illustrated on Figure 2-26. The loading of the microcommand into the R Register and the processing of the command into op codes is illustrated on Figure 2-27. Processing of the various op codes is illustrated on Figure 2-28 through 2-38, which are figure and area reference to one another by input or output symbols as described on Figure 2-26.

The following paragraphs provide descriptions and conditions for the execution of op codes illustrated on Figures 2-27 through 2-28.

Note

Unless otherwise specified, op codes in the following descriptions are given in hexadecimal form.

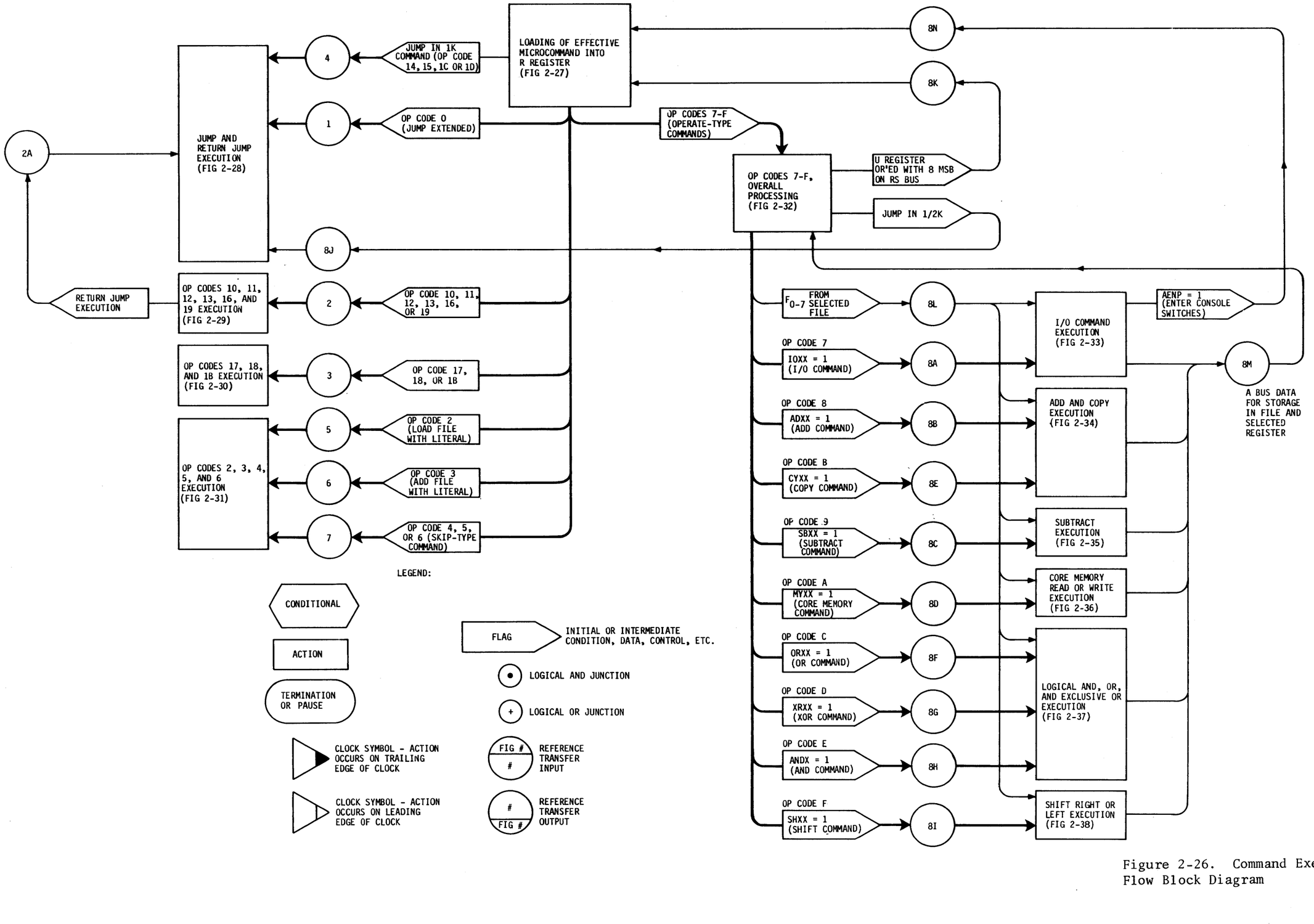


Figure 2-26. Command Execution Flow Block Diagram

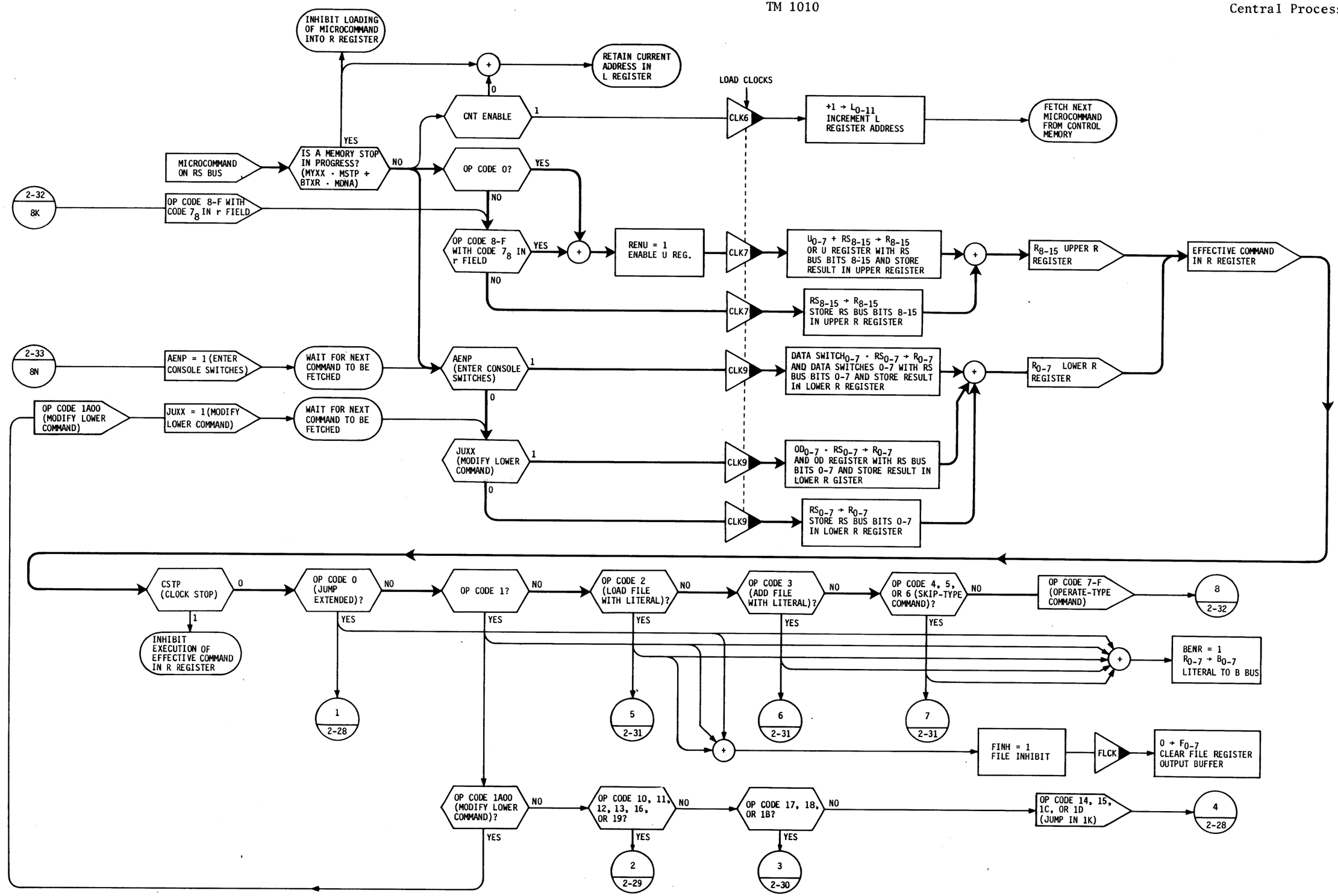


Figure 2-27. Loading of Effective Microcommand in R Register

2.8.1 LOADING OF EFFECTIVE MICROCOMMAND INTO R REGISTER (Figure 2-27)

Clocks CLK7 and CLK9 occur simultaneously at 200-nanosecond intervals when a memory stop is not in progress. When CLK7 and CLK9 occur, the R Register is loaded with an effective microcommand. The command loaded into the R Register is the same as the command on the RS bus, except when one of the following conditions exist:

a. Op code 0 is on the RS bus. In this case, RENU = 1 occurs, the contents of the U Register are OR'ed with bits 8 through 15 of the command on the RS bus, and the result is stored in R register stages 8 through 15 at CLK7 time.

b. An operate-type microcommand (op code 8 through F) is present on the RS bus, and the r field of this microcommand contains code 7_8. In this case, RENU = 1 occurs, the contents of the U Register are OR'ed with bits 8 through 15 of the command on the RS bus, and the result is stored in R register stages 8 through 15 at CLK7 time.

c. The preceding command (already stored in the R Register) was Enter Console Switches (ECS). In this case, AENP = 1 occurs, the states of console data switches 0 through 7 are AND'ed with bits 0 through 7 of the next command on the RS bus, and the result is stored in the R register stages 0 through 7 at CLK9 time.

d. The preceding command (already stored in the R Register) was Modify Lower Command (MLC). In this case, JUXX = 1 occurs, the contents of the output data (OD) register are AND'ed with bits 0 through 7 of the next command on the RS bus, and the result is stored in R register stages 0 through 7 at CLK9 time.

Clock CLK6 occurs at the same time as CLK7 and CLK9. If CNT ENABLE = 1 exists when CLK6 occurs, the L Register is incremented one count to fetch the next microcommand from control memory.

The op code in the high-order bits of the effective microcommand stored in the R Register determines the type of command execution performed, as follows:

1. Op codes 0, 14, 15, 1C and 1D are executed as shown on Figure 2-28.
2. Op codes 10, 11, 12, 13, 16 and 19 are executed as shown on Figure 2-29.
3. Op codes 17, 18 and 1B are executed as shown on Figure 2-30.
4. Op codes 2, 3, 4, 5 and 6 are executed as shown on Figure 2-31.
5. Op codes 7 through F are executed as shown on Figure 2-32.
6. Op code 1A00 (modify lower command) produces JUXX = 1, which modifies the next command as described in condition d above.

For op codes 0, 1 and 2; FINH = 1 occurs, clearing the file register output buffer at the next occurrence of FLCK. With the file register output buffer cleared, the ALU operates only on data supplied from the B bus.

For op codes 0 through 6; BENR = 1 occurs, selecting R register bits 0 through 7 as data (literal) on the B bus.

2.8.2 JUMP AND RETURN JUMP EXECUTION (Figure 2-28)

A jump or return jump is executed when any of the following four conditions are present:

a. A jump extended command (op code 0) is in the R Register, producing JPXX = 1. In this case, R register bits 11 through 0 contain the extended jump address.

b. A return jump is being executed as a result of one of the commands shown on Figure 2-29 being executed. In this case, the sequence shown on Figure 2-29 enters op code 0 into R register bits 15 through 12 (producing JPXX = 1), and enters the L Save address into R register bits 11 through 0.

c. A jump in 1K command (op code 14, 15, 1C or 1D) is in the R Register. In this case the jump address is contained in R register bits 11 and 8 through 0.

d. A jump in 1/2K is being executed as a result of one of the commands shown on Figure 2-32 being executed (op code 7 through F with code 4_8 or 5_8 in r field). In this case, the MSB of the jump address is contained in R register bit 0, and the eight LSB of the jump address are on the A bus.

When condition a, b, or c is present, the eight LSB of the jump address from R register states 7 through 0 are supplied via the B bus and the ALU to the A Register. When condition d is present, the eight LSB of the jump address are already on the A bus. Then, at CLK6 time, the eight LSB from the A bus are loaded into L register stages 7 through 0. At the same CLK6 time, L register bits 11 through 8 are loaded as follows:

a. For a jump extended or return jump execution (JPXX = 1), R register bits 11 through 8 are loaded into L register stages 11 through 8.

b. For a jump in 1K execution, R register bit 8 is loaded into L register stage 8, R register bit 11 is loaded into L register stage 9, and L register bits 10 and 11 are unchanged.

c. For a jump in 1/2K execution, R register bit 0 is loaded into L register stage 8, and L register bits 9 through 11 remain unchanged.

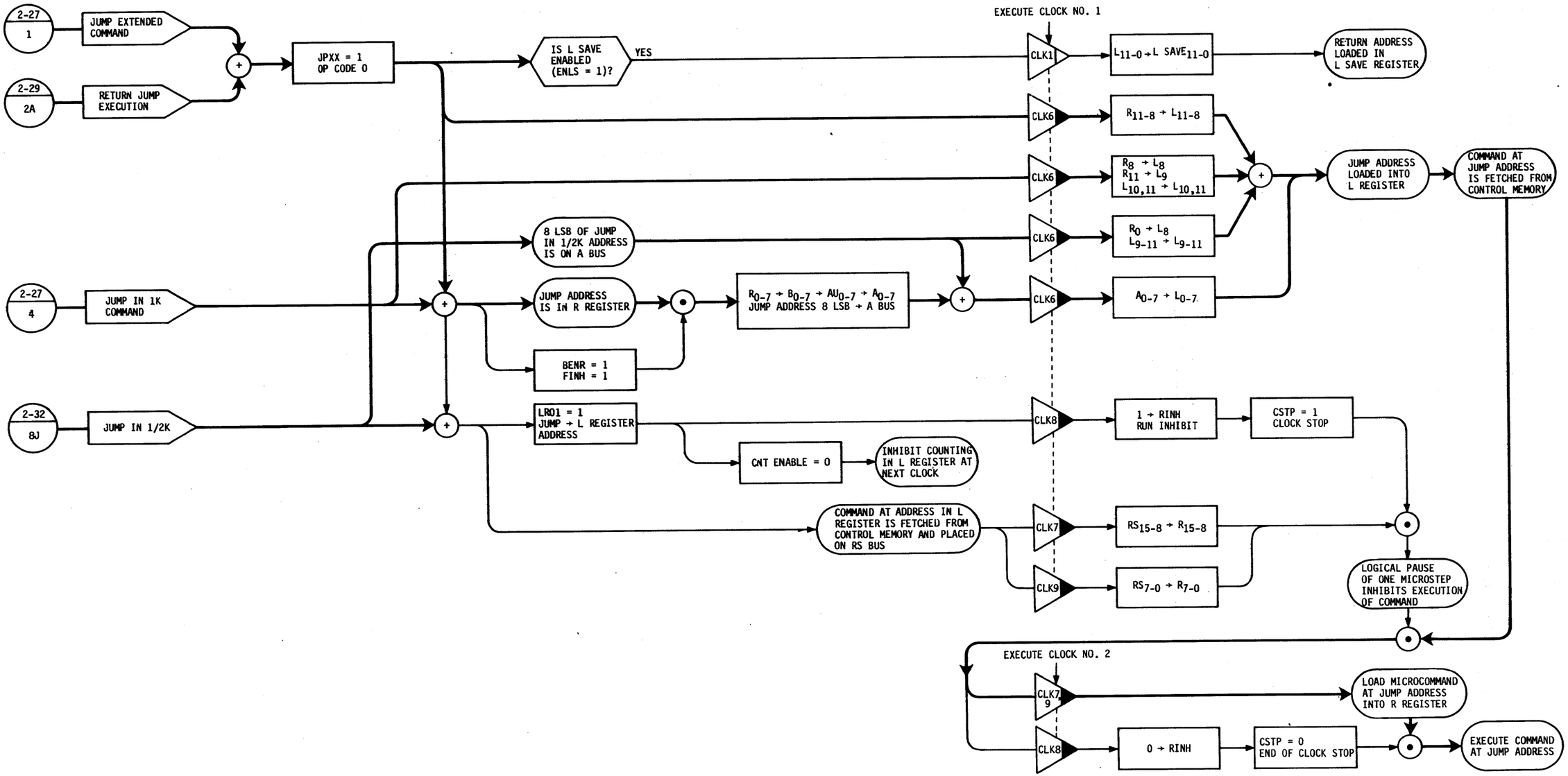


Figure 2-28. Jump and Return Jump Execution

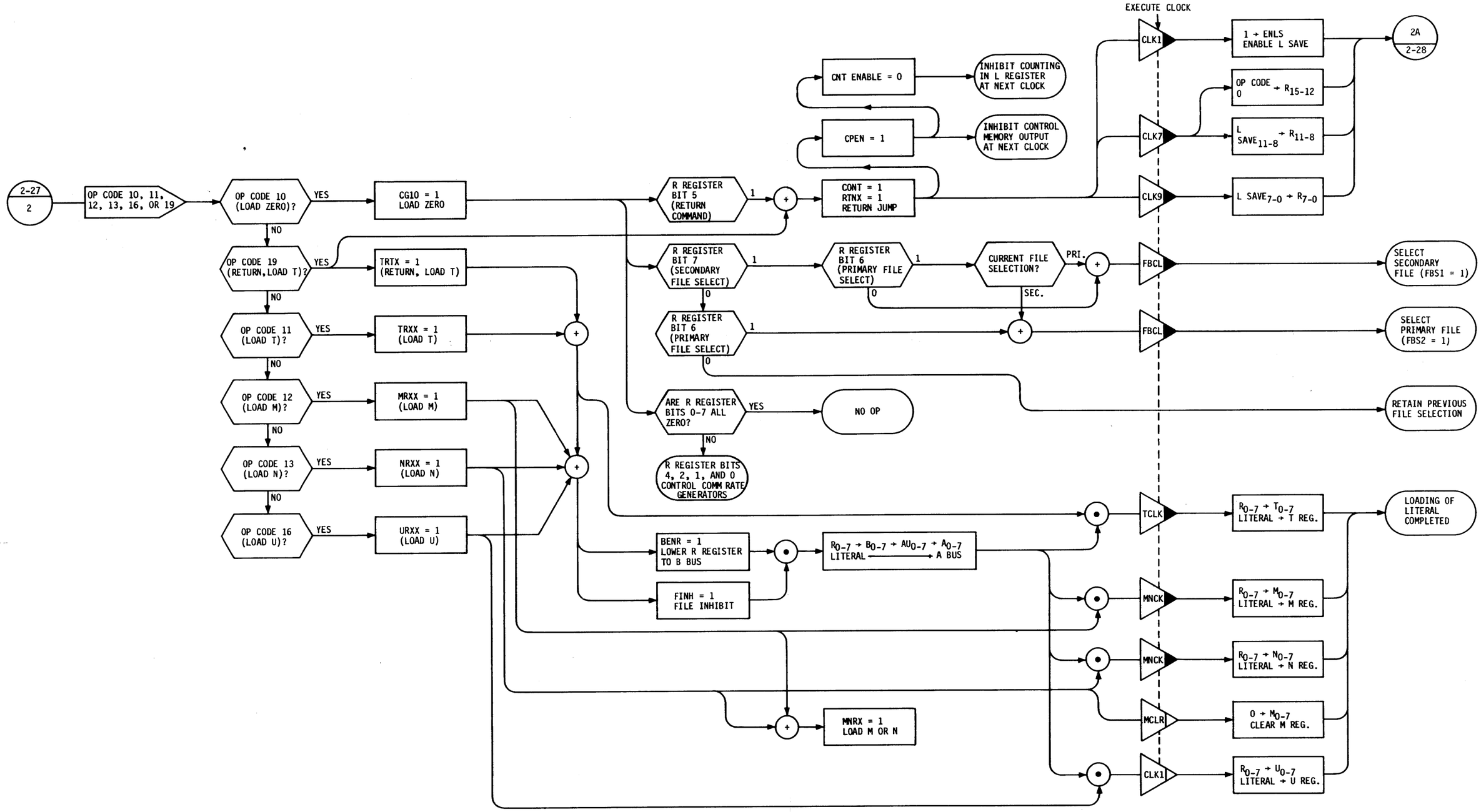


Figure 2-29. Op Codes 10, 11, 12, 13, 16 and 19 Execution

At the same clock time that the L Register is loaded with the jump address, the following actions also occur:

a. If JPXX = 1 exists (jump extended or return jump), and if ENLS = 1 also exists (ENLS is always set to 1 for a return jump, and is conditionally set to 1 for a jump extended), the contents (address that was present before the jump address is loaded) of the L Register are transferred to the L Save Register.

b. The occurrence of LR01 = 1 sets the RINH (run inhibit) flip-flop for one clock interval. This produces clock stop CSTP = 1 that inhibits execution of the command loaded into the R Register (execution of this command must be inhibited because it is pre-empted by the jump). The presence of LR01 = 1 also produces CNT ENABLE = 0, inhibiting counting in the L Register.

At the following clock time, RINH = 0 occurs, and the microcommand from the jump address is loaded into the R Register.

2.8.3 OP CODE 10, 11, 12, 13, 16 AND 19 EXECUTION (Figure 2-29)

2.8.3.1 LOAD ZERO COMMAND. Op code 10 is Load Zero (LZ). For op code 10, CG10 = 1 is present and bits 7, 6, 5, 2, 1 and 0 define the actions that are performed at the execute clock time. The actions defined by each bit are as follows:

a. If R register bit 5 = 1 (return jump), CONT = 1, RTNX = 1, CPEN = 1 and CNT ENABLE = 0 all occur, the following actions are produced:

1. The L register count is not advanced at the next clock.

2. The addressed command from control memory is inhibited.

3. ENLS = 1 occurs (enable L Save for return jump).

4. Op code 0 is loaded into R register stages 15 through 12.

5. The return address from the L Save Register is loaded into R register stages 11 through 0. Execution of the return jump is continued at the next clock, as shown on Figure 2-28.

b. R register bits 7 and 6 determine which file (primary or secondary) is selected.

R Register Bit 7	R Register Bit 6	Current File Selection	Selected File
1	1	Primary	Secondary
1	1	Secondary	Primary
1	0	Either	Secondary
0	1	Either	Primary
0	0	Either	No Change

c. R register bits 2, 1 and 0 control optional communication rate generators. There is no command execution by the CPU for these bits.

2.8.3.2 RETURN, LOAD T COMMAND. Op code 19 is Return, Load T (RLT). This command initiates a return jump (same actions as those described in item a of Paragraph 2.8.3.1), and also produces TRTX = 1, BENR = 1 and FINH = 1. These signals cause the literal in R register bits 7 through 0 to be supplied, via the B bus and the ALU, to the A bus. The literal on the A bus is then loaded into the T Register at the next TCLK time.

2.8.3.3 LOAD T COMMAND. Op code 11 is Load T (LT). This command produces TRXX = 1, BENR = 1 and FINH = 1. These signals cause the literal in R register stages 7 through 0 to be supplied, via the B bus and ALU, to the A bus. The literal on the A bus is then loaded into the T Register at the next TCLK time.

2.8.3.4 LOAD M COMMAND. Op code 12 is Load M (LM). This command produces MRXX = 1, BENR = 1 and FINH = 1. These signals cause the literal in R register stages 7 through 0 to be supplied, via the B bus and ALU, to the A bus. The literal on the A bus is then loaded into the M Register at the next MNCK time.

2.8.3.5 LOAD N COMMAND. Op code 13 is Load N (LN). This command produces NRXX = 1, BENR = 1 and FINH = 1. These signals cause the literal in R register stages 7 through 0 to be supplied, via the B bus and ALU, to the A bus. The literal on the A bus is loaded into the N Register at the next MNCK time. During the same MNCK time the M Register is cleared.

2.8.3.6 LOAD U COMMAND. Op code 16 is Load U (LU). This command produces URXX = 1, BENR = 1 and FINH = 1. These signals cause the literal in R register stages 7 through 0 to be supplied, via the B bus and ALU, to the A bus. The literal on the A bus is loaded into the U Register at the next CLK1 time.

2.8.4 OP CODE 17, 18 AND 1B EXECUTION (Figure 2-30)

2.8.4.1 LOAD SEVEN COMMAND. Op code 17 is Load Seven (LS). This command produces CG7 = 1. R register bits 7, 5, 4, 3 and 2 determine the actions that are produced.

a. If R register bit 7 = 1 (Halt), RUNR = 1 occurs. At the next CLK6 time, the RUNX flip-flop is reset and the CPU is placed in the Halt state.

b. The real time clock is enabled or disabled by R register bits 5 and 4.

R Register Bit 5	R Register Bit 4	Current Status of Real Time Clock	Resulting Real Time Clock Status
1	1	Disabled	Enabled
1	1	Enabled	Disabled
1	0	Either	Enabled
0	1	Either	Disabled
0	0	Either	No Change

c. External interrupts are enabled or disabled by R register bits 3 and 2.

R Register Bit 3	R Register Bit 2	Current Status of External Interrupt	Resulting External Interrupt Status
1	1	Disabled	Enabled
1	1	Enabled	Disabled
1	0	Either	Enabled
0	1	Either	Disabled
0	0	Either	No Change

2.8.4.2 LOAD EIGHT COMMAND. This command is not implemented, and no action is produced in the CPU.

2.8.4.3 L SAVE COMMANDS. Commands in this group all have op code 1B, producing CG1B = 1. If the command is Inhibit L Save (1B00), AZ07 = 1 occurs (A bus Zero Decode), and, at the next CLK1 time, flip-flop ENLS is reset. Resetting ENLS inhibits storage in the L Save Register until a return jump is executed. Other commands in the L Save group produce no action in the CPU. These commands are decoded and acted upon by the L Save Stack Option.

2.8.5 OP CODE 2, 3, 4, 5 AND 6 EXECUTION (Figure 2-31)

2.8.5.1 LOAD FILE COMMAND. Op code 2 is Load File (LF). This command produces LFXX = 1, BENR = 1, FINH = 1 and LRXX = 1. These signals cause the literal in R register bits 7 through 0 to be supplied, via the B bus and the ALU, to the A bus.

At the next CLK2 time, the literal on the A bus is stored in the File Register (primary or secondary) selected by R register bits 11 through 8.

2.8.5.2 ADD FILE COMMAND. Op code 3 is Add File (AF). This command produces AFXX = 1, FINH = 0, BENR = 1 and LRXX = 1. These signals cause the following actions:

a. At the next FLCK time, the contents of the File Register (primary or secondary), selected by R register bits 11 through 8, are copied by the file register output buffer and supplied to the ALU as F00 through F07.

b. The literal in R register bits 7 through 0 is supplied via the B bus to the ALU.

c. The ALU adds the literal on the B bus to the contents of the selected file, and presents the sum on the A bus.

d. At the next CLK2 time, the sum on the A bus is written back in the same file location that was accessed in step a.

2.8.5.3 TEST IF ZERO COMMAND. Op code 4 is Test If Zero (TZ). This command produces TZXX = 1, BENR = 1, LRXX = 1 and FINH = 0. These signals cause the following actions:

a. At the next FLCK time, the contents of the File Register (primary or secondary), selected by R register bits 11 through 8, are copies by the file register output buffer and supplied to the ALU as F00 through F07.

b. The literal in R register bits 7 through 0 is supplied via the B bus to the ALU.

c. The ALU AND's the literal on the B bus with the contents of the selected file, and presents the result on the A bus.

d. If the result on the A bus is all zeros, AZ07 = 1 occurs. At the next CLK8 time, the RINH (run inhibit) flip-flop is set to 1 for one-clock interval and produces clock stop CSTP = 1. The clock stop inhibits execution of the microcommand at the skip address. If AZ07 = 0 occurs, the command at the skip address is executed (RINH = 1 does not occur).

2.8.5.4 TEST IF NOT ZERO COMMAND. Op code 5 is Test If Not Zero (TNZ). This command produces TNXX = 1, BENR = 1, LRXX = 1 and FINH = 0. These signals cause the following actions:

a. At the next FLCK time, the contents of the File Register (primary or secondary), selected by R register bits 11 through 8, are copied by the file register output buffer and supplied to the ALU as F00 through F07.

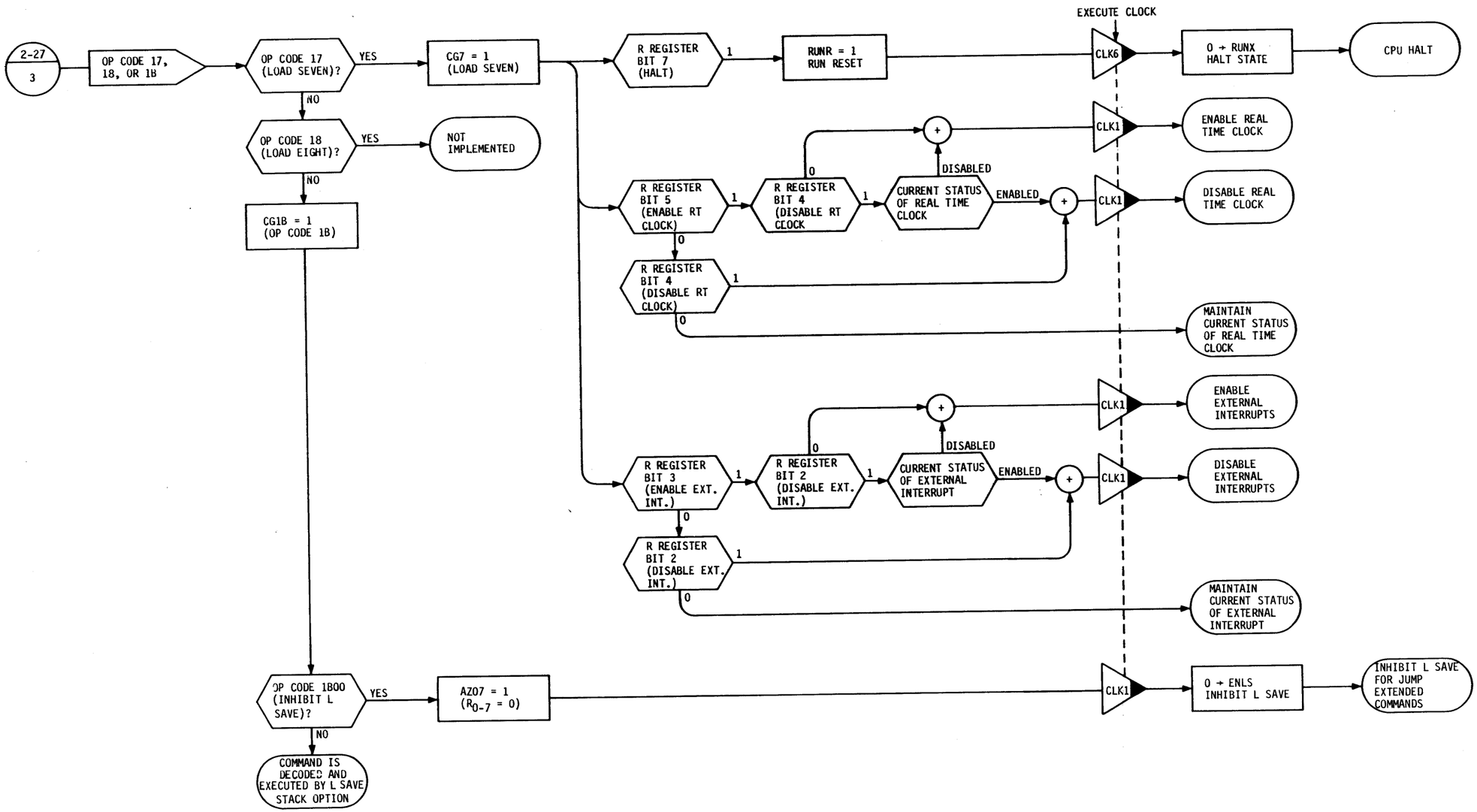


Figure 2-30. Op Codes 17, 18 and 1B Execution

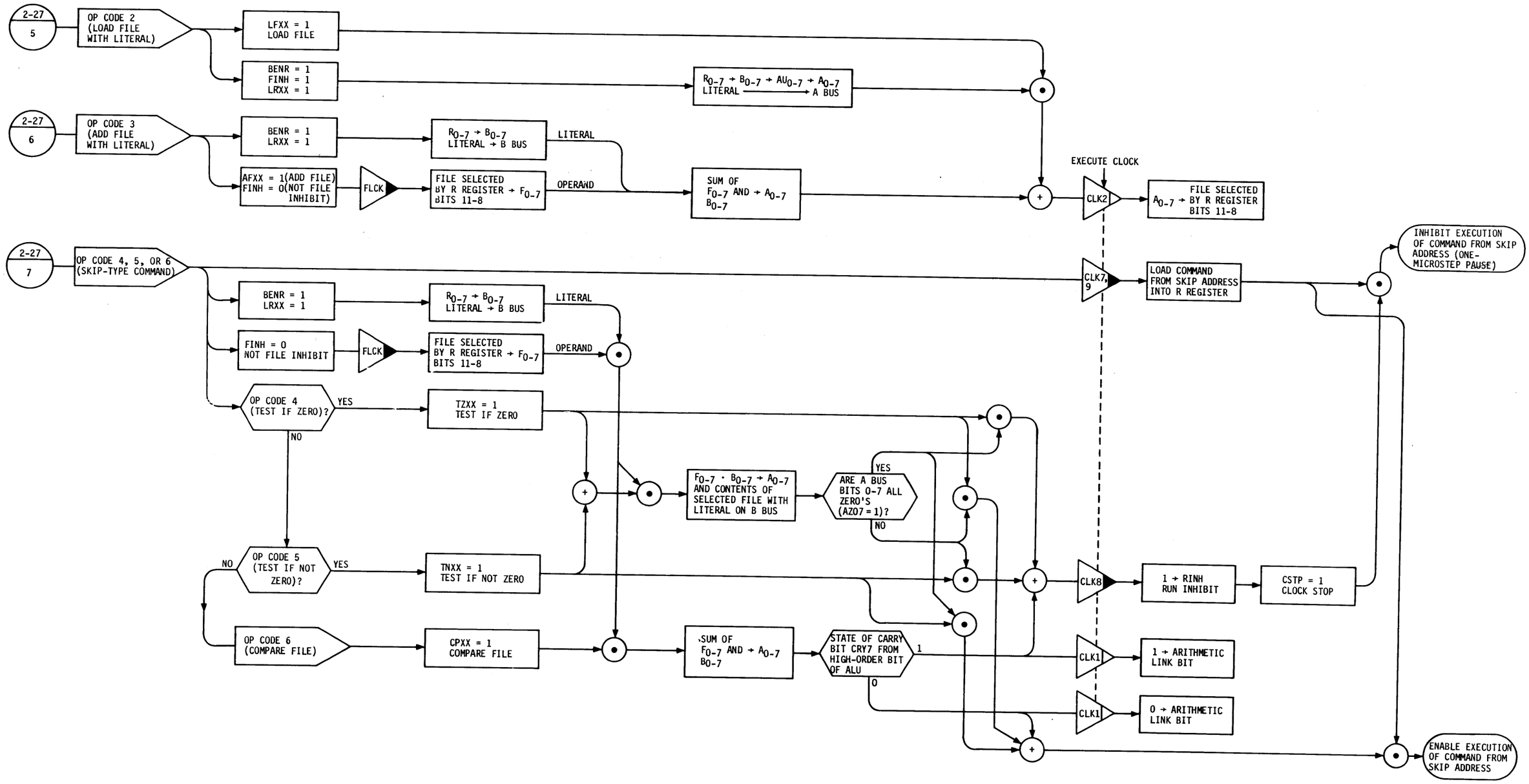


Figure 2-31. Op Codes 2, 3, 4, 5 and 6 Execution

b. The literal in R register bits 7 through 0 is supplied via the B bus to the ALU.

c. The ALU AND's the literal on the B bus with the contents of the selected file, and presents the result on the A bus.

d. If the result on the A bus is not all zeros, AZ07 = 0 occurs. At the next CLK8 time, the RINH (run inhibit) flip-flop is set to 1 for one-clock interval and produces clock stop CSTP = 1. The clock stop inhibits execution of the microcommand at the skip address. If AZ07 = 1, occurs, the command at the skip address is executed (RINH = 1 does not occur).

2.8.5.5 COMPARE FILE COMMAND. Op code 6 is Compare File (CP). This command produces CPXX = 1, BENR = 1, LRXX = 1 and FINH = 0. These signals cause the following actions:

a. At the next FLCK time, the contents of the File Register (primary or secondary), selected by R register bits 11 through 8, are copied by the file register output buffer and supplied to the ALU as F00 through F07.

b. The literal in R register bits 7 through 0 is supplied via the B bus to the ALU.

c. The ALU adds the literal on the B bus to the contents of the selected file, and presents the sum on the A bus.

d. If carry bit CRY7 from the high-order stage of the ALU is a 1 (sum greater than 255), the arithmetic link bit is set to 1 at the next CLK1 time. At CLK8 time the RINH (run inhibit) flip-flop is set for one clock interval and produces clock stop CSTP = 1. The clock stop inhibits execution of the microcommand at the skip address. If CRY7 = 0 occurs, the arithmetic link bit is set to 0, and the command at the skip address is executed (RINH = 1 does not occur).

2.8.6 OP CODE 7 THROUGH F OVERALL PROCESSING (Figure 2-32)

For op codes 7 through F (operate-type commands), the following actions occur:

a. For op code B (Copy), FINH = 1 occurs, and the file register output buffer is cleared when FLCK time occurs. For any other op code, FINH = 0 occurs, and the contents of the File Register (primary or secondary), selected by R register bits 11 through 8 (f field), are copied by the file register output buffer when FLCK time occurs.

b. Detailed execution (operation on the selected file and/or a literal on the B bus) of the commands are as follows:

1. Commands with op code 7 (I/O commands) are executed as shown on Figure 2-33.

2. Commands with op code 8 (add) or B (copy) are executed as shown on Figure 2-34.

3. Commands with op code 9 (subtract) are executed as shown on Figure 2-35.

4. Commands with op code A (core memory commands) are executed as shown on Figure 2-36.

5. Commands with op code C (OR), D (XOR) or E (AND) are executed as shown on Figure 2-37.

6. Commands with op code F (shift) are executed as shown on Figure 2-38.

c. Execution of step b produces a result on the A bus. This result is used as follows:

1. If enabled, the condition code flags and the link register are updated when CLK1 time occurs.

2. If R register bit 3 (file write inhibit) is a 0, the result on the A bus is written into the File Register (primary or secondary), selected on R register bits 11 through 8, at the next CLK2 time.

3. The register selected by the code in R register bits 2 through 0 (r field) is loaded with the result on the A bus, as follows:

(a) r field code 0_8 inhibits storage in any register

(b) r field code 1_8 selects the T Register

(c) r field code 2_8 selects the M Register

(d) r field code 3_8 selects the N Register

(e) r field code 4_8 or 5_8 selects jump in 1/2K (see Figure 2-28)

(f) r field code 6_8 selects the U Register

(g) r field code 7_8 causes the contents of the U Register to be OR'ed with RS bus bits 15 through 8 of the command (see Figure 2-27). This action is not performed for op code 7.

The operation performed by operate-type commands is determined by the op code and the code in the c field (R register bits 7 through 4). These operations are described in the following paragraphs.

2.8.7 I/O COMMAND EXECUTION (Figure 2-33)

The following describes the states of c field bits 7 through 4 which define the type of operation performed for op code 7.

a. Code 1 of the c field is Enable Sense Switches (ESS), which produces AENS = 1. The states of the four sense switches are presented as A bus bits 7 through 4. A bus bits 3 through 0 are forced to 1.

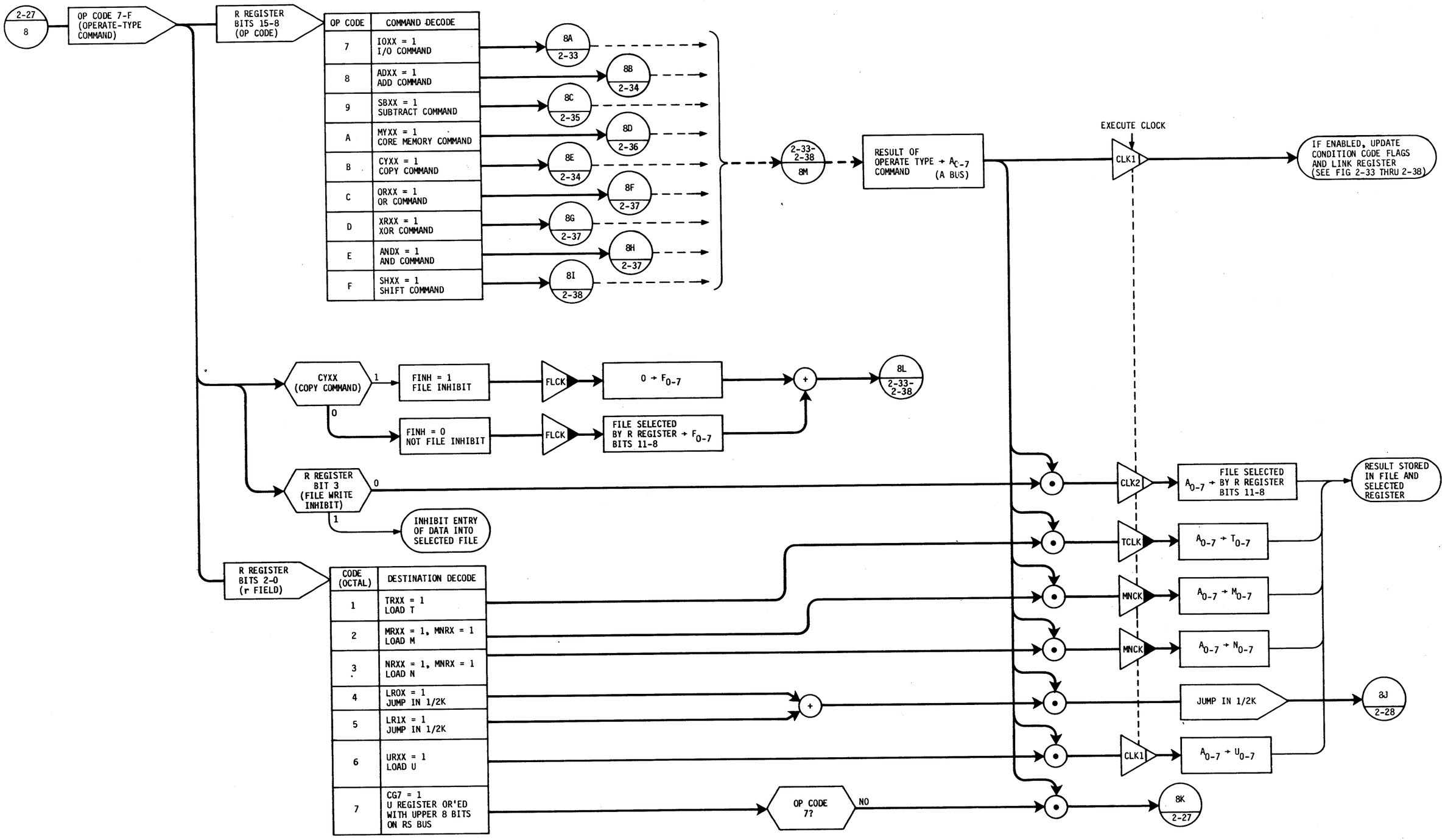


Figure 2-32. Op Codes 7 Through F Overall Processing

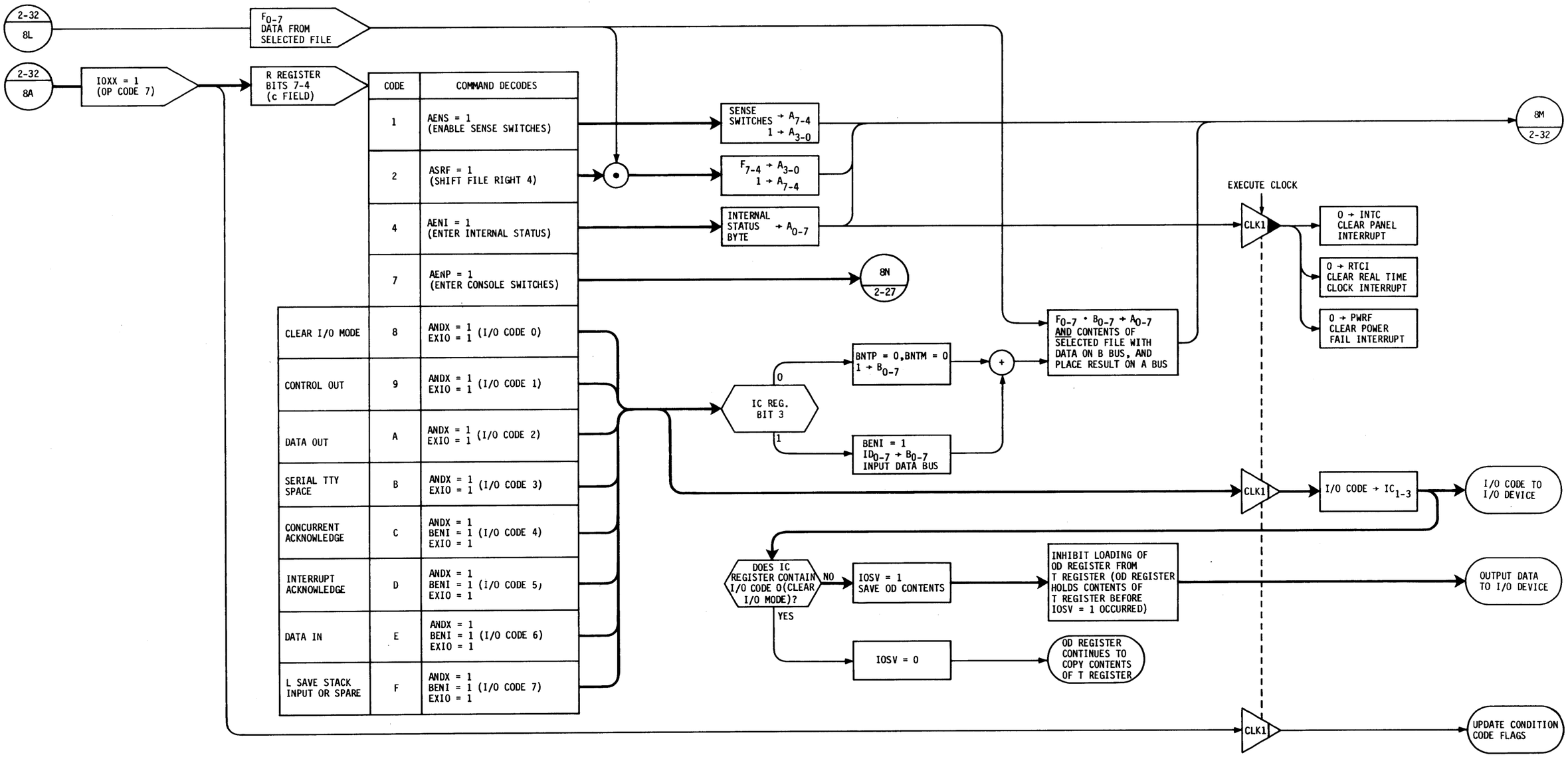


Figure 2-33. I/O Command Execution

b. Code 2 of the c field is Shift Right Four (SRF), which produces ASRF = 1. Bits 7 through 4 of the selected file are presented as A bus bits 3 through 0. A bus bits 7 through 4 are forced to 1.

c. Code 4 of the c field is Enter Internal Status (EIS), which produces AENI = 1. Status signals are presented as A bus bits 7 through 0. At the next CLK1 time, INTC (panel interrupt), RTC1 (real time clock interrupt), and PWRF (power fail interrupt) are all cleared.

d. Code 7 of the c field is Enter Console Switches (ECS), which produces AENP = 1. The states of data switches 7 through 0 are entered into R register stages 7 through 0, as shown on Figure 2-27.

e. Codes 8 through F of the c field are external I/O commands (EXIO), which produce ANDX = 1 and EXIO = 1. The presence of EXIO = 1 causes the code in c field bits 6 through 4 to be loaded into the IC Register at the next CLK1 time. If the code loaded into the IC Register is not 0_8, IOSV = 1 occurs, which prevents the OD Register from copying the T Register. The OD Register then retains the T register data that was present before the external I/O command was received. Also, the contents of the selected file are AND'ed in the ALU with the data selected for presentation on the B bus, and the result is presented on the A bus. The data selected for placement on the B bus is as follows:

   1. If IC register bit 3 contained a 1 before the EXIO command was executed, the bits from the input data bus are placed on the B bus.

   2. If IC register bit 3 contained a 0 before the EXIO command was executed, all ones are placed on the B bus.

Any command with op code 7 causes the condition code flags to be updated at the next CLK1 time. No updating is performed in the Link Register.

2.8.8 ADD AND COPY COMMAND EXECUTION (Figure 2-34)

Op code 8 is Add and produces ADXX = 1 and FINH = 0. Op code B is Copy and produces CYXX = 1 and FINH = 1. An addition is performed by the ALU for both commands. For Add, the selected operand on the B bus is added to the contents (F00 through F07) of the selected file, and the sum is placed on the A bus. For Copy, F00 through F07 are all zeros, thus the sum of the B bus operand and the selected file is the same as the B bus operand. The main difference between these two commands is the presence of FINH = 1 for the Copy command forces file register bits F00 through F07 to 0. Also, the link bit is updated only for the Add command.

2.8.8.1 INITIAL CARRY. For both Add and Copy, a 1 is added to the sum when initial carry bit CYIA is set to 1. CYIA = 1 occurs when either of the following conditions exists:

a. c field bit 6 = 1 (increment)

b. c field bit 7 (link control) is a 1, and the selected Link Register (arithmetic or memory) contains a 1. The memory link bit is selected when the r field is selecting the M or N Register (MNRX = 1). Otherwise, the arithmetic link bit is selected.

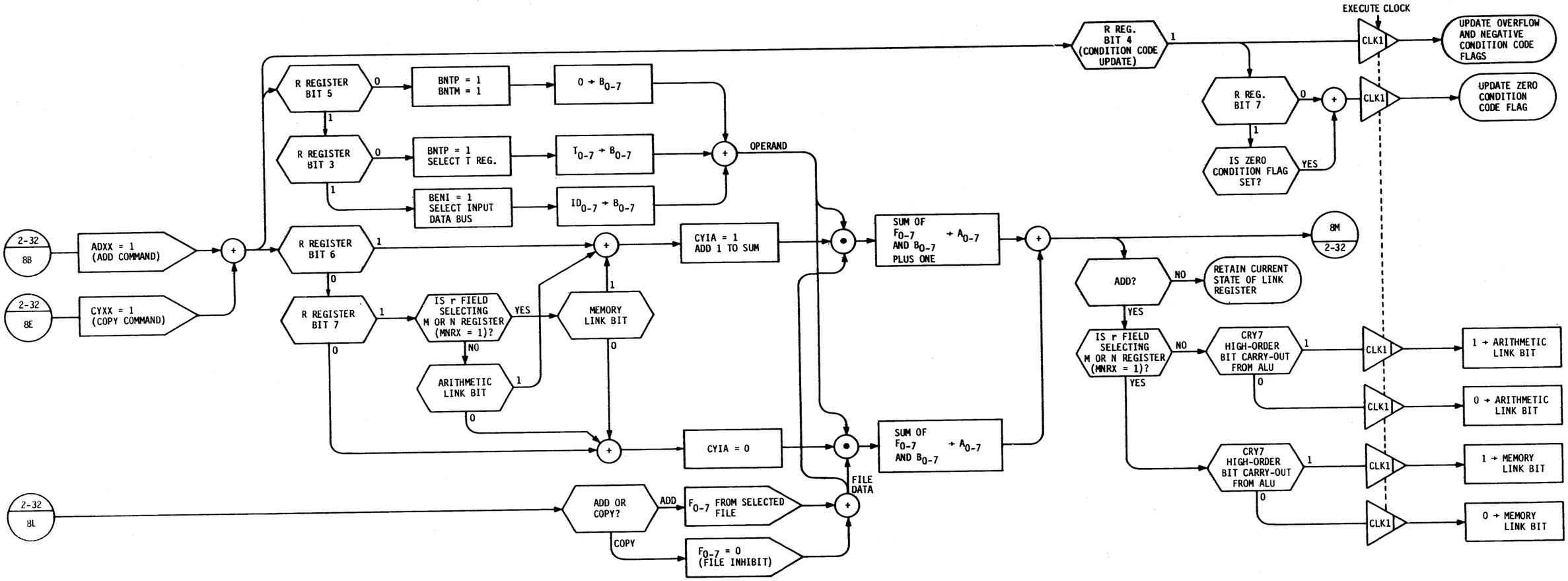
2.8.8.2 SELECTION OF B BUS OPERAND. For both Add and Copy, c field bit 5 and IC register bit 3 select the operand on the B bus.

c Field Bit 5	IC Register Bit 3	Selected Operand On B Bus
1	1	Data from Input Data Bus
1	0	Contents of T Register
0	-	All Zeros

2.8.8.3 CONDITION CODE UPDATING. For both Add and Copy, c field bits 7 and 4 control updating of the condition code flags.

c Field Bit 4	c Field Bit 7	Resulting Condition Code Updating
0	-	None
1	0	Update overflow, negative, and zero condition code flags.
1	1	Update overflow and negative condition code flags. Update zero condition code flag only if flag is already set.

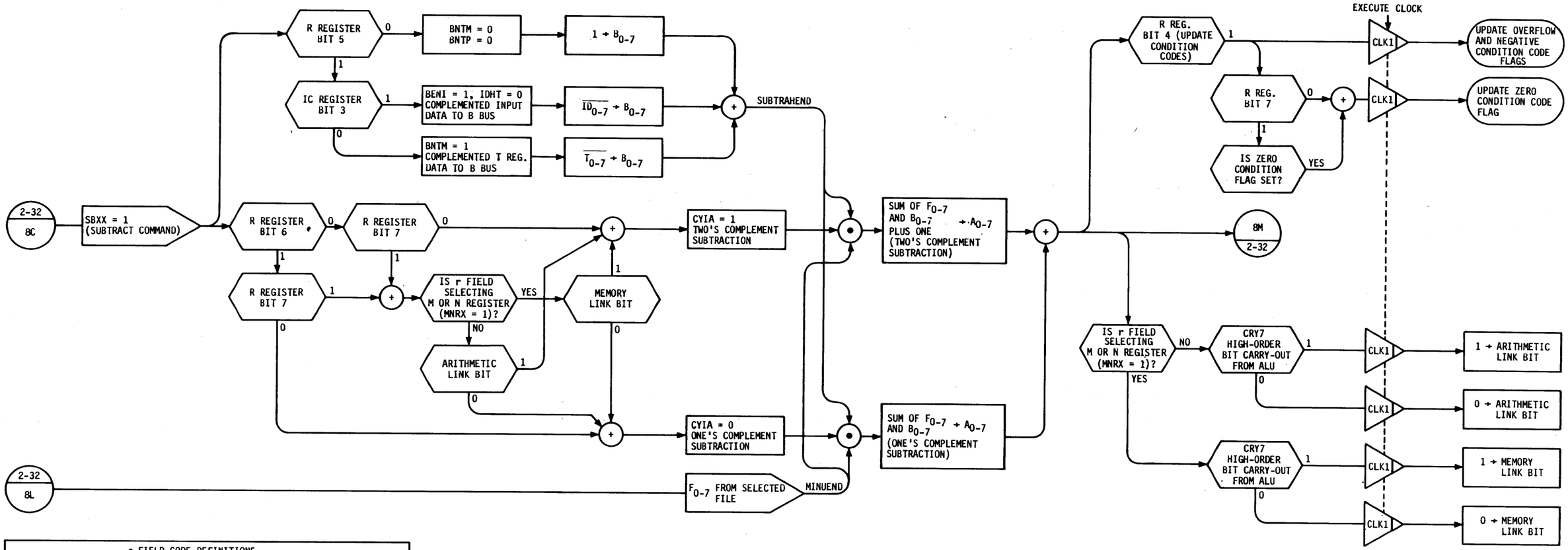
2.8.8.4 LINK REGISTER UPDATING. For the Add command only, the selected link register is updated. When the carry-out (CRY7) from the high order bit of the ALU is a 1, the link bit is set to 1. When CRY7 = 0 exists, the link bit is set to 0. When the r field is selecting the M or N Register (MNRX = 1), the memory link register is selected; otherwise, the arithmetic link register is selected.



c FIELD CODE DEFINITIONS

R REG. BIT 7	R REG. BIT 6	R REG. BIT 5	IC REG. BIT 3	ACTION PERFORMED	
				ADD	COPY
0	0	0	-	COPY FILE TO A BUS	ALL ZEROS TO A BUS
1	0	0	-	ADD LINK TO FILE, AND COPY TO A BUS	COPY LINK TO A BUS BIT 0
0	1	0	-	INCREMENT FILE AND COPY TO A BUS	ENTER 1_{16} ON A BUS
0	0	1	0	ADD T REGISTER TO CONTENTS OF SELECTED FILE, AND COPY SUM TO A BUS	COPY T REGISTER TO A BUS
0	0	1	1	ADD INPUT DATA BUS TO CONTENTS OF SELECTED FILE, AND COPY SUM TO A BUS	COPY INPUT DATA BUS TO A BUS
1	0	1	0	ADD LINK TO SUM OF T REGISTER AND CONTENTS OF SELECTED FILE, AND COPY SUM TO A BUS	ADD LINK TO T REGISTER AND COPY TO A BUS
1	0	1	1	ADD LINK TO SUM OF INPUT DATA BUS AND CONTENTS OF SELECTED FILE, AND COPY SUM TO A BUS	ADD LINK TO INPUT DATA BUS, AND COPY TO A BUS
0	1	1	0	ADD ONE TO SUM OF T REGISTER AND CONTENTS OF SELECTED FILE, AND COPY TO A BUS	ADD ONE TO T REGISTER AND COPY TO A BUS
0	1	1	1	ADD ONE TO SUM OF INPUT DATA BUS AND CONTENTS OF SELECTED FILE, AND COPY TO A BUS	ADD ONE TO INPUT DATA BUS AND COPY TO A BUS

Figure 2-34. Add and Copy Execution



c FIELD CODE DEFINITIONS				
R REG. BIT 7	R REG. BIT 6	R REG. BIT 5	IC REG. BIT 3	ACTION PERFORMED
1	-	0	-	ADD LINK TO COMPLEMENTED CONTENTS OF SELECTED FILE, AND COPY TO A BUS
0	1	0	-	COPY COMPLEMENTED CONTENTS OF SELECTED FILE TO A BUS
0	0	0	-	ADD ONE TO COMPLEMENTED CONTENTS OF SELECTED FILE, AND COPY TO A BUS
0	0	1	0	PERFORM TWO'S COMPLEMENT SUBTRACTION OF T REGISTER FROM SELECTED FILE, AND COPY TO A BUS
0	0	1	1	PERFORM TWO'S COMPLEMENT SUBTRACTION OF INPUT DATA BUS FROM SELECTED FILE, AND COPY TO A BUS
0	1	1	0	PERFORM ONE'S COMPLEMENT SUBTRACTION OF T REGISTER FROM SELECTED FILE, AND COPY TO A BUS
0	1	1	1	PERFORM ONE'S COMPLEMENT SUBTRACTION OF INPUT DATA BUS FROM SELECTED FILE, AND COPY TO A BUS
1	-	1	0	PERFORM ONE'S COMPLEMENT SUBTRACTION OF T REGISTER FROM SELECTED FILE, ADD LINK TO DIFFERENCE, AND COPY TO A BUS
1	-	1	1	PERFORM ONE'S COMPLEMENT SUBTRACTION, OF INPUT DATA BUS FROM SELECTED FILE, ADD LINK TO DIFFERENCE, AND COPY TO A BUS

Figure 2-35. Subtract Execution

2.8.9 SUBTRACT COMMAND EXECUTION (Figure 2-35)

Op code 9 is Subtract and produces SBXX = 1 and FINH = 0. The selected operand on the B bus (subtrahend) is subtracted from the contents (F00 through F07) of the selected file, and the difference is placed on the A bus. Subtraction is performed by adding the selected file to the complemented literal on the B bus.

2.8.9.1 INITIAL CARRY. The state of initial carry bit CYIA determines whether the differences on the A bus is a one's complement or two's complement difference between the literal and the selected file. When CYIA = 1 exists, a two's complement subtraction is performed. When CYIA = 0 exists, a one's complement subtraction is performed. The state of CYIA is determined by c field bits 7 and 6.

c Field Bit 7	c Field Bit 6	Resulting State of CYIA	Type of Operation
1	1	Same as link bit*	Add link to one's complement difference
1	0	Same as link bit*	Add link to two's complement difference
0	1	0	One's complement subtraction
0	0	1	Two's complement subtraction

*Memory link bit is selected if r field is selecting M or N Register (MNRX = 1); otherwise, arithmetic link bit is selected.

2.8.9.2 SELECTION OF OPERAND ON B BUS. The states of c field bit 5 and IC register bit 3 determine which operand is placed on the B bus.

c Field Bit 5	IC Register Bit 3	Selected Operand On B Bus
0	-	All One's
1	0	Complemented T Register Bits
1	1	Complemented Input Data Bus Bits

2.8.9.3 CONDITION CODE UPDATING. Condition code updating is identical to that described in Paragraph 2.8.8.3.

2.8.9.4 LINK REGISTER UPDATING. Link register updating is identical to that described in Paragraph 2.8.8.4.

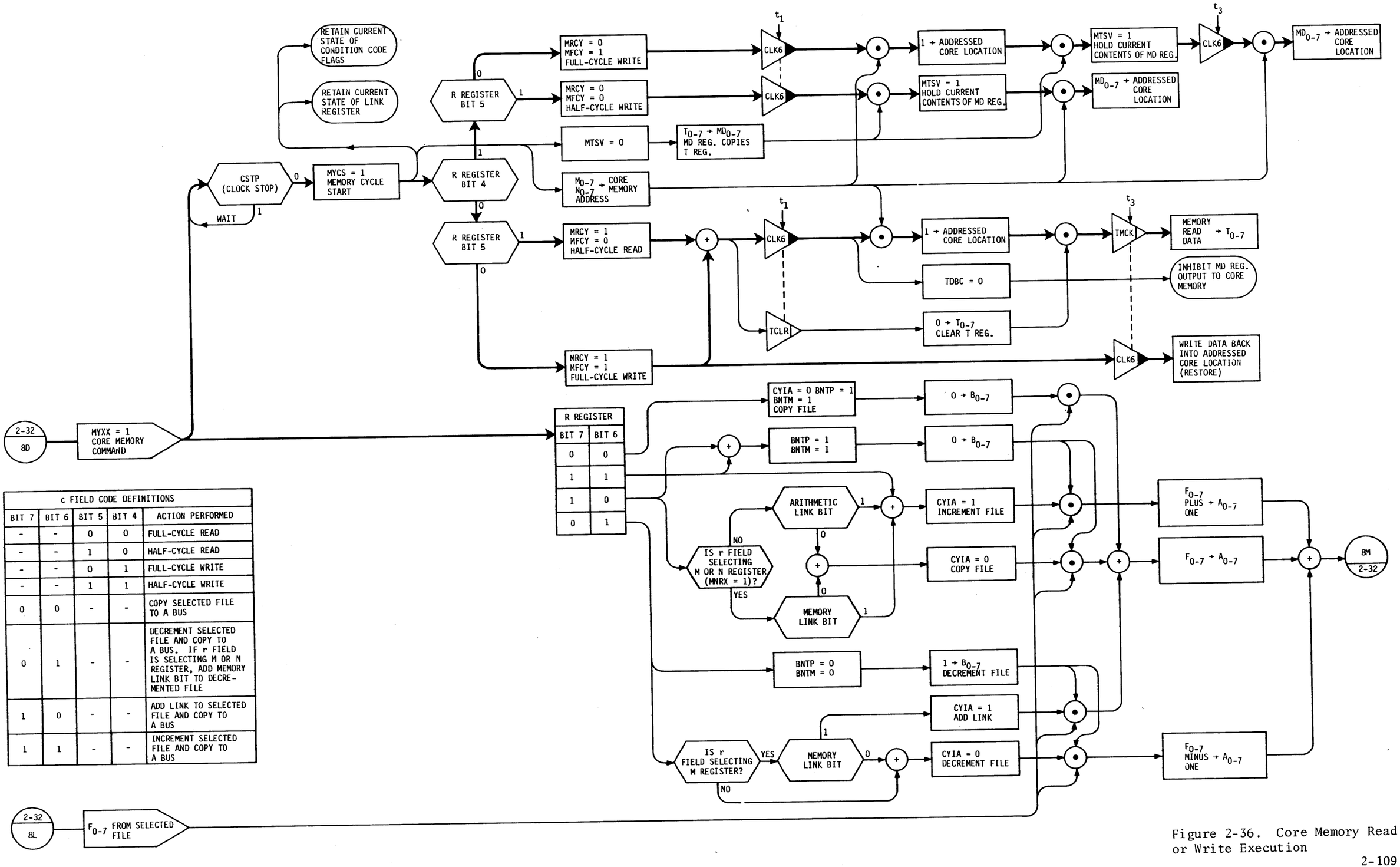
2.8.10 CORE MEMORY READ OR WRITE EXECUTION (Figure 2-36)
Op code A initiates a core memory cycle and produces MYXX = 1. During the core memory cycle, data is read from, or written into, core memory. While the read or write operation is in progress, an operation is also performed by the ALU on the contents of the selected file. The result of this operation is placed on the A bus. When MYXX = 1 is present, MYCS = 1 is produced to initiate the memory cycle; provided that no clock stop is in progress (CSTP = 0 must exist).

2.8.10.1 HALF-CYCLE MEMORY WRITE. For a half-cycle memory write operation, c field bits 4 and 5 are both 1 and produce MRCY = 0 and MFCY = 0. The presence of MTSV = 0 allows the MD Register to copy the contents of the T Register. At the next CLK6 time, MTSV = 1 occurs, and the current contents of the MD Register are held. Following the occurrence of MTSV = 1, the contents of the MD Register are written into the memory location addressed by the M and N Registers.

2.8.10.2 FULL-CYCLE MEMORY WRITE. For a full-cycle memory write operation, c field bits 5 and 4 contain code 01_2, and produce MRCY = 0 and MFCY = 1. The presence of MTSV = 0 allows the MD Register to copy the contents of the T Register. Following the next CLK6 time, all ones are written into the memory location addresssed by the M and N Registers. Following the next CLK6 (time t_3), the contents of the MD Register are written into the memory location addressed by the M and N Registers.

2.8.10.3 HALF-CYCLE MEMORY READ. For a half-cycle memory read operation, c field bits 5 and 4 contain code 10_2 and produce MRCY = 1 and MFCY = 0. Following the next CLK6 time, all ones are written into the memory location addressed by the M and N Registers. This causes the sense amplifiers and data register in the core memory to sense and store the data that was in the addressed location. Also, TCLR occurs, clearing the T Register in preparation for storing the memory read data. Following the next CLK6 (time t_3), the occurrence of TMCK causes the memory read data stored in the data register (in core memory) to be entered into the T Register.

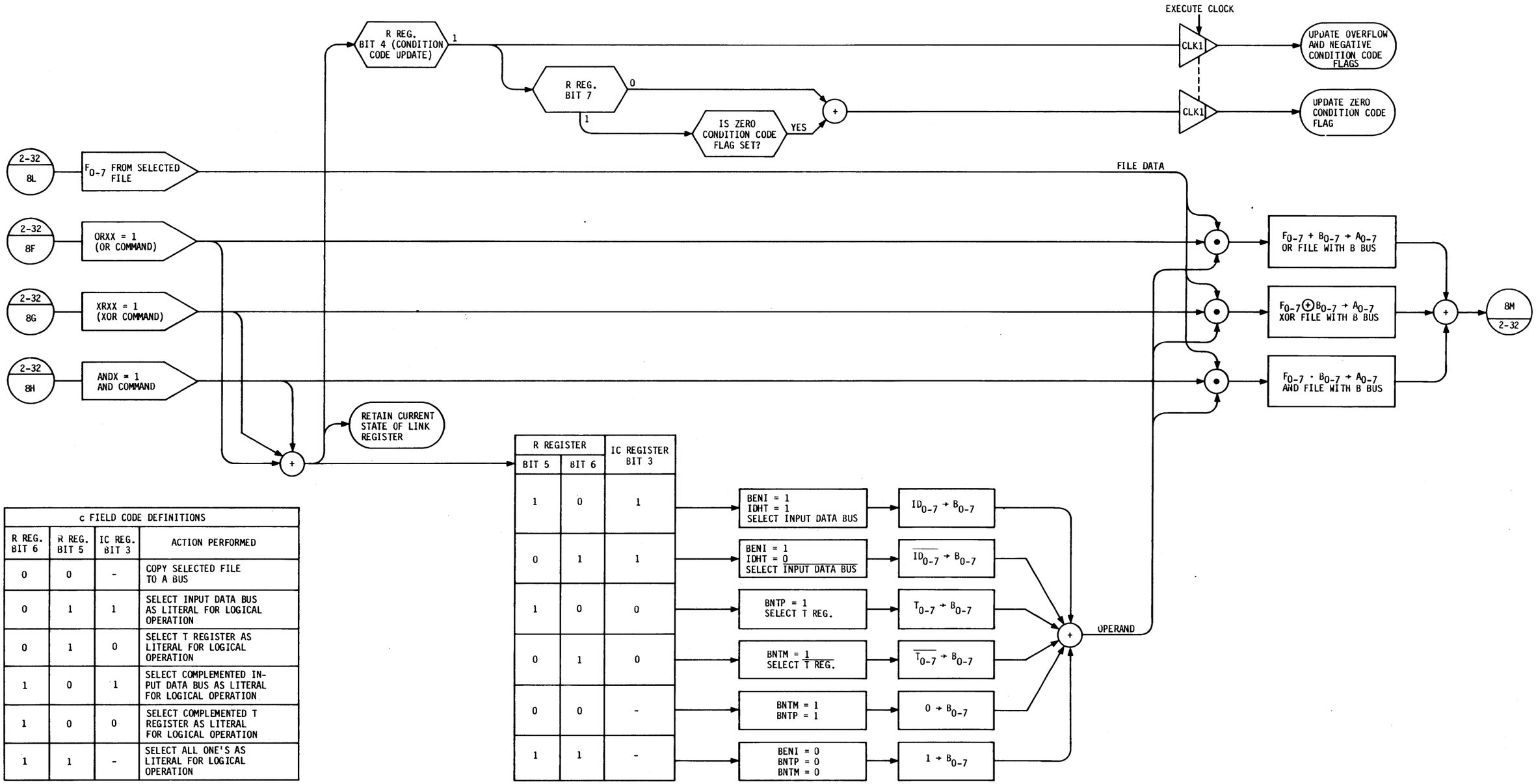
2.8.10.4 FULL-CYCLE MEMORY READ. For a full-cycle memory read operation, c field bits 5 and 4 contain code 00_2, and produce MRCY = 1 and MFCY = 1. Following the next CLK6 time, all ones are written into the memory location addressed by the M and N Registers. This causes the sense amplifiers to Data Register in the core memory to sense and store the data that was in the addressed location. Also, TCLR occurs,



c FIELD CODE DEFINITIONS

BIT 7	BIT 6	BIT 5	BIT 4	ACTION PERFORMED
-	-	0	0	FULL-CYCLE READ
-	-	1	0	HALF-CYCLE READ
-	-	0	1	FULL-CYCLE WRITE
-	-	1	1	HALF-CYCLE WRITE
0	0	-	-	COPY SELECTED FILE TO A BUS
0	1	-	-	DECREMENT SELECTED FILE AND COPY TO A BUS. IF r FIELD IS SELECTING M OR N REGISTER, ADD MEMORY LINK BIT TO DECREMENTED FILE
1	0	-	-	ADD LINK TO SELECTED FILE AND COPY TO A BUS
1	1	-	-	INCREMENT SELECTED FILE AND COPY TO A BUS

Figure 2-36. Core Memory Read or Write Execution



c FIELD CODE DEFINITIONS

R REG. BIT 6	R REG. BIT 5	IC REG. BIT 3	ACTION PERFORMED
0	0	-	COPY SELECTED FILE TO A BUS
0	1	1	SELECT INPUT DATA BUS AS LITERAL FOR LOGICAL OPERATION
0	1	0	SELECT T REGISTER AS LITERAL FOR LOGICAL OPERATION
1	0	1	SELECT COMPLEMENTED INPUT DATA BUS AS LITERAL FOR LOGICAL OPERATION
1	0	0	SELECT COMPLEMENTED T REGISTER AS LITERAL FOR LOGICAL OPERATION
1	1	-	SELECT ALL ONE'S AS LITERAL FOR LOGICAL OPERATION

Figure 2-37. Logical AND, OR, and Exclusive OR Execution

clearing the T Register in preparation for storing the memory read data. Following the next CLK6 (time t_3), the occurrence of TMCK causes the memory read data stored in the Data Register (in core memory) to be entered into the T Register. Also, the data stored in the Data Register (in core memory) is written back into the addressed memory location.

2.8.10.5 OPERATIONS ON SELECTED FILE. Following the occurrence of MYCS = 1, the operation performed on the selected file is determined by the states of c field bits 7 and 6.

c Field Bit 7	c Field Bit 6	Selected Link Bit*	Resulting State of Initial Carry (CYIA)	Resulting Operation Performed on File
1	1	-	1	Increment file and copy to A bus
0	0	-	0	Copy file to A bus
1	0	Memory or Arithmetic	Same as link bit	Add link to file and copy to A bus
0	1	Memory	Same as link bit	Decrement file, and add link bit
0	1	Arithmetic	0	Decrement file and copy to A bus

*Memory link bit is selected if r field is selecting M or N Register (MNRX = 1); otherwise, arithmetic link bit is selected.

2.8.10.6 UPDATING OF CONDITION CODES AND LINK REGISTER. No updating is performed.

2.8.11 LOGICAL AND, OR, AND EXCLUSIVE OR COMMAND EXECUTION (Figure 2-37)
Op code C is logical OR, and produces ORXX = 1. Op code D is Exclusive OR, and produces XRXX = 1. Op code E is logical AND, and produces ANDX = 1. For all three commands, the logical operation is performed on the selected operand from the B bus, and the contents of the selected file. The result is placed on the A bus.

2.8.11.1 SELECTION OF OPERAND ON B BUS. The selected operand on the B bus is determined by c field bits 5 and 6, in conjunction with IC register bit 3.

c Field Bit 6	c Field Bit 5	IC Register Bit 3	Selected Operand On B Bus
1	1	-	All ones
0	0	-	All zeros
1	0	1	Uncomplemented Input Data Bus Bits
1	0	0	Uncomplemented T Register Bits
0	1	1	Complemented Input Data Bus Bits
0	1	0	Complemented T Register Bits

2.8.11.2 UPDATING OF CONDITION CODES. Updating of condition code flags is performed identically to that described in Paragraph 2.8.8.3.

2.8.11.3 UPDATING OF LINK REGISTER. No updating is performed.

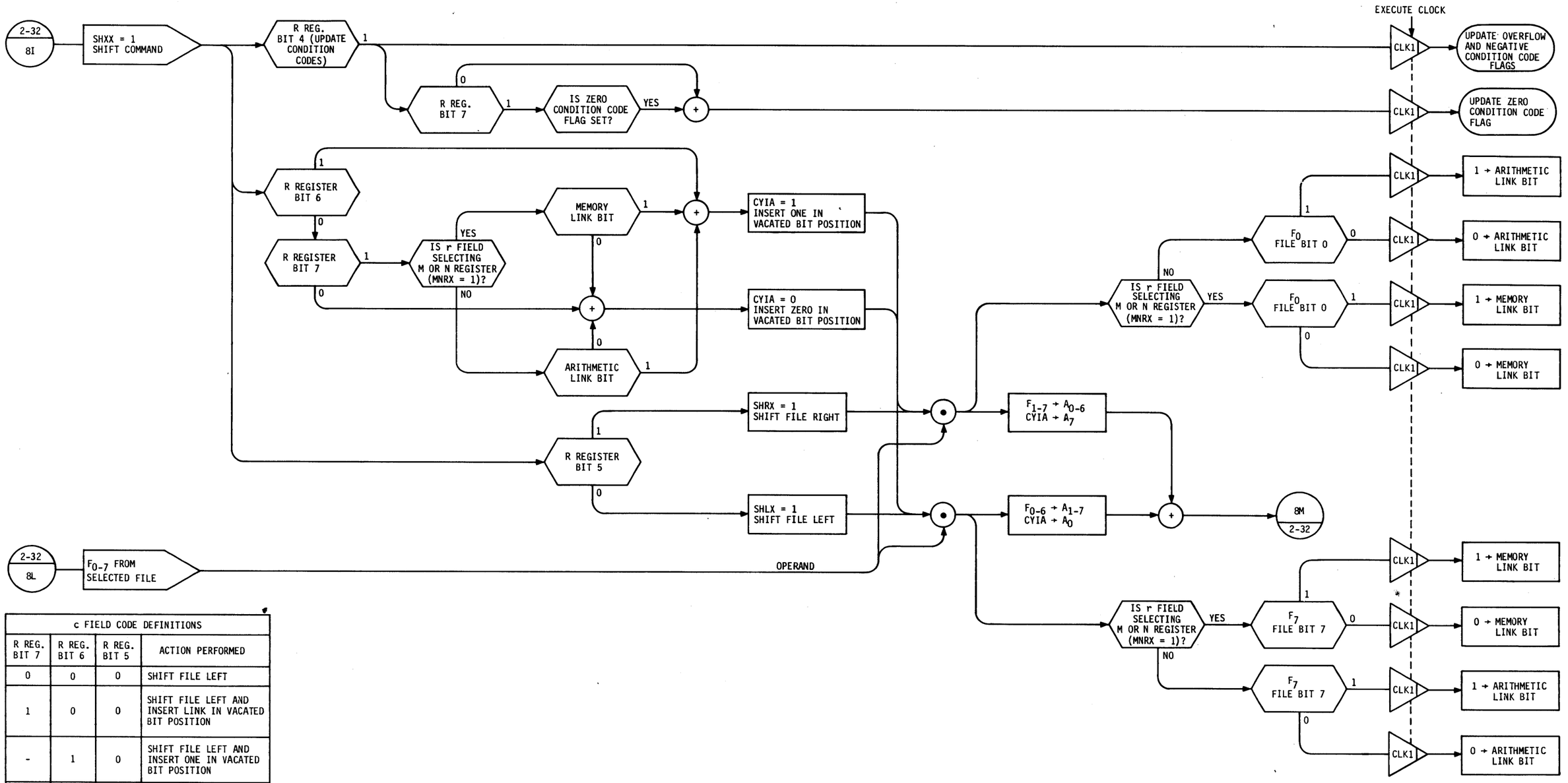
2.8.12 SHIFT RIGHT OR LEFT COMMAND EXECUTION (Figure 2-38)
Op code F is Shift, and produces SHXX = 1. The state of c field bit 5 determines whether a Shift Left (bit 5 = 0) or Shift Right (bit 5 = 1) operation is performed. For Shift Left, SHLX = 1 is produced. For Shift Right, SHRX = 1 is produced. The shift operation is performed on the contents of the selected file. The state of initial carry CYIA determines the state of the vacated bit position in the shifted file data. The Shift Left operation is performed by the ALU and the Shift Right operation is performed by the A bus multiplexer. The shifted data is always placed on the A bus.

2.8.12.1 INITIAL CARRY. The state of initial carry CYIA is determined by the states of c field bits 7 and 6.

c Field Bit 7	c Field Bit 6	Resulting State of CYIA
-	1	1
0	0	0
1	0	Same as link bit*

*Memory link bit is selected if r field is selecting M or N Register (MNRX = 1); otherwise, arithmetic link bit is selected.

2.8.12.2 CONDITION CODE UPDATING. Condition code updating is identical to that described in Paragraph 2.8.8.3.



c FIELD CODE DEFINITIONS			
R REG. BIT 7	R REG. BIT 6	R REG. BIT 5	ACTION PERFORMED
0	0	0	SHIFT FILE LEFT
1	0	0	SHIFT FILE LEFT AND INSERT LINK IN VACATED BIT POSITION
-	1	0	SHIFT FILE LEFT AND INSERT ONE IN VACATED BIT POSITION
0	0	1	SHIFT FILE RIGHT
1	0	1	SHIFT FILE RIGHT AND INSERT LINK IN VACATED BIT POSITION
-	1	1	SHIFT FILE RIGHT AND INSERT ONE IN VACATED BIT POSITION

Figure 2-38. Shift Right or Left Execution

2.8.12.3 LINK REGISTER UPDATING. The state of the out-shifted bit (F00 for shift right; F07 for shift left) is stored in the selected Link Reigster. When the r field is selecting the M or N Register (MNRX = 1), the memory link register is selected; otherwise, the arithmetic link register is selected.

SECTION 3

OPERATION

3.1 INTRODUCTION

This section describes the functions and capabilities of the System Front Panel. Included are descriptions of controls and indicators, and procedures for operating the CPU from the front panel. Additional information is provided in Section IV, Maintenance.

3.2 FRONT PANEL CONTROLS AND INDICATORS

Figure 3-1 identifies and describes the front panel controls and indicators and contains turn-on instructions.

3.3 PURPOSES AND CAPABILITIES OF FRONT PANEL MODES OF OPERATION

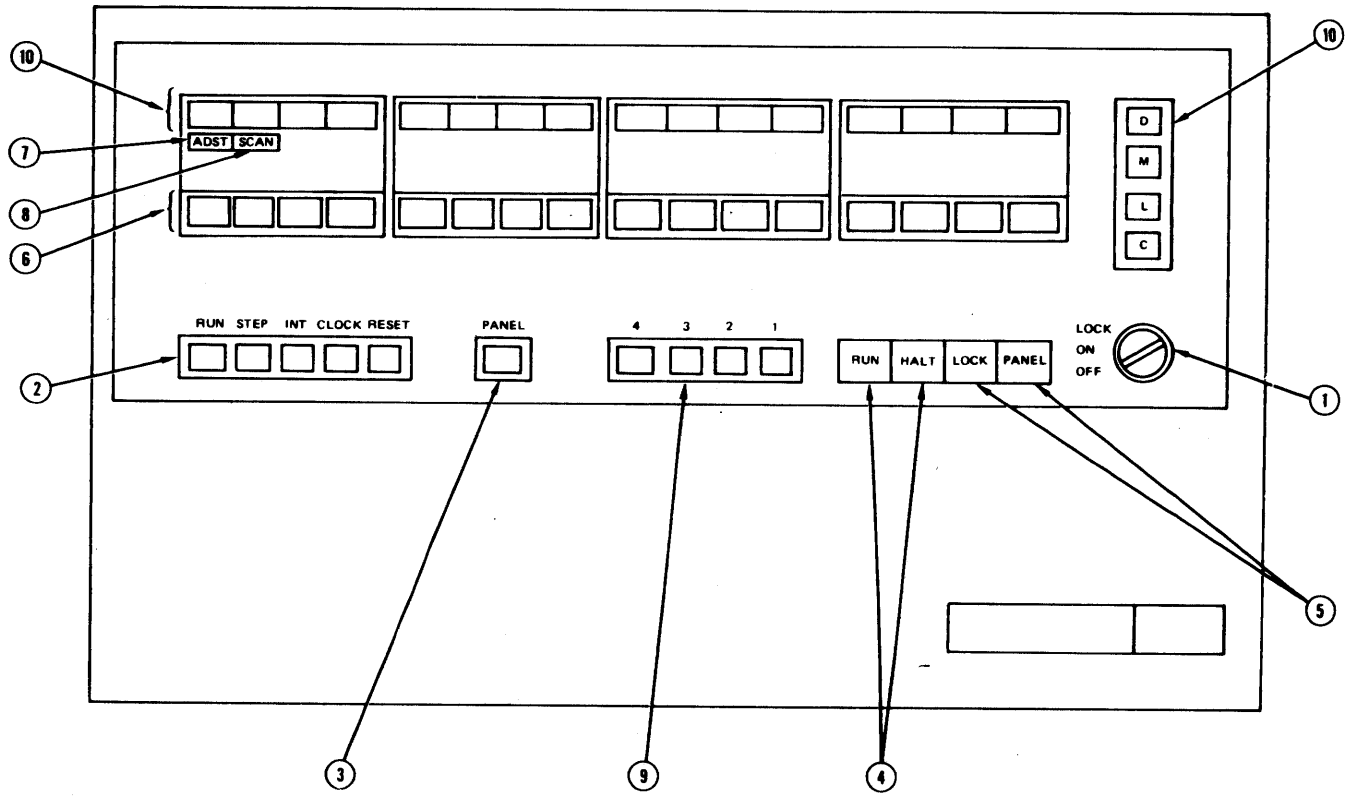
The front panel is capable of operating in a number of different modes. Each mode of operation and its purposes and capabilities are described in the following subparagraphs.

3.3.1 PANEL MODE

The PANEL mode is selected when the key-lock switch is set to ON and the PANEL switch is activated (PANEL indicator lights). Microcommand outputs of control memory are inhibited, and the source of the microcommands is the front panel data switches. Counting action in the L Register is inhibited, causing the L Register to retain the address that was present when the PANEL mode was selected (normally, the L Register is cleared before entering the PANEL mode by momentarily activating the RESET switch).

In the PANEL mode, the microcommand set into the data switches is entered into the R Register at each clock time, and the microcommand is decoded and executed. Although the STEP or INT switch may be used to place the CPU in the RUN state, either the CLOCK or RUN switch is normally used.

When the RUN switch is used, the CPU remains in the RUN state and repetitively executes the command entered by the data switches. In this case, the CPU enters the HALT state only when the CLOCK switch is activated, or when a HALT microcommand is entered by the data switches.



(1) KEY-LOCK SWITCH

This three-position, key-operated switch applies power to the CPU, and either enables or locks out the front panel controls. The key may be removed in any position. In the OFF position, power to the CPU is interrupted. In the ON position, power is applied, and control of the CPU from the front panel is enabled. In the LOCK position, power is applied to the CPU, and the machine state control switches and PANEL mode switch are locked out. With the switch in the LOCK position, the display select pushbuttons and data display are still enabled, and data entry is still possible via the sense switches (enter sense switches command) and the low-order data switches (Enter Console Switches Command).

(2) MACHINE STATE CONTROL SWITCHES

This group of momentary contact switches, located at the lower left side of the panel, are enabled whenever the key-lock switch is in the ON position. The switches are activated when placed in the down position. Each switch is described in the following paragraphs:

A. RESET Switch. When this switch is momentarily activated, a master reset pulse is produced, clearing the CPU, and placing it in the HALT state. The following reset actions occur:

a. The L, IC, and U Registers are cleared.
b. Panel, real time clock, and external interrupts are cleared.
c. External interrupts are disabled.
d. The real time clock is disabled.
e. Storage in the L Save Register is enabled (inhibit L Save is cleared).
f. The primary file is enabled.
g. All memory cycle control and machine state control flip-flops are reset.

B. CLOCK Switch. When the CPU is in the RUN state, momentary activation of the CLOCK switch produces a forced HALT following execution of the current microcommand. When the CPU is in the HALT state, momentary activation of the CLOCK switch places the CPU in the RUN state for execution of a single microcommand. When execution of the microcommand is completed, the CPU returns to the HALT state.

C. INT Switch. When the CPU is in the HALT state, momentary activation of the INT switch places the CPU in the RUN state, and a panel internal status interrupt is initiated. The panel interrupt is bit 0 of the internal status byte. Normally the firmware responds to a panel interrupt by executing an Enter Internal Status (EIS) microcommand that clears the panel interrupt.

D. STEP Switch. When the CPU is in the HALT state, momentary activation of the STEP switch places the CPU in the RUN state. As long as the switch is activated, a Step internal status interrupt occurs, and the CPU is forced to run until a HALT occurs. The step interrupt is bit 7 of the internal status byte. Normally, the firmware responds by executing an Enter Internal Status (EIS) microcommand, followed by execution of a single macro-instruction or subroutine. Upon completion of the macro-instruction, the firmware is normally microprogrammed to cause a processor HALT.

E. RUN Switch. When the CPU is in the HALT state, momentary activation of the RUN switch places the CPU in the RUN state. The CPU remains in the RUN state until a HALT microcommand is executed, or until the CLOCK switch is operated.

(3) PANEL MODE SWITCH

When the key-lock switch is in the ON position, placing the PANEL switch to the down position places the CPU in the PANEL mode until the PANEL switch is changed. In this mode, microcommand outputs from control memory are inhibited, and the CPU executes microcommands entered via the front panel data switches. When the switch is in the up position, the source of microcommands is the control memory.

Figure 3-1 (Sheet 1). Front Panel Controls and Indicators

(4) MACHINE STATE INDICATOR LIGHTS

Either the RUN or HALT indicator lights to indicate whether the CPU is in the RUN or HALT state.

(5) PANEL STATUS INDICATOR LIGHTS

When the key-lock switch is set to ON and the PANEL switch selects the PANEL mode, the PANEL indicator lights. When the key-lock switch is set to LOCK, the LOCK indicator lights.

(6) DATA SWITCHES

The function of the 16 data switches is determined by the mode of panel operation. Switch 15 (MSB) is located at the left side of the panel, and switch 0 (LSB) at the right side. A binary 1 is entered into a bit position when a data switch is in the down position. A binary 0 is entered when the switch is in the up position.

When the PANEL mode is selected, data switches are used to enter 16-bit microcommands into the CPU.

When the key-lock switch is set to ON, the PANEL mode is not selected, and data switch 15 is down, the ADDRESS STOP mode is selected, and data switches 11 through 0 are used to select a control memory stopping address.

When the key-lock switch is set to ON, the PANEL mode is not selected, and data switch 14 is down, the SCAN mode is selected. In this condition the control memory addresses are scanned sequentially, but the microcommands read from control memory are not executed. The ADDRESS STOP and SCAN modes can be selected simultaneously.

Data switches 11 through 0 can be used at any time to select a control memory address at which a sync pulse is produced for test purposes.

When an Enter Console Switches (ECS) microcommand is being executed and the PANEL mode is not selected, the eight lower data switches (7 through 0) can be used to modify the eight low-order bits of the microcommand that follows the ECS microcommand. Normally, the microcommand to be modified is a literal-type command that contains all 1's in the literal field. If a data switch is not set, corresponding bit of the literal field is changed from a 1 to a 0.

(7) ADDRESS STOP INDICATOR

When the ADDRESS STOP mode is selected by data switch 15 the ADST indicator lights.

(8) SCAN INDICATOR

When the SCAN mode is selected by data switch 14 the SCAN indicator lights.

(9) SENSE SWITCHES

Sense switches 4 through 1 are always enabled. When a switch is in the down position, a binary 1 is entered; a binary 0 is entered when the switch is in the up position. The states of the sense switches are entered into A bus bits 7 through 4 when an Enter Sense Switches (ESS) microcommand is executed. These switches are normally used to initiate special processing actions in the firmware or software.

(10) DISPLAY SELECT PUSHBUTTONS AND DATA DISPLAY

The four interlocked display select pushbuttons select the source of the data displayed in the 16 data lights. Only one pushbutton switch at a time may be pressed. Display indicator 15 (MSB) is at the left side of the panel, and display indicator 0 (LSB) is at the right side. When a display indicator lights, a binary 1 is indicated; when a display indicator goes off, a binary 0 is indicated. The four display select pushbutton switches, and the resulting data selected for display when each pushbutton is pressed, are as follows:

D - Data. This switch selects an 8-bit display of the data byte on the A bus. The data on the A bus is displayed in indicators 7 (MSB) through 0 (LSB).

M - Core Memory Address. This switch selects a 16-bit display of the current core memory address contained in the M and N Registers. Indicators 15 (MSB) through 8 (LSB) display the contents of the M Register; indicators 7 (MSB) through 0 (LSB) display the contents of the N Register.

L - Control Memory Address. This switch selects a 12-bit display of the current control memory address contained in the L Register. Indicators 11 (MSB) through 0 (LSB) display the L register contents.

C - Microcommand. This switch selects a display of the 16-bit microcommand on the RS bus. Indicator 15 is the MSB, and indicator 0 is the LSB. When the PANEL mode is not selected, or when the front panel is locked out, the source of the microcommand is the control memory. When the PANEL mode is selected, the source of the microcommand is the data switches.

POWER TURN-ON PROCEDURE

1. Assure power plug is connected to source of ac power, and that PWR switch (remote power supply only) is set to ON.
2. Set Key Lock switch (1) to ON.
3. Momentarily activate RESET switch.
4. Refer to initialization instructions for program to be run.
5. After initialization is completed, set Key Lock switch (1) to LOCK.

Figure 3-1 (Sheet 2). Front Panel Controls and Indicators

When the CLOCK switch is used, the CPU switches from the HALT to the RUN state when the CLOCK switch is activated, executes the command entered by the data switches, and returns to the HALT state. If a Return Jump microcommand (op code 1020, 1060, 10A0, or 19) is entered by the data switches, complete execution of the return jump is not performed since correct loading of the L Save address into the R Register cannot be accomplished, and the HALT occurs after one microstep. When a jump extended, jump in 1K or jump in 1/2K microcommand is entered by the data switches, the jump address is loaded into the L Register, and the HALT occurs after two microsteps. When a skip-type microcommand (op code 4, 5, or 6) is entered by the data switches and the skip is taken, the skip occurs before a HALT is produced (two microsteps required).

In the PANEL mode, the available data displays are as follows:

a. When the D display select pushbutton is pressed, the data on the A bus is displayed.

b. When the M display select pushbutton is pressed, the contents of the M and N Registers are displayed.

c. When the L display select pushbutton is pressed, the contents of the 12-bit L Register are displayed. If no jump command has been executed, the address in the L Register is the address that was present when the PANEL mode was entered. If a jump command has been executed, the last jump address is displayed.

d. When the C display select pushbutton is pressed, the microcommand entered by the data switches is displayed.

3.3.2 SCAN MODE

The SCAN mode is selected when the key-lock switch is set to ON, the PANEL switch is not activated (PANEL light is not lit), and data switch 14 is set to the down position (SCAN indicator lights). In the SCAN mode, the SCAN indicator lights, and the L Register sequentially scans all addresses in control memory. Storage of the accessed microcommands in the R Register is inhibited, thus no command execution is performed. Normally, the L Register is cleared before entering the SCAN mode by momentarily activating the RESET switch.

Although the STEP or INT switch may be used to place the CPU in the RUN state, either the CLOCK or RUN switch is normally used. When the RUN switch is used, the control memory addresses are scanned at the 5-MHz clock rate until a HALT is initiated by momentarily activating the CLOCK switch. When the CLOCK switch is used, the L Register advances one address each time the CLOCK switch is activated.

In the SCAN mode, the available data displays are as follows:

a. When the D display select pushbutton is pressed, an all-zero display is produced, indicating that R Register bits 7 through 0 have been forced to the zero state by the SCAN mode selection.

b. When the M display select pushbutton is pressed, the data contained in the M and N Registers when the SCAN mode was selected, are displayed.

c. When the L display select pushbutton is pressed, the contents of the 12-bit L Register are displayed.

d. When the C display select pushbutton is pressed, the microcommand in the control memory location addressed by the L Register is displayed, thus all locations in control memory may be examined in the SCAN mode.

3.3.3 ADDRESS STOP MODE

The ADDRESS STOP mode is selected when the key-lock switch is set to ON, the PANEL switch is not activated (PANEL light is not lit), and data switch 15 is set to the down position (ADST indicator lights). In the ADDRESS STOP mode, a stopping address is selected with data switches 11 through 0. The CPU is placed in the RUN mode, and a CPU HALT is initiated when the L Register contains the selected stopping address. After the HALT occurs, the L Register contains the stopping address plus one, except when the microcommand in the selected stopping address is one of the following:

a. If a jump extended (JE) jump in 1K (JP) microcommand is contained in the selected stopping address, the L Register contains the jump address after the CPU HALT occurs.

b. If an operate-type microcommand with code 4_8 or 5_8 in the r field is contained in the selected stopping address, the L Register contains the jump address developed by the operate-type command.

c. If a return jump microcommand is contained in the selected stopping address, the L Register contains the L Save (return) address after the CPU HALT occurs.

d. If a skip-type microcommand is contained in the selected stopping address and the skip is taken, the L Register contains the selected stopping address plus two after the HALT occurs.

No CPU HALT occurs in the ADDRESS STOP mode when the selected stopping address is one of the following:

a. The starting address of the microprogram, containing the microcommand that sets the CPU in the RUN mode, will not produce an address stop if the microprogram does not loop back through the starting address.

b. The selected stopping address is the next address beyond a location containing a jump or return jump microcommand.

Although the STEP, INT, or CLOCK switches may be used to place the CPU in the RUN state, the RUN switch is normally used. When the RUN switch is activated, the CPU enters the RUN state and the L Register begins accessing control memory locations at the normal 5-MHz rate. Accessing commences at the address that was in the L Register when the RUN state was entered. When the stopping address is reached in the L Register, the CPU HALT is initiated, conditioned by the rules just described.

In the ADDRESS STOP mode, the available data display after the address stop occurs are as follows:

a. When the D display select pushbutton is pressed, the data on the A bus are displayed.

b. When the M display select pushbutton is pressed, the last data entered into the M and N Registers before the address stop occurred are displayed.

c. When the L display select pushbutton is pressed, the contents of the 12-bit L Register are displayed. When the stopping address does not contain a jump, return jump or skip-type command, the address in the L Register is the selected stopping address plus one. When the stopping address contains a jump or return jump command, the L Register contains the jump or jump return (L Save) address. If the stopping address contains a skip-type command, and the skip is taken, the L Register contains the stopping address plus two.

d. When the C display select pushbutton is pressed, the 16-bit lmicrocommand in the control memory location, accessed by the address contained in the L Register, is displayed.

3.3.4 COMBINED SCAN AND ADDRESS STOP MODE

The combined SCAN and ADDRESS STOP mode is selected when the key-lock switch is set to ON, the PANEL switch is not activated (PANEL light is not lit), and data switches 15 and 14 are both in the down position (SCAN and ADST indicators light). A stopping address is selected with data switches 11 through 0. Normally, the L Register is cleared by momentarily activating the RESET switch while the CPU is still in the HALT state. The CPU is then placed in the RUN state by momentarily activating the RUN switch. Commencing at the starting address in the L Register, the L Register sequentially scans the control memory locations at the 5-MHz rate. Storage of the accessed microcommands in the R Register is inhibited, hence no commands are executed. When the L Register reaches the selected stopping address, the CPU unconditionally halts with the stopping address plus one in the L Register.

After the address stop occurs, the available data displays are as follows:

a. When the D display select pushbutton is pressed, an all-zero display is produced, indicating that R Register bits 7 through 0 have been forced to the zero state by the SCAN mode selection.

b. When the M display select pushbutton is pressed, the contents of the M and N Registers when the CPU entered the RUN mode are displayed.

c. When the L display select pushbutton is pressed, the selected stopping address plus one is displayed in indicators 11 through 0.

d. When the C display select pushbutton is pressed, the 16-bit microcommand in the selected stopping address plus one is displayed.

3.3.5 STEP MODE

When the key-lock switch is set to ON and the PANEL switch is not activated (PANEL light is not lit), the STEP mode is initiated by activating the STEP switch, which places the CPU in the RUN state. This mode requires firmware that is normally microprogrammed to respond to the resulting step interrupt by executing an Enter Internal Status (EIS) microcommand, followed by a firmware-chosen macro-instruction (routine or subroutine). Normally, the macro-instruction ends with a Halt microcommand, producing a CPU HALT.

If implemented by the firmware, the EIS microcommand may be followed by an Enter Console Switches (ECS), or Enter Sense Switches (ESS) microcommand. In this case, the low-order data switches (7 through 0) or sense switches must be set to the desired states before the STEP switch is activated.

3.3.6 PANEL INTERRUPT MODE

When the key-lock switch is set to ON and the PANEL switch is not activated (PANEL light goes off), the PANEL INTERRUPT mode is initiated by activating the INT switch, placing the CPU in the RUN state. This mode requires firmware that is normally microprogrammed to respond to the resulting panel interrupt by executing an Enter Internal Status (EIS) microcommand, followed by a firmware-chosen macro-instruction (routine or subroutine). Normally, the macro-instruction ends with a Halt microcommand, producing a CPU HALT.

If implemented by the firmware, the EIS microcommand may be followed by an Enter Console Switches (ECS) or Enter Sense Switches (ESS) microcommand. In this case,

the low-order data switches (7 through 0) or the sense switches must be set to the desired states before the INT switch is activated.

3.3.7 PANEL LOCKED-OUT MODE

The panel locked-out mode is always selected when the key-lock switch is set to the LOCK position. In this mode, the following front panel capabilities exist:

a. Display of any desired data may be selected by the display select pushbuttons.

b. Data is entered onto the A bus from the four sense switches each time a firmware Enter Sense Switches (ESS) microcommand is executed.

c. The states of the low-order data switches (7 through 0) can be used to modify the literal field (eight low-order bits) of the microcommand that follows execution of a firmware Enter Console Switches (ECS) microcommand. Normally, the literal field of the microcommand to be modified contains all one's, and the resulting literal entered into the R Register is identical to the literal entered into the low-order data switches.

d. Data switches 11 through 0 may be used to select any desired address in control memory. When the L Register contains this address, a 200-nanosecond positive pulse is produced at the LSYC test point on the System Front Panel Board. This pulse provides a convenient means for synchronizing oscilloscopes and time interval counters so the execution of a specific microcommand can be monitored. Execution of the microcommand at the selected address commences on the negative-going trailing edge of the LSYC pulse.

3.4 DATA ENTRY AND DISPLAY CAPABILITIES

In the PANEL mode (PANEL indicator lights), data may be entered into the CPU registers, and into core memory. In most cases, the data that was entered may be subsequently displayed (directly or indirectly) in the display indicators. There is no direct or indirect method for displaying the contents of the IC, MD and upper R Registers. However, the upper R Register will hold the same information as that present on the RS bus (selected for display by the C display select pushbutton) after the CPU is halted in the ADDRESS STOP or PANEL modes. The procedures in the following subparagraphs fully demonstrate the data entry and display capabilities. These data entry and display capabilities, and the applicable procedures, are summarized in Table 3-1.

TABLE 3-1. DATA ENTRY DISPLAY CAPABILITIES

DATA ENTRY AND DISPLAY	PROCEDURE (PARA NO.)
Enter and Display M Register Data	3.4.1
Enter and Display N Register Data	3.4.2
Enter and Display L Register Data	3.4.3
Enter and Display U Register Data	3.4.4
Enter and Display T Register Data	3.4.5
Enter and Display File Register Data	3.4.6
Enter and Display Output Data (OD) Register Data	3.4.7
Enter and Display Internal Status Data	3.4.8
Enter and Display File Zero Flag Data	3.4.9
Enter and Display Sense Switch Data	3.4.10
Enter and Display Memory Link Bit	3.4.11
Enter and Display Arithmetic Link Bit	3.4.12
Enter and Display Core Memory Data (single location)	3.4.13
Accessing Consecutive Core Memory Locations for Data Entry and Display	3.4.14
Enter and Display L Save Register Data	3.4.15
Enter and Display Lower R Register Data	3.4.16

3.4.1 ENTER AND DISPLAY M REGISTER DATA

To enter and display M register data, proceed as follows:

1. Select PANEL mode, and press M display select pushbutton.

2. Activate CLOCK switch to place CPU in HALT state.

3. Enter op code 12_{16} into data switches 15 through 8 (Load M).

4. Enter the data to be loaded into M Register into data switches 7 through 0.

5. Momentarily activate CLOCK switch. When CPU returns to HALT state, the states of display indicators 15 through 8 should correspond to the states of data switches 7 through 0.

3.4.2 ENTER AND DISPLAY N REGISTER DATA

To enter and display N Register data, proceed as follows:

1. Select PANEL mode, and press M display select pushbutton.

2. Activate CLOCK switch to place CPU in HALT state.

3. Enter op code 13_{16} into data switches 15 through 8 (Load N).

4. Enter the data to be loaded into the N Register into data switches 7 through 0.

5. Momentarily activate CLOCK switch. When CPU returns to HALT state, the states of display indicators 7 through 0 should correspond to the states of data switches 7 through 0. Display indicators 15 through 8 should go off, indicating that the M Register was cleared by the Load N command.

3.4.3 ENTER AND DISPLAY L REGISTER DATA

To enter and display L Register data, proceed as follows:

1. Select PANEL mode, and press L display select pushbutton.

2. Momentarily activate RESET switch. This action places the CPU in the HALT state, and clears the U and L Registers (the U Register must be cleared for proper entry of data into the L Register). Display indicators 11 through 0 should be off, indicating the L Register was cleared by the reset.

3. Enter op code 0_{16} into data switches 15 through 12 (jump extended).

4. Enter the data to be loaded into the L Register into data switches 11 through 0.

5. Momentarily activate CLOCK switch. When CPU returns to HALT state, the states of display indicators 11 through 0 should correspond to the states of data switches 11 through 0.

3.4.4 ENTER AND DISPLAY U REGISTER DATA

Data can be entered into the U Register by executing a Load U microcommand. The four low-order bits of the U Register can then be displayed by executing a jump extended microcommand with 0's in bits 11 through 8. The four high-order bits of the U Register cannot be displayed. To enter and display U register data, proceed as follows:

1. Select PANEL mode, and press L display select pushbutton.

2. Activate CLOCK switch to place CPU in HALT state.

3. Enter op code 16_{16} into data switches 15 through 8.

4. Enter the data to be loaded into the U Register into data switches 7 through 0. Record the settings of data switches 3 through 0.

5. Momentarily activate CLOCK switch. When CPU HALT occurs, proceed to step 6.

6. Enter op code 0000_{16} into data switches 15 through 0 (jump extended).

7. Momentarily activate CLOCK switch. When CPU returns to HALT state, display indicators 11 through 8 should be in the same states as the data switch 3 through 0 settings made in step 4. Display indicators 7 through 0 should all be off.

3.4.5 ENTER AND DISPLAY T REGISTER DATA

To enter and display T Register data, proceed as follows:

1. Select PANEL mode, and press D display select pushbutton.

2. Activate CLOCK switch to place CPU in HALT state.

3. Enter op code 11_{16} into data switches 15 through 8 (Load T).

4. Enter the data to be loaded into the T Register into data switches 7 through 0. Record the settings of data switches 7 through 0.

5. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 6.

6. Enter op code $B020_{16}$ into data switches 15 through 0 (copy T Register).

7. Momentarily activate CLOCK switch. When CPU returns to HALT state, the states of display indicators 7 through 0 should correspond to the data switch 7 through 0 settings that were made in step 4.

3.4.6 ENTER AND DISPLAY FILE REGISTER DATA

To enter and display file register data, proceed as follows:

1. Select PANEL mode, and press D display select pushbutton.

2. Activate CLOCK switch to place CPU in HALT state.

3. If entry and display for a register in the primary file is desired, enter op code 1040_{16} into data switches 15 through 0 (select primary file). If entry and display for a register in the secondary file is desired, enter op code 1080_{16} into data switches 15 through 0 (select secondary file).

4. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 5.

5. Enter op code 2_{16} into data switches 15 through 12 (load file with literal).

6. Enter the address (1_{16} through F_{16}) of the desired File Register into data switches 11 through 8.

7. Enter the data to be loaded into the selected File Register into data switches 7 through 0. Record these data switch settings.

8. Momentarily activate CLOCK switch. When CPU returns to the HALT state, proceed to step 9.

9. Enter op code C_{16} into data switches 15 through 12 (move file).

10. For data switches 11 through 8, retain the same settings that were made in step 6.

11. Enter code 00_{16} into data switches 7 through 0.

12. Momentarily activate CLOCK switch. When CPU returns to the HALT state, the states of display indicators 7 through 0 should be the same as the data switch 7 through 0 settings that were made in step 7.

3.4.7 ENTER AND DISPLAY OUTPUT DATA (OD) REGISTER DATA

To enter and display OD Register data, proceed as follows:

1. Perform steps 1 through 4 of the procedure in Paragraph 3.4.5. The data entered into the T Register in step 4 of this procedure will subsequently be entered into the OD Register.

2. Enter op code $1A00_{16}$ into data switches 15 through 0 (modify lower command).

3. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 4.

4. Enter op code $20FF_{16}$ into data switches 15 through 0 (load file with literal FF_{16}).

5. Momentarily activate CLOCK switch. When CPU returns to HALT state, the states of display indicators 7 through 0 should be the same as the data that was entered into the T Register in step 1.

3.4.8 ENTER AND DISPLAY INTERNAL STATUS DATA

In this procedure, real time clock and panel interrupts are entered into bits 0 and 2 of the internal status byte, and an Enter Internal Status microcommand is executed to place the internal status byte on the A bus. To enter and display internal status, proceed as follows:

1. Select PANEL mode and press M display select pushbutton.

2. Momentarily activate RESET switch to clear the interrupt flip-flop, and to disable the real time clock.

3. Enter op code 1720_{16} into data switches 15 through 0 (enable real time clock).

4. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 5.

5. Momentarily activate INT switch. Then, momentarily activate CLOCK switch to return the CPU to the HALT state.

6. Enter op code $704B_{16}$ into data switches 15 through 0 (enter internal status into N Register).

7. Momentarily activate CLOCK switch. When CPU returns to HALT state, display indicators 7 through 0 display the internal status byte. Indicators 0 and 2 should light. Indicators 1, 3, 4, 5, 6, and 7 should go off. Refer to Paragraph 2.4.4.1 for a description of the internal status byte format.

3.4.9 ENTER AND DISPLAY FILE ZERO FLAG DATA

To enter and display file zero flag data, proceed as follows:

1. Perform steps 1, 2, and 5 of the procedure in Paragraph 3.4.8. This procedure enters an internal status interrupt into bit 4 of the file zero flag byte.

2. Enter op code $C00B_{16}$ into data switches 15 through 0 (Enter file zero flags into N Register).

3. Momentarily activate CLOCK switch. When CPU returns to HALT state, display indicators 7 through 0 display the file zero flags byte. Indicator 4 should light. Indicators 0 through 2 display the states of the condition code flip-flops. Refer to Paragraph 2.4.4.2 for a description of the file zero flag byte format.

4. Momentarily activate RESET switch to reset the panel interrupt.

3.4.10 ENTER AND DISPLAY SENSE SWITCH DATA

To enter and display data entered by the sense switches, proceed as follows:

1. Select PANEL mode, and press M display select pushbutton.

2. Activate CLOCK switch to place CPU in the HALT state.

3. Set the four sense switches to the desired states.

4. Enter op code $701B_{16}$ into data switches 15 through 8 (enter sense switch data into N Register).

5. Momentarily activate CLOCK switch. When CPU returns to HALT state, display indicators 7 through 0 should display the sense switch data. Indicators 7 through 4 should be in the same states as sense switches 4 through 1. Indicators 3 through 0 should light.

3.4.11 ENTER AND DISPLAY MEMORY LINK BIT

This procedure enters a 1 into the memory link bit, and the link bit is then displayed. The procedure is as follows:

1. Perform steps 1 through 6 of the procedure in Paragraph 3.4.6 to set up for execution of a load file with literal command.

2. Enter code FF_{16} into data switches 7 through 0.

3. Momentarily activate CLOCK switch to enter the FF_{16} literal into the selected file register.

4. Enter op code 8_{16} into data switches 15 through 12 (Add), and retain data switches 11 through 8 in the same states that were established in step 1.

5. Enter code 43_{16} into data switches 7 through 0 (load file and N Register with incremented file data, and store high-order bit carry-out in memory link register).

6. Momentarily activate CLOCK switch. When CPU returns to HALT state, display indicators 7 through 0 should all go off.

7. Retain the previous data switch settings for data switches 15 through 8. Set code $8B_{16}$ into data switches 7 through 0 (add link to file and store in N Register).

8. Momentarily activate CLOCK switch. When CPU returns to HALT state, display indicator 0 should light, indicating that a 1 was stored in the memory link register in step 6. Display indicators 7 through 1 should go off.

3.4.12 ENTER AND DISPLAY ARITHMETIC LINK BIT

This procedure enters a 1 into the arithmetic link bit, and the link bit is then displayed. The procedure is as follows:

1. Perform steps 1 through 6 of the procedure in paragraph 3.4.6 to set up for execution of a load file with literal command.

2. Enter code FF_{16} into data switches 7 through 0.

3. Momentarily activate CLOCK switch to enter the FF_{16} literal into the selected file register.

4. Enter op code 8_{16} into data switches 15 through 12 (Add), and retain data switches 11 through 8 in the same states that were established in step 1.

5. Enter op code 41_{16} into data switches 7 through 0 (load file and T Register with incremented file data, and store high-order bit carry-out in arithmetic link registers).

6. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 7.

7. Press the D display select pushbutton.

8. Retain the previous data switch settings for data switches 15 through 8. Set code 89_{16} into data switches 7 through 0 (add link to file and store in T Register).

9. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 10.

10. Enter op code $B020_{16}$ into data switches 15 through 0 (copy T Register).

11. Momentarily activate CLOCK switch. When CPU returns to HALT state, display indicator 0 should light, indicating a 1 was stored in the arithmetic link register in step 6. Display indicators 7 through 1 should go off.

3.4.13. ENTER AND DISPLAY CORE MEMORY DATA (SINGLE LOCATION)

In this procedure, data is written into a single core memory location during a full-cycle memory write operation. Then, a half-cycle memory read operation is performed, and the data previously written into the memory is retrieved and stored in the T Register. Finally, the contents of the T Register are selected for display on the A bus. The procedure is as follows:

CAUTION

When performing this procedure, assure that the core memory location(s) into which data is written do not contain information or software required for normal operation.

1. Perform steps 1 through 3 of the procedure in Paragraph 3.4.5 to set up for a Load T operation.

2. Set data switches 7 through 0 to contain the data to be written into memory. Record the settings of data switches 7 through 0.

3. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 4.

4. Set op code 13_{16} into data switches 15 through 8 (Load N), and set data switches 7 through 0 to contain the low-order bits of the core memory location to be accessed for a write operation.

5. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 6.

6. Set op code 12_{16} into data switches 15 through 8 (Load M), and set data switches 7 through 0 to contain the high-order bits of the core memory location to be accessed for a write operation.

7. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 8.

8. Set op code $A010_{16}$ into data switches 15 through 0 (write memory full cycle).

9. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 10.

10. Set op code $A020_{16}$ into data switches 15 through 8 (read memory, half cycle).

11. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 12.

12. Set op code $B020_{16}$ into data switches 15 through 0 (copy T Register).

13. Momentarily activate CLOCK switch. When CPU returns to HALT state, the states of display indicators 7 through 0 should be the same as the data switch 7 through 0 settings that were made in step 2.

3.4.14 ACCESSING CONSECUTIVE CORE MEMORY LOCATIONS FOR DATA ENTRY AND DISPLAY

This procedure can be used for writing data into consecutive locations in core memory and then reading the data back from the same core memory locations for display. The procedure as given permits sequential accessing of up to 255 consecutive locations in core memory. Proceed as follows:

CAUTION

When performing this procedure, assure that the core memory location(s) into which data is written do not contain information or software required for normal operation.

1. Select PANEL mode, and press D display select pushbutton.

2. Momentarily activate CLOCK switch to place CPU in HALT state.

3. Enter op code 11_{16} into data switches 15 through 8, and enter the data to be written into memory into data switches 7 through 0. Record the settings of data switches 7 through 0.

4. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 5.

5. Enter op code 12_{16} into data switches 15 through 8 (Load M).

6. Select a 255-word block of core memory to be accessed, in which the eight low-order bits of the starting address contain code 00_{16}. Enter the eight high-order bits of the selected core memory starting address into data switches 7 through 0.

7. Momentarily activate CLOCK switch to enter the eight high-order address bits into the M Register. When CPU returns to HALT state, proceed to step 8.

8. Enter op code 2100_{16} into data switches 15 through 0 (load address 00_{16} into file register 1).

9. Momentarily activate CLOCK switch to enter code 00_{16} into file register 1.

10. Enter op code $A1D3_{16}$ into data switches 15 through 0. Commencing at the first core memory location selected in step 6, a new core memory location (up to 255) is accessed each time the clock switch is activated. The data entered in step 3 is written into each location. After data has been written into each location of interest by means of the CLOCK switch, proceed to step 11 to read the data back for display.

11. Repeat steps 8 and 9.

12. Enter op code $A1E3_{16}$ (half-cycle memory read) into data switches 15 through 0, and momentarily activate CLOCK switch.

13. Enter op code $B020_{16}$ into data switches 15 through 0 (copy T Register), and momentarily activate CLOCK switch. When CPU returns to HALT state, display indicators 7 through 0 should display the data that was entered in step 3.

14. Repeat steps 12 and 13 to read out and display the contents of each core memory location that was accessed in step 10.

3.4.15 ENTER AND DISPLAY L SAVE REGISTER DATA

This procedure involves the execution of a jump extended command, followed by execution of a return jump command. The address contained in the L Register when the jump extended command is executed is stored in the L Save Register. When the return jump command is executed, the contents of the L Save Register are returned to the L Register, and made available for display. The procedure is as follows:

1. Momentarily activate RESET switch to enable the L Save Register and to clear the L and U Registers.

2. With PANEL switch inactive (PANEL light is off), set data switches 14 and 15 to the up position to select the combined SCAN and ADDRESS STOP mode.

3. Press L display select pushbutton.

4. Enter any desired stopping address into data switches 11 through 0.

5. Momentarily activate RUN switch. When CPU returns to the HALT state, record the L address displayed in display indicators 11 through 0.

6. Select the PANEL mode.

7. Enter op code 0000_{16} into data switches 15 through 0 (jump extended to address 000).

8. Momentarily activate CLOCK switch. When CPU returns to HALT state, all display indicators should be off.

9. Set op code 1020_{16} into data switches 15 through 8 (Return).

10. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 11.

11. Enter op code $OFFF_{16}$ into data switches 15 through 0.

12. Momentarily activate CLOCK switch. When CPU returns to HALT state, display indicators 11 through 0 should display the address that was recorded in step 5.

3.4.16 ENTER AND DISPLAY LOWER R REGISTER DATA

To enter and display data in the lower R Register, proceed as follows:

1. Select PANEL mode, and press D display select pushbutton.

2. Activate CLOCK switch to place CPU in HALT state.

3. Enter op code 20_{16} into data switches 15 through 8 (load file with literal).

4. Enter the data to be loaded into the lower R Register into data switches 7 through 0.

5. Momentarily activate CLOCK switch. When CPU returns to HALT state, the states of display indicators 0 through 7 should correspond to the states of data switches 7 through 0.

3.5 EXECUTION AND DISPLAY OF LITERAL-TYPE ARITHMETIC AND LOGIC OPERATIONS

Literal-type arithmetic and logic operations (op codes 4, 5, and 6) can be executed and the result is made available for display on the A bus. To execute a literal-type arithmetic or logic operation, any desired literal is loaded into a file register (in the procedure that is given, primary File Register 1 is selected), and the literal to be tested or compared with the literal stored in file is entered into the literal-type microcommand. The arithmetic or logic operation is then executed, and

the result may be displayed in display indicators 7 through 0 by pressing the D display select pushbutton. The procedure is as follows:

1. Momentarily activate RESET switch to select primary file.

2. Select PANEL mode, and press D display select pushbutton.

3. Activate CLOCK switch to place CPU in HALT state.

4. Enter op code 21_{16} into data switches 15 through 8 (load primary File Register 1 with literal).

5. Enter the desired literal to be loaded into primary File Register 1 into data switches 7 through 0.

6. Enter op code 4 (Test If Zero), 5 (Test If Not Zero), or 6 (Compare) into data switches 15 through 12.

7. Enter op code 1_{16} into data switches 11 through 8 (select primary File Register 1).

8. Enter the literal to be tested or compared with the literal that was entered in step 5 into data switches 7 through 0.

9. Momentarily activate CLOCK switch. When CPU returns to HALT state, the result of the arithmetic or logic operation is displayed in display indicators 7 through 0. If op code 4 or 5 was entered in step 6, a logical AND is performed between the literal entered in step 5 and the literal entered in step 8, and the display contains one's in the bits where a 1-bit match existed. If op code 6 was entered in step 6, the literal entered in step 5 is added to the literal entered in step 8, and the display indicates the sum. If the sum is greater than 255, the arithmetic link register bit was set to 1. The state of the arithmetic link register bit can be checked by entering op code B080 into data switches 15 through 0, and momentarily activating the CLOCK switch. When the CPU returns to the HALT state, the state of the link bit is displayed in display indicator 0.

3.6 EXECUTION AND DISPLAY OF OPERATE-TYPE ARITHMETIC AND LOGIC OPERATIONS

Operate-type arithmetic and logic operations (op codes 7020, 8, 9, B, D, E, and F) can be executed, and the result is made available for display on the A bus. To execute an operate-type arithmetic or logic operation, any desired literal is loaded into a file register (in the procedure that is given, primary File Register 1 is selected), and any desired operand is entered into the T Register. The arithmetic or logic operation is then executed, and the result may be displayed in display indicators 7 through 0 by pressing the D display select pushbutton. The procedure is as follows:

1. Momentarily activate RESET switch to clear U Register and select primary file.

2. Select PANEL mode, and press D display select pushbutton.

3. Activate CLOCK switch to place CPU in HALT state.

4. Enter op code 21_{16} into data switches 15 through 8 (load primary File Register 1 with literal).

5. Enter the desired literal to be loaded into primary File Register 1 into data switches 7 through 0. Record the settings of data switches 7 through 0.

6. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 7.

7. If the arithmetic or logic operation to be performed has op code 8, 9, B, D, or E proceed to step 8. If the arithmetic operation to be performed has op code 7020 or F (shift file), proceed to step 12.

8. Enter op code 11_{16} into data switches 15 through 8 (Load T).

9. Enter the desired operand to be loaded into the T Register into data switches 7 through 0. Record the settings of data switches 7 through 0.

10. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 11.

11. Set data switch 5 to the up position to select the T Register. Do not change the setting of this switch in step 14.

12. Referring to Table 2-3 (op code 7020) or 2-4 (op codes 8, 9, B, D, E, or F), enter the op code for the arithmetic or logic operation to be performed into data switches 15 through 12.

13. Enter op code 1_{16} into data switches 11 through 8 (select primary File Register 1).

14. Referring to Table 2-4, enter the appropriate data into the c field with data switches 7 through 4.

15. Enter op code 8_{16} into data switches 3 through 0 (file inhibit, and no register selection).

16. Momentarily activate CLOCK switch. When CPU returns to HALT state, the result of the arithmetic or logic operation is displayed in display indicators 7 through 0. Verify that the result is correct by calculating the result of combining the literal entered in step 5 with the operand entered in step 9 (or the result of shifting the literal entered in step 5 for op code 7020 or F).

3.7 ENTRY OF MICROCOMMANDS VIA THE U AND OD REGISTERS

Any desired effective microcommand may be entered into the U and OD Registers. The eight high-order bits of the microcommand are entered into the U Register. The eight low-order bits are entered into the T Register, and are copied by the OD Register. Then, execution of a Modify Lower Command microcommand, followed by execution of an Execute command causes the contents of the U and OD Registers to be loaded into the R Register and decoded. By checking the resulting command decodes, correct entry of the effective microcommand into the R Register can be verified, permitting an indirect check of the U Register, the OD Register, and the R Register. The procedure is as follows:

1. Momentarily activate RESET switch to clear U and IC Registers.

2. Select PANEL mode.

3. Momentarily activate CLOCK switch to place CPU in HALT state.

4. Referring to Tables 2-2 through 2-4, select any desired effective microcommand except jump extended (op code 0), and write the states of the 16 bits of the selected microcommand.

5. Enter the eight high-order bits of the effective microcommand selected in step 4 into data switches 7 through 0.

6. Enter op code 16_{16} into data switches 15 through 8 (Load U).

7. Momentarily activate CLOCK switch. When CPU returns to HALT state, the U Register has been loaded with the eight high-order bits of the effective microcommand selected in step 4.

8. Enter the eight low-order bits of the effective microcommand selected in step 4 into data switches 7 through 0.

9. Enter op code 11_{16} into data switches 15 through 8 (Load T).

10. Momentarily activate CLOCK switch. When CPU returns to HALT state, the T and OD Registers have been loaded with the eight low-order bits of the effective microcommand selected in step 4.

11. Enter op code $1A00_{16}$ into data switches 15 through 0 (Modify Lower Command).

12. Momentarily activate CLOCK switch. When CPU returns to HALT state, proceed to step 13.

13. Enter op code $00FF_{16}$ into data switches 15 through 0 (Execute command with literal $0FF_{16}$ in bits 11 through 0).

14. Momentarily activate CLOCK switch. When CPU returns to HALT state, the R Register will contain the effective microcommand selected in step 4.

SECTION 4

MAINTENANCE

4.1 INTRODUCTION

The procedures and techniques described in this section expand the procedures given in Section 3, Operation, and relate these procedures to specific maintenance and testing actions. Test equipment tools and other materials required for maintenance include the following:

a. Oscilloscope, Tektronix Model 453 or equivalent
b. Digital Voltmeter, Data Technology Corporation DT-430 or equivalent
c. Extender Board
d. Card Puller
e. Electronic Tool Kit
f. Vacuum Cleaner
g. Cleaning Solvent (Methyl-Ethyl-Ketone, Freon, or equivalent)
h. Wiping Cloths

4.2 CHECKING COMMAND DECODES

The active command decodes for each microcommand in the Central Processing Unit (CPU) repertoire are given in Tables 2-2 through 2-4 (Section 2). To check the command decodes for any microcommand, proceed as follows:

1. Install the control and data boards on extenders to permit access to internal points.

2. Place PANEL switch to down position.

3. Activate CLOCK switch to place the CPU in the HALT state.

4. Referring to Tables 2-2 through 2-4, enter the code for the microcommand to be checked into data switches 15 through 0.

5. Momentarily activate CLOCK switch. When the CPU returns to the HALT state, the microcommand entered in step 4 is contained in the R Register, except when a return jump is executed. In this case, the R Register contains op code 0 (jump extended) and the L Save address.

6. To check the active command decodes for the microcommand entered in step 4, refer to Table 2-1 to locate the logic diagram sheet of origin for each command decode to be checked. Locate the command decode signal on the logic diagram, and select an appropriate test point where the signal can be checked. Connect a voltmeter or oscilloscope to the test point, and verify that the command decode signal is in the correct state.

4.3 CHECKING CORE MEMORY TIMING

Figures 2-21 and 2-22 (Section 2) show the core memory timing details for full- and half-cycle memory read and write operations. Table 2-4 shows the microcommand formats for memory read or write operations. The signals shown in Figures 2-21 and 2-22 can be located on logic diagrams 0220 (basic clocks), 0230 (gated clocks), and 0450 and 0451 (memory control). To check core memory timing, proceed as follows:

1. Install the control board on an extender to permit access to internal points.

2. Select the PANEL mode.

3. Activate CLOCK switch to place the CPU in the HALT state.

4. Referring to Table 2-4, enter the code for the desired memory read or write operation into data switches 15 through 0.

5. Momentarily activate the RUN switch. The CPU will repetitively execute the selected memory read or write cycle. A clock stop (CSTP/ = 0) will occur at clock time t_1 of each memory cycle. When the clock stop terminates, another memory cycle is executed.

6. Referring to Figure 2-21 (memory write cycle) or Figure 2-22 (memory read cycle) as applicable, select the timing signal to be checked, and locate the signal on the logic diagrams. Also, select and locate an appropriate signal for synchronizing the oscilloscope. Connect the oscilloscope to the selected test points, and observe the timing signal on the oscilloscope.

4.4 DYNAMIC CHECKS OF COMMAND EXECUTION

The LSYC test point on the System Front Panel Board provides a convenient means for synchronizing an oscilloscope or time interval counter in order to monitor command

execution at normal speeds. With the PANEL mode inactive (PANEL light is off), the address of a known microcommand is entered into data switches 11 through 0. The monitoring device is synchronized with the LSYC signal, and any signal of interest may be selected as the input to the monitoring device. Execution of the command at the selected address occurs on the negative-going trailing edge of the 200-nanosecond-wide LSYC pulse.

4.5 CLOCK CHECKS

The timing for the internal clocks in the CPU is shown in Figures 2-12, and 2-16 through 2-20 (Section 2). These clocks are all shown on logic diagrams 0220 and 0230. Correct clock timing can be checked with an oscilloscope. For the clocks shown in Figures 2-17 through 2-19, the appropriate microcommand must be entered by selecting the PANEL mode and entering the microcommand with the data switches. The clock can then be observed by momentarily activating the RUN switch to place the CPU in the RUN state.

4.6 L REGISTER TIMING CHECKS

The L Register timing for the various modes is shown in Figure 2-20 (Section 2). Timing for a specific mode can be checked by selecting the PANEL mode, entering the appropriate microcommand, and activating the RUN switch to place the CPU in the RUN state.

4.7 MACHINE STATE CONTROL TIMING CHECKS

Figures 2-13 through 2-15 (Section 2) are timing diagrams for the machine state control circuits shown on logic diagrams 0210 and 0211. The timing shown on these diagrams can be verified by operating the appropriate machine state control switches, and/or by selecting the PANEL mode and entering the appropriate microcommands.

4.8 DATA CHECKS

Table 3-1 (Section 3) lists the various data entry and display checks that are possible in the PANEL mode of operation. These checks can be used to verify correct operation of most registers in the CPU.

4.9 CONTROL MEMORY CHECKS

All locations in control memory can be checked in the SCAN mode (refer to Section 3 Paragraph 3.3.2).

4.10 REGISTER CHECKS

With a few exceptions, correct entry of data into the CPU registers can be checked by the front panel display indicators. Refer to Table 3-1 (Section 3) for the applicable procedure.

4.11 CHECKING ARITHMETIC AND LOGIC OPERATIONS

Correct execution of arithmetic and logic operations can be verified by performing the procedures in Section 3, Paragraphs 3.5 and 3.6.

4.12 CHECKING I/O DEVICES AND CONTROLLERS

Normally, diagnostics should be used to check out I/O devices and controllers. However, it is possible to check out some devices by executing external I/O commands with the front panel data switches. Table 2-3 (Section 2) lists the external I/O commands. Function commands and output data bytes may be entered into the OD Register by performing steps 1 through 5 of the procedure in Section 3, Paragraph 3.4.5.

The function command (entered into the T Register by setting the appropriate code into front panel data switches 7 through 0) consists of a device order code in bits 7 through 5, and the device address code in bits 4 through 0. After the function command has been entered into the OD Register, enter code 7090 (Control Out) into the front panel data switches, and activate the CLOCK switch. Then, enter code 7080 (Clear I/O Mode) into the front panel data switches and again activate the CLOCK switch. These actions should enter the function command into the I/O controller.

4.13 TROUBLESHOOTING

CAUTION

Always shut power off by turning the key-lock switch to OFF when removing or replacing printed circuit boards. Failure to observe this caution will cause damage to printed circuit boards.

The correct operation of the control memory and the core memory should be checked after eliminating the power supply as the source of the fault. The control memory is checked by operating the CPU in the SCAN mode (refer to Section 3, Paragraph 3.3.2). The core memory is checked by performing the procedures in Section 3,

Paragraph 3.4.14. If these elements are operating satisfactorily, and the fault is not associated with a specific I/O device, the fault may be assumed to be in the CPU (data or control boards). Since the front panel circuits are inactive during normal operation, it may be assumed that the front panel circuits do not cause a malfunction that appears during normal operation. If the CPU is suspected, the most effective and rapid troubleshooting technique is to replace the data board (slot J3) with a spare board known to be operating satisfactorily. If replacement of the data board does not correct the fault, the control board (slot J2) may be assumed to be defective and should be replaced.

4.14 PREVENTIVE MAINTENANCE

The unit does not require a rigid schedule of preventive maintenance. However, to insure correction operation of the system it is advisable to perform the following actions on a regular basis:

a. Conduct diagnostic tests.

b. Check that the +5-volt supply voltage is within ±5 percent of its nominal value. When checking this voltage, turn off all I/O devices and measure the voltage with a digital voltmeter having a full-scale accuracy of 3 percent.

c. General Cleaning. Vacuum, as necessary, inside the enclosure and especially the power supply and air vents. Heavy accumulation of dirt will hinder the circulation of cooling air. Occasional cleaning of the printed circuit boards, using Methyl-Ethyl-Ketone, Freon, or equivalent, may be advisable.

4.15 MARGINAL VOLTAGE CHECKING

Checking operation of the computer when the voltages are marginal can often detect incipient faults before they actually cause a malfunction. The technique involves varying the power supply voltages while the computer is being exercised by a diagnostic program. To perform the marginal voltage check, proceed as follows:

1. Referring to Figure 4-1, connect a digital voltmeter to the appropriate terminals on terminal board TB1 of the power supply, and adjust the corresponding potentiometer (R25 for +5 volts, R15 for +12 volts, and R5 for -16.75 volts) to produce the voltage values given for each of the tests listed in Table 4-1.

2. For each of the tests in Table 4-1, perform one pass of the diagnostic. If an error occurs, refer to your diagnostic reference list to analyze the fault.

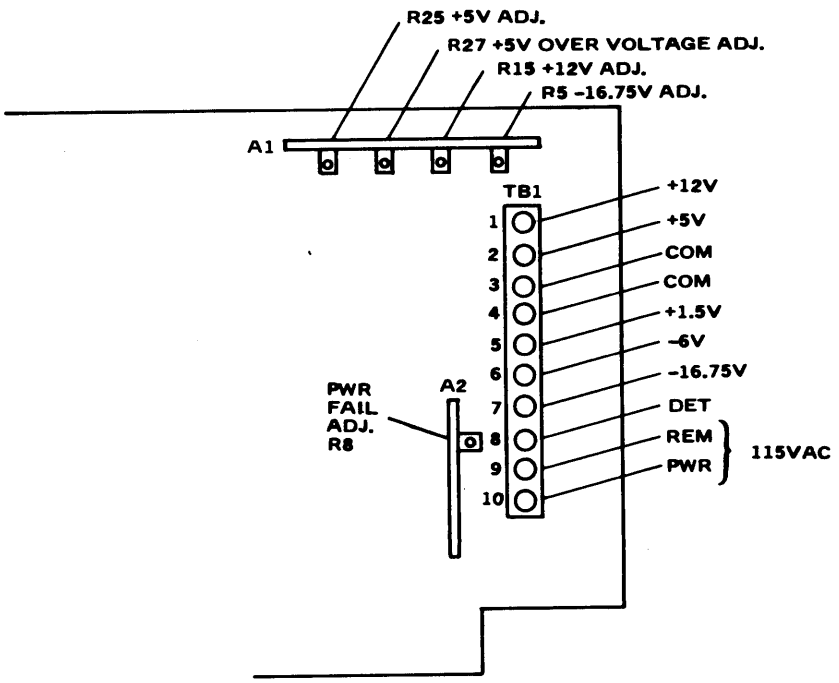


Figure 4-1. Remote Power Supply Voltage Adjustments and Test Points

TABLE 4-1. MARGINAL VOLTAGE TESTS

TEST NO.	ADJUSTMENTS		
	ADJUST R25 (+5 V)	ADJUST R15 (+12 V)	ADJUST R5 (-16.75 V)
1	+5.25 V	+12.60 V	-17.09 V
2	+5.25 V	+12.60 V	-16.41 V
3	+5.25 V	+11.40 V	-17.09 V
4	+5.25 V	+11.40 V	-16.41 V
5	+4.75 V	+12.60 V	-17.09 V
6	+4.75 V	+12.60 V	-16.41 V
7	+4.75 V	+11.40 V	-17.09 V
8	+4.75 V	+11.40 V	-16.41 V

4.16 ADJUSTMENTS

4.16.1 POWER SUPPLY ADJUSTMENTS

The power supply adjustments and test points for the Remote Power Supply unit are shown in Figure 4-1. Power supply adjustments and test points for the Integral be checked and adjusted as necessary on a regular basis. Use a digital voltmeter having a full-scale accuracy of 3 percent or better when measuring voltages. To check and adjust the power supply voltages, proceed as follows:

CAUTION

The +5 volt overvoltage potentiometer (R27) and the power fail adjustment (R8) are factory adjusted, and readjustment should not be attempted in the field.

Note

Although the power supply voltages can be monitored at terminal board TB1 of the power supply, the voltages should be adjusted while they are being monitored on the backplane for optimum performance.

1. Refer to Table 4-2, and connect the digital voltmeter to the indicated backplane test points for each voltage. Always check and adjust the voltages in the sequence given in Table 4-2.

2. If the voltage indicated on the digital voltmeter is not within the acceptable check range as indicated in Table 4-2, proceed to step 3. If the voltage is within the acceptable check range, no further action is required.

3. Adjust the potentiometer as indicated in Table 4-2 to obtain a voltage indication within the acceptable adjustment range.

TABLE 4-2. POWER SUPPLY ADJUSTMENTS

VOLTAGE	ADJUST-MENT	TEST POINTS ON BACKPLANE (POWER SUPPLY SLOT)	ACCEPTABLE CHECK RANGE	ACCEPTABLE ADJUSTMENT RANGE
+5 V	R25	A13 (-) to A26 (+)	+4.75 to +5.25	+4.90 to +5.10
+12 V	R15	A13 (-) to A06 (+)	+11.40 to +12.60	+11.90 to +12.10
-16.75 V	R25	A13 (+) to A09 (-)	-16.41 to -17.09	-16.70 to -16.80

4.16.2 WRITE INHIBIT ONE-SHOT ADJUSTMENT

The width of the write inhibit one-shot pulse is adjustable by means of potentiometer R57 on the control board. The timing for this pulse is shown in Figures 2-21 and 2-22 (Section 2). A dual-trace oscilloscope is required for adjustment. The adjustment should be performed whenever the control board is replaced, or when memory timing problems are suspected. To perform the adjustment, proceed as follows:

1. Place key-lock switch at ON, and select PANEL mode.

2. Set the oscilloscope for negative sync internal triggering of the trace. Set sweep for 500 nanoseconds, or 50 nanoseconds per centimeter.

3. Connect the trace 1 input lead of the oscilloscope to pin J2-B20 (RTXX/) of the backplane.

4. Connect the trace 2 input lead of the oscilloscope to pin J2-A21 (WTXX/) of the backplane.

5. Enter code 1300_{16} (Lead N) into the front panel data switches and activate the CLOCK switch. This action clears the M and N Registers.

6. Enter code $A000_{16}$ (read memory, full cycle) into the front panel data switches and activate the RUN switch. Full-cycle memory read operations should be performed repetitively, and waveforms should appear on the oscilloscope.

7. Examine the waveforms on the oscilloscope. The negative-going leading edge of the WTXX pulse on trace 2 shall occur 400 ±10 nanoseconds after the occurrence of the negative-going leading edge of the RTXX pulse on trace 1 as measured at the +1.5-volt points on both waveforms. If the WTXX leading edge occurs too early or too late, adjust R57 for the correct relationship.

8. Remove all test leads, set key-lock switch to OFF.

SECTION 5

PARTS LIST

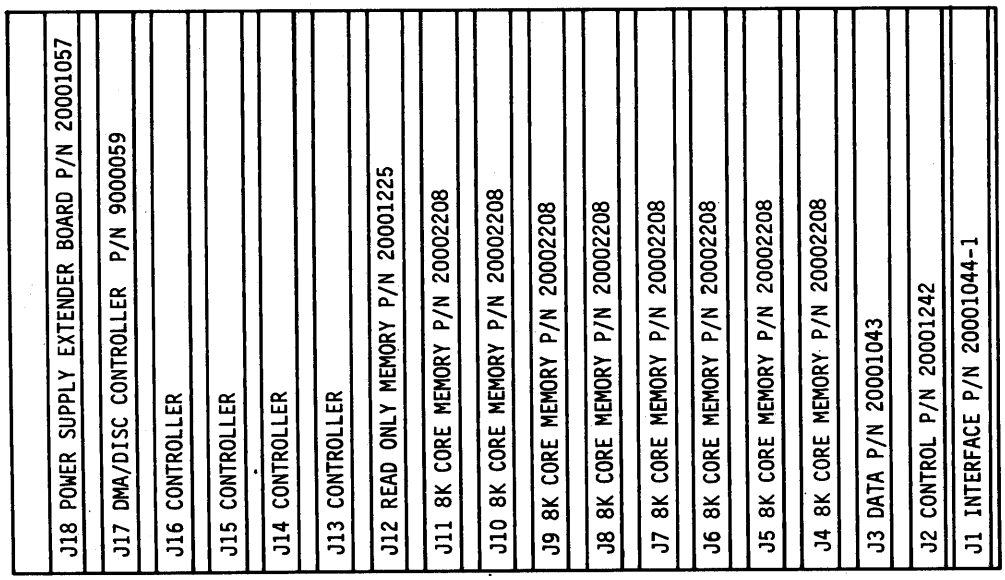
5.1 GENERAL

This section contains parts location diagrams and parts lists for the major assemblies of the Central Processing Unit.

The locations available in the CPU for the different printed circuit cards are shown below for a machine with a 64K memory. For a CPU with a 24K memory, the Read Only Memory will be in J7 and the first controller will be in J8. The DMA/Disc Controller is always in the highest numbered jack used for controllers and the Power Supply Extender Board is always in J18. There is a standard priority sequence for controllers whenever there are two or more controllers in the CPU, as follows:

Eight-Channel Controller, P/N 20002612
Four-Channel Controller, P/N 20002611-1 or P/N 20002612-1
Single Channel Controller, P/N 813002 or P/N 20002610
Magnetic Tape Controller, CPU Interface, P/N 10001124] both required.
Magnetic Tape Controller, MTU Interface, P/N 10001120]
Paper Tape Controller, P/N 400025 or P/N 400108
Card Reader Controller, P/N 400021 or P/N 900350
Printer Controller, P/N 900062 or P/N 900729

For any number of controllers selected, the controller nearest the top must be placed in the lowest number jack available to controllers and the next controller down the list in the next higher numbered jack.



5.2 SYSTEM PANEL P.C. CARD PARTS LIST (FIGURE 5-1)

Reference Designator	Part Number	Nomenclature	Qty
C1 through C18,C23,C24, C25,C26	TG-S10	Capacitor, 0.01 µf, 100V	22
C19,C20,C21,C22,C27,C29, C32,C34,C36,C38,C40	150D226X0015B2	Capacitor, 22 µf, 15V	11
C28,C30,C31,C35,C37,C39	DL46BU1042	Capacitor, 0.1 µf, 100V	7
DS1 through DS22	OL330BP	Lamp	22
J1,J2	3433-1002	Header, 50 pin (connector)	2
Q1 through Q22	2N2369A	Transistor	22
R1,R2,R5,R7,R9,R11,R13, R15,R17,R19,R21,R23, R25,R27,R29,R31,R33, R91,R93,R95,R97	RC07GF330J	Resistor, 33 ohm, 1/4W, ±5%	22
R2,R4,R6,R8,R10,R12,R14, R16,R18,R20,R22,R24, R26,R28,R30,R32,R34, R36 through R90, R92, R94, R96, R98 through R105, R109, R111, R112	RC07GF102J	Resistor, 1K, 1/4W, ±5%	85
R106,R107,R108	RC07GF471J	Resistor, 470 ohm, 1/4W, ±5%	3
R110	RC07GF561J	Resistor, 560 ohm, 1/4W, ±5%	1
S1,S2	D20001055	Switch, special	2
S3 through S16, S22, S23, S24,S25,S26	8931K253	Switch, SPDT	19
S17,S18,S19,S20,S21	8931K254	Switch, SPDT (momentary)	5
S27	277179-A1	Switch, key lock	1
Z1,Z2,Z10,Z12,Z18,Z34	SN7438N	Integrated circuit	6
Z3,Z4,Z7,Z8,Z15	SN7404N	Integrated circuit	5
Z5,Z6,Z11	SN7417N	Integrated circuit	3
Z9,Z16,Z33	SN7400N	Integrated circuit	3
Z17,Z22,Z23	SN7408N	Integrated circuit	3
Z20	SN74H05N	Integrated circuit	1
Z21,Z29,Z30	SN7485N	Integrated circuit	3
Z27	SN74H103N	Integrated circuit	1
Z28	SN74H11N	Integrated circuit	1

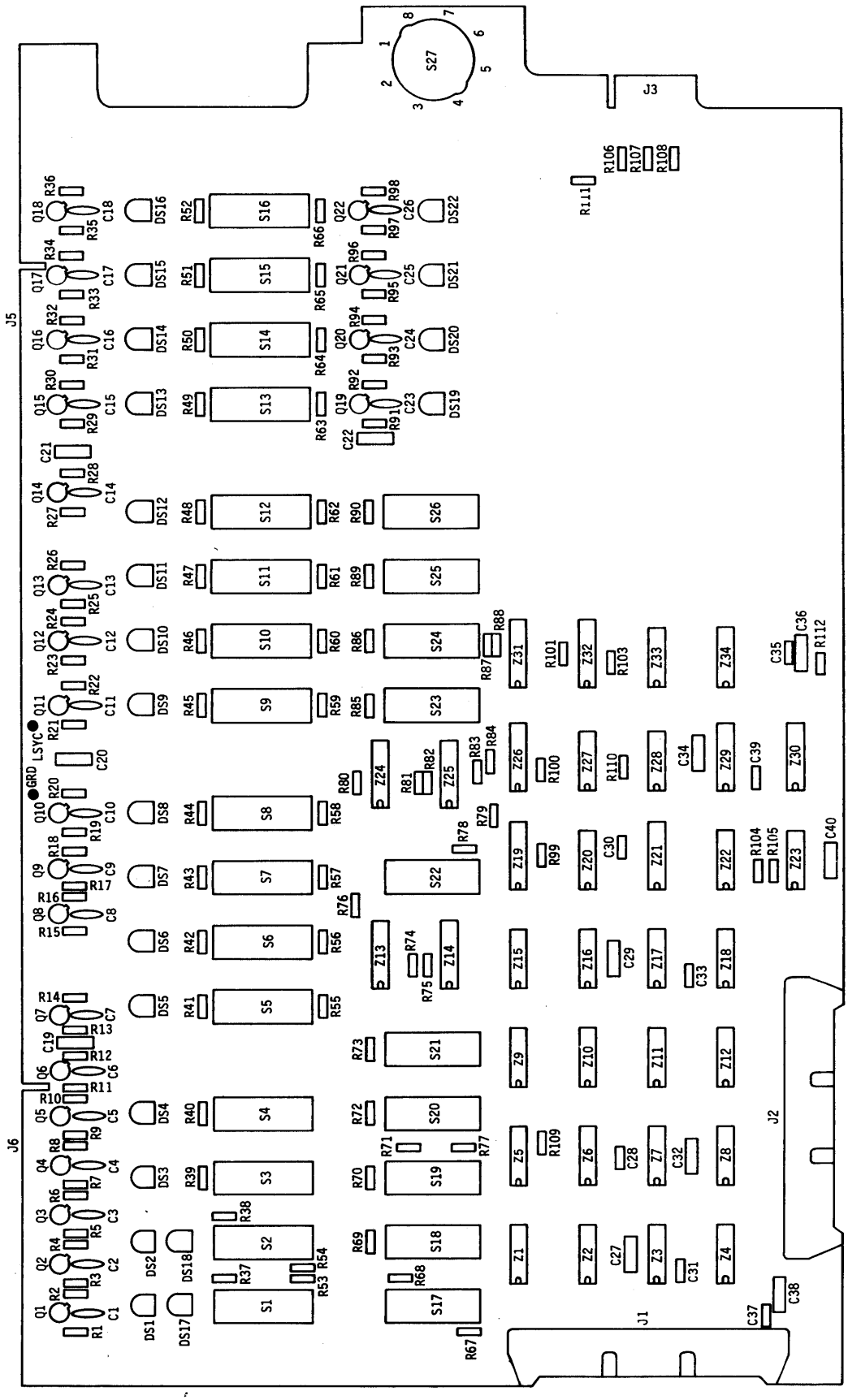


Figure 5-1. System Panel P.C. Card

5.3 BACKPLANE ASSEMBLY PARTS LIST (FIGURE 5-2)

Reference Designator	Part Number	Nomenclature	Qty
C1 through C5	CK06BX104M	Capacitor, 0.1 µf	5
CR1 through CR10	FDH600	Diode	10
R1,R18,R27,R32,R40	RC20GF390J	Resistor, 39 ohm, 1/2W, ±5%	5
R2,R3,R4,R5,R7,R9,R11, R12,R13,R16,R17,R19, R21,R23,R24,R25,R28, R30,R33 through R39	RC07GF121J	Resistor, 120 ohm, 1/4W, ±5%	25
R6,R8,R10,R20,R22,R26, R29,R31	RC07GF331J	Resistor, 330 ohm, 1/4W, ±5%	8
R41,R42,R43	RC07GF471J	Resistor, 470 ohm, 1/4W, ±5%	3

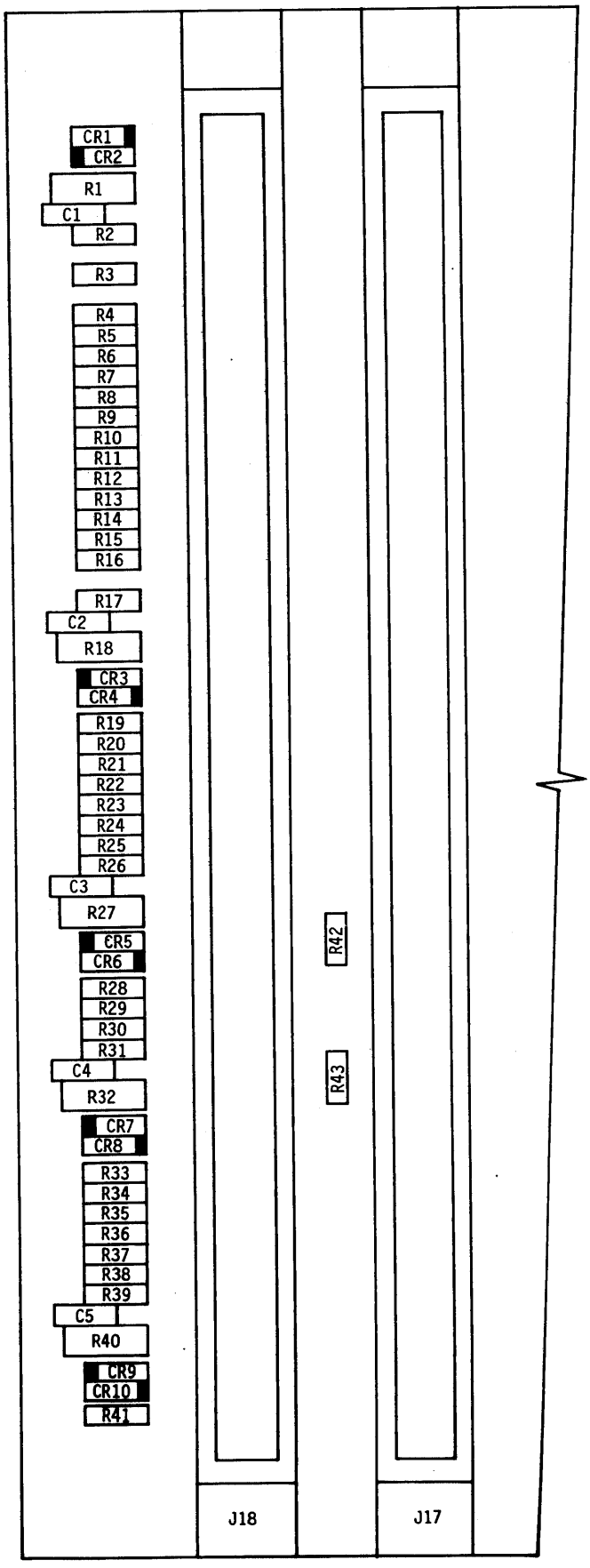


Figure 5-2. Backplane Assembly

5.4 INTERFACE CARD WITH INTEGRAL TTY PARTS LIST (FIGURE 5-3)

Reference Designator	Part Number	Nomenclature	Qty
C3,C9	150D475X0010A2	Capacitor, 4.7 µf, 10V	2
C4,C5,C6,C7,C8	TG-S10	Capacitor, 0.01 µf, 100V	5
C10	T362A684M035AS	Capacitor, 0.68 µf, 35V	1
CR2	1N759	Diode	1
CR3,CR4	FDH600	Diode	2
J2/P2	A20001066-16	Cable	1
J3/P1	A20001066-12	Cable	1
R7	RC07GF202J	Resistor, 2K, 1/4W, ±5%	1
R8,R13	RC07GF221J	Resistor, 220 ohm, 1/4W, ±5%	2
R9	RC42GF151J	Resistor, 150 ohm, 2W, ±5%	1
R10	RC07GF562J	Resistor, 5.6K, 1/4W, ±5%	1
R12	RC07GF471J	Resistor, 470 ohm, 1/4W, ±5%	1
Q3	2N2369A	Transistor	1
Z1	SN74L04N	Integrated circuit	1
Z2,Z3,Z11	SN7404N	Integrated circuit	3
Z4,Z9,Z14	SN7438N	Integrated circuit	3
Z5	SN7410N	Integrated circuit	1
Z6	SN7400N	Integrated circuit	1
Z7	SN7430N	Integrated circuit	1
Z8	TR1402A	Integrated circuit	1
Z10,Z15	SN7442N	Integrated circuit	2
Z12	SN7402N	Integrated circuit	1
Z13,Z19	SN74107N	Integrated circuit	2
Z16,Z17,Z18	SN74161N	Integrated circuit	3

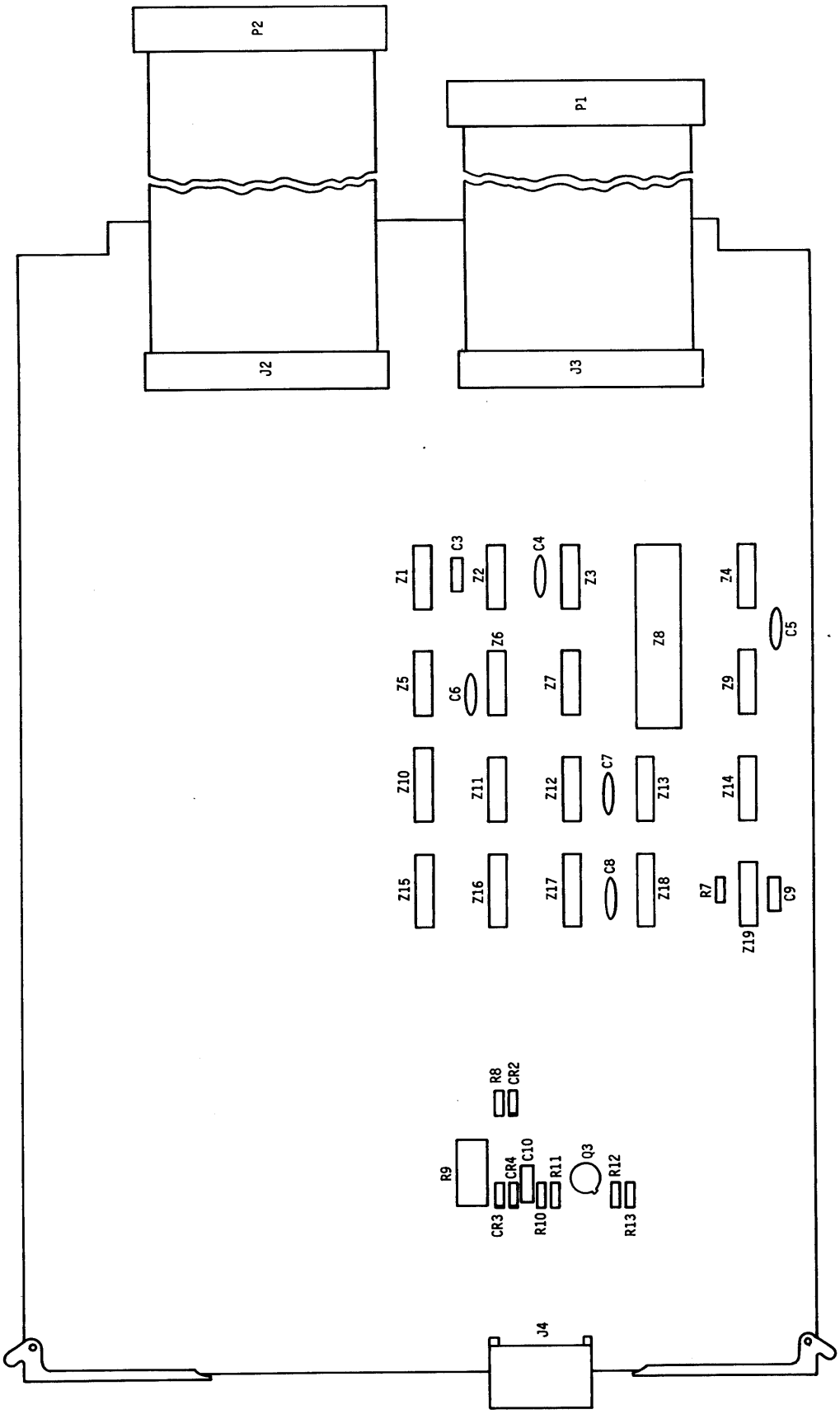


Figure 5-3. Interface Card with Integral TTY

5.5 DATA CARD PARTS LIST (FIGURE 5-4)

Reference Designator	Part Number	Nomenclature	Qty
C1,C3,C5,C7,C9,C12,C13, C16,C18,C19,C21,C23	150D226X0015B2	Capacitor, 22 µf, 15V	12
C2,C4,C6,C8,C10,C11, C14,C15,C17,C20,C22, C24	CK06BX104K	Capacitor, 0.1 µf	12
J2,J3	3433-1002	Connector	2
R1 through R9	RC07GF471J	Resistor, 470 ohm, 1/4W, ±5%	9
Z1,Z3,Z4,Z9,Z10	CS20000129-3	Resistor module	5
Z2,Z4,Z8,Z11	CS20000129-1	Resistor module	4
1A,1B,1D,1E,1H,1J,1L, 4E,4F,8E,8F,9A	SN7438N	Integrated circuit	12
1C	SN7437N	Integrated circuit	1
1F,1K,8H,8L	SN74H05N	Integrated circuit	4
2A,2B,3A	SN74161N	Integrated circuit	3
2C,3D,3E,3H	SN74H08N	Integrated circuit	4
2D,3C,3F,4L,5F,7C,8B,8K	U6N930859X	Integrated circuit	8
2E,2F,7A	SN74H04N	Integrated circuit	3
2H,2J,2K,2L,6B	SN74H51N	Integrated circuit	5
3B	U6B932259X	Integrated circuit	1
3J,3K,3L,4A,4B,4H,5C,5D	SN74H74N	Integrated circuit	8
4C,4D	SN74S00N	Integrated circuit	2
4J,5J,6A	SN7486N	Integrated circuit	3
5A,5B,5H,6C,6D,6E,6F,6H	SN74H53N	Integrated circuit	8
5L,6K	SN74181N	Integrated circuit	2
6L,7D	SN74H10N	Integrated circuit	2
7E,9J	SN74H00N	Integrated circuit	2
7F	SN74H20N	Integrated circuit	1
7H,7J,7K,7L	P3101A	Integrated circuit	4
8C,9L	SN74H30N	Integrated circuit	2
8D	SN74H103N	Integrated circuit	1
9B	SN74H22N	Integrated circuit	1
9C,9D,9E,9H	U6B930959X	Integrated circuit	4
9F	SN74S04N	Integrated circuit	1
9K	SN74H11N	Integrated circuit	1

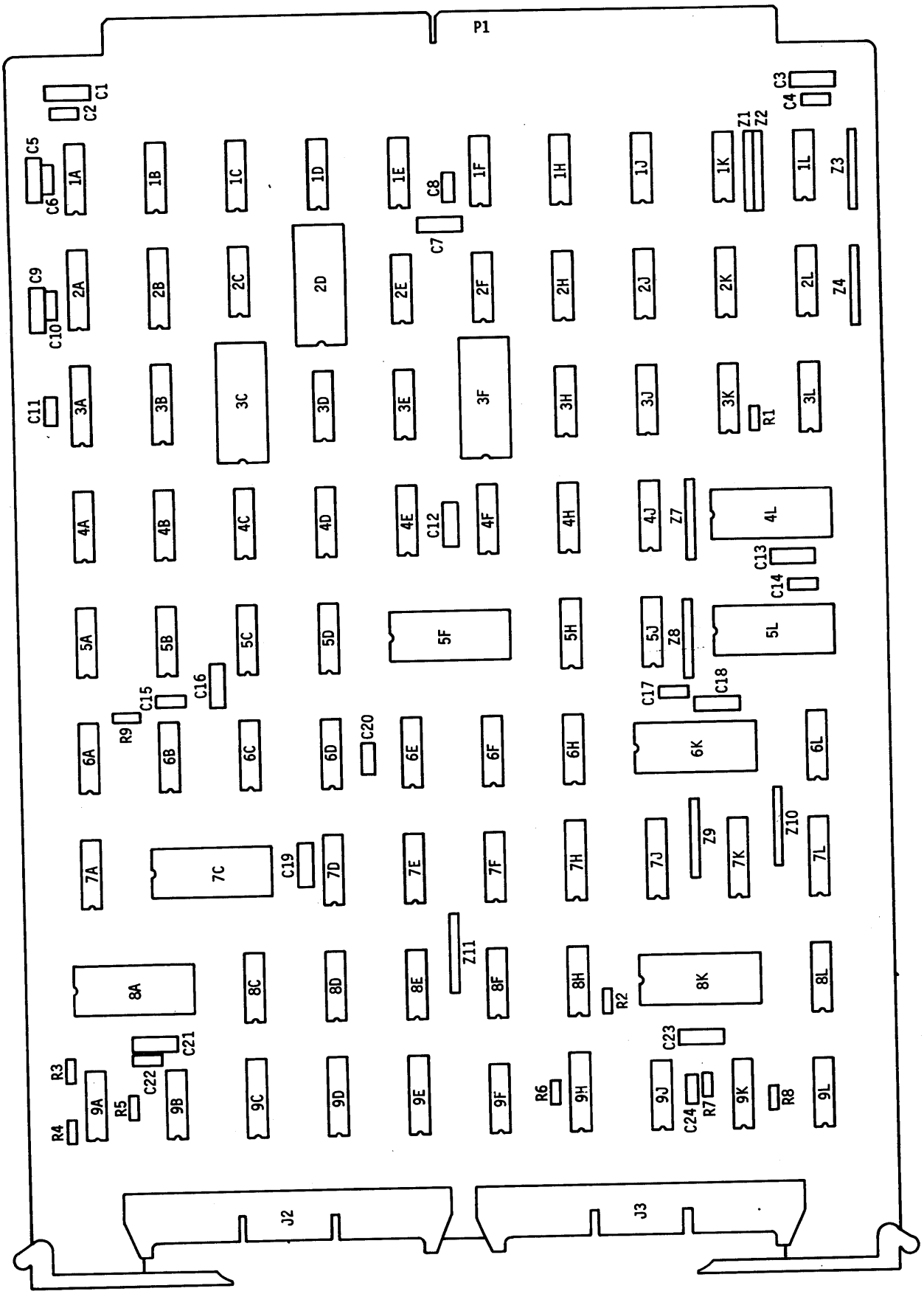


Figure 5-4. Data Card

5.6 READ ONLY MEMORY CARD PARTS LIST (FIGURE 5-5)

Reference Designator	Part Number	Nomenclature	Qty
C1 through C10,C17	TG-S10	Capacitor, 0.01 µf, 100V	11
C18,C19,C23,C24	150D475X0010A2	Capacitor, 4.7 µf, 10V	4
R4	RC07GF471J	Resistor, 470 ohm, 1/4W, ±5%	1
Z1	0980002-01	Integrated circuit	1
Z2	0980002-02	Integrated circuit	1
Z3	0980002-03	Integrated circuit	1
Z4	0980002-04	Integrated circuit	1
Z5	0980002-05	Integrated circuit	1
Z6	0980002-06	Integrated circuit	1
Z7	0980002-07	Integrated circuit	1
Z8	0980002-08	Integrated circuit	1
Z9	0980002-09	Integrated circuit	1
Z10	0980002-10	Integrated circuit	1
Z11	0980002-11	Integrated circuit	1
Z12	0980002-12	Integrated circuit	1
Z33,Z34,Z35,Z36,Z37	U6B900959X	Integrated circuit	5
Z38	U6B930159X	Integrated circuit	1

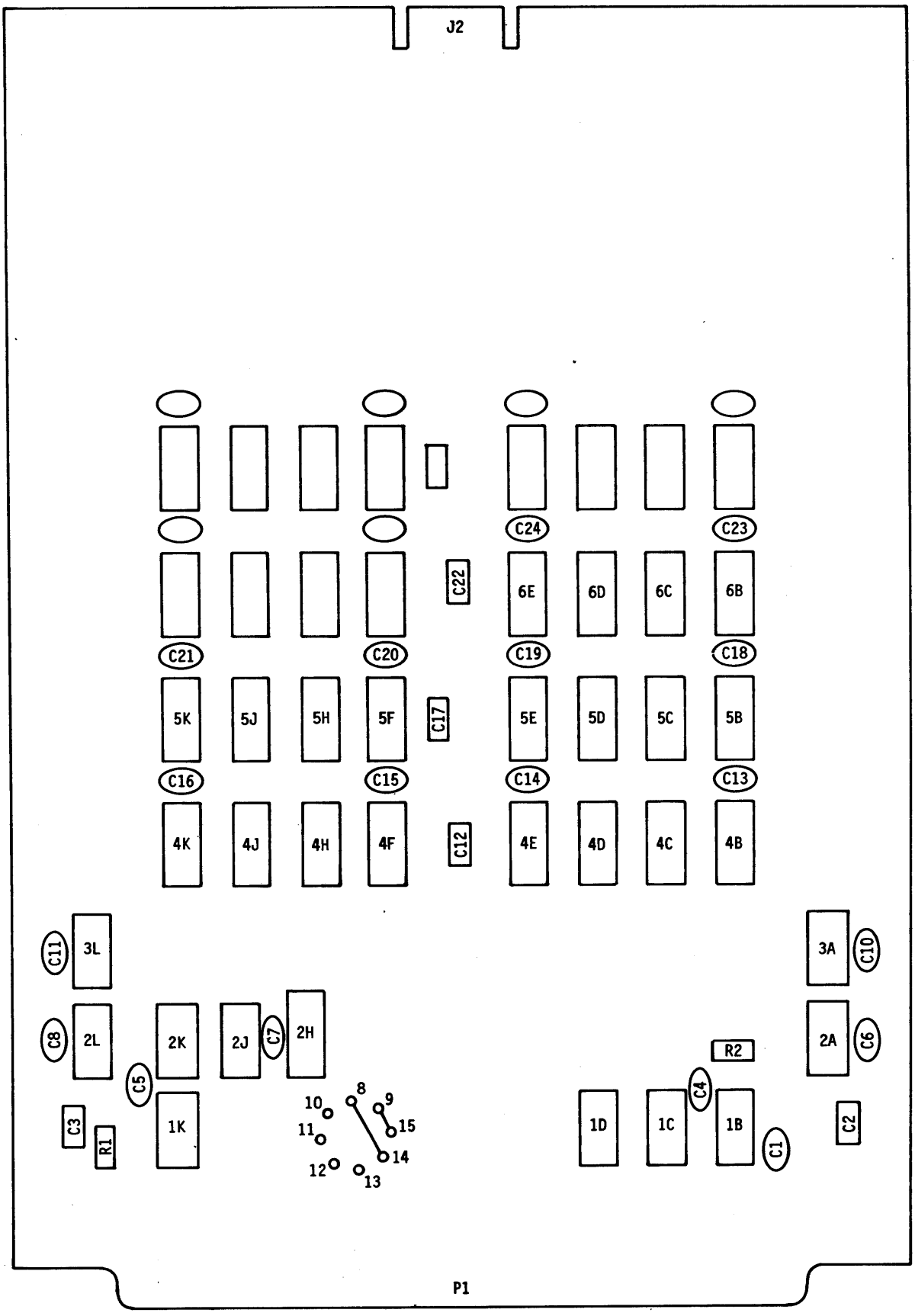


Figure 5-5. Read Only Memory Card

5.7 CONTROL CARD PARTS LIST (FIGURE 5-6)

Reference Designator	Part Number	Nomenclature	Qty
C1,C3,C5,C7,C12,C17,C20, C23,C24,C27,C29,C32, C36,C38,C40,C45,C46	150D226X0015B2	Capacitor, 22 µf, 15V	17
C2,C4,C8,C10,C13,C15, C16,C18,C19,C21,C25, C26,C28,C30,C31,C33, C34,C35,C37,C39,C41, C42,C43,C44,C49,C50, C51	DL46BU104Z	Capacitor, 0.1 µf, 100V	27
C6	DL46BX203J	Capacitor, 0.02 µf, 25V ±5%	1
C9	T362A155K025AS	Capacitor, 1.5 µf, 25V	1
C11	DM15-470J	Capacitor, 47 pf	1
C14	T362C686K015AS	Capacitor, 68 µf, 15V	1
C22	DM15-101J	Capacitor, 100 pf	1
C47	DM15-330J	Capacitor, 33 pf	1
C48	DM15-270J	Capacitor, 27 pf	1
CR1,CR2,CR3,CR4	FDH600	Diode	4
DL1	CS20000127	Delay line, 20 ns	1
J2,J3	3433-1002	Connector	2
L1	WEE-0.56	Inductor, 56 µh, ±10%	1
Q1	2N2369A	Transistor	1
Q2	2N3725	Transistor	1
R1,R22,R27	RC07GF101J	Resistor, 100 ohm, 1/4W, ±5%	3
R2 through R10,R17,R21, R28 through R41,R43 through R50,R52,R53, R54,R58	RC07GF471J	Resistor, 470 ohm, 1/4W, ±5%	37
R11	RC07GF201J	Resistor, 200 ohm, 1/4W, ±5%	1
R12,R13,R14	RC07GF301J	Resistor, 300 ohm, 1/4W, ±5%	3
R15	RC07GF473J	Resistor, 47K, 1/4W, ±5%	1
R16	RC07GF273J	Resistor, 27K, 1/4W, ±5%	1
R18,R20,R26,R42,R51	RC07GF221J	Resistor, 220 ohm, 1/4W, ±5%	5
R19	RC07GF561J	Resistor, 560 ohm, 1/4W, ±5%	1
R23	RC07GF682J	Resistor, 6.8K, 1/4W, ±5%	1
R24	RC07GF512J	Resistor, 5.1K, 1/4W, ±5%	1
R25	RC07GF331J	Resistor, 330 ohm, 1/4W, ±5%	1
R55	RC07GF103J	Resistor, 10K, 1/4W, ±5%	1
R56	RC07GF203J	Resistor, 20K, 1/4W, ±5%	1
R57	3009P-1-103	Resistor, variable, 10K	1

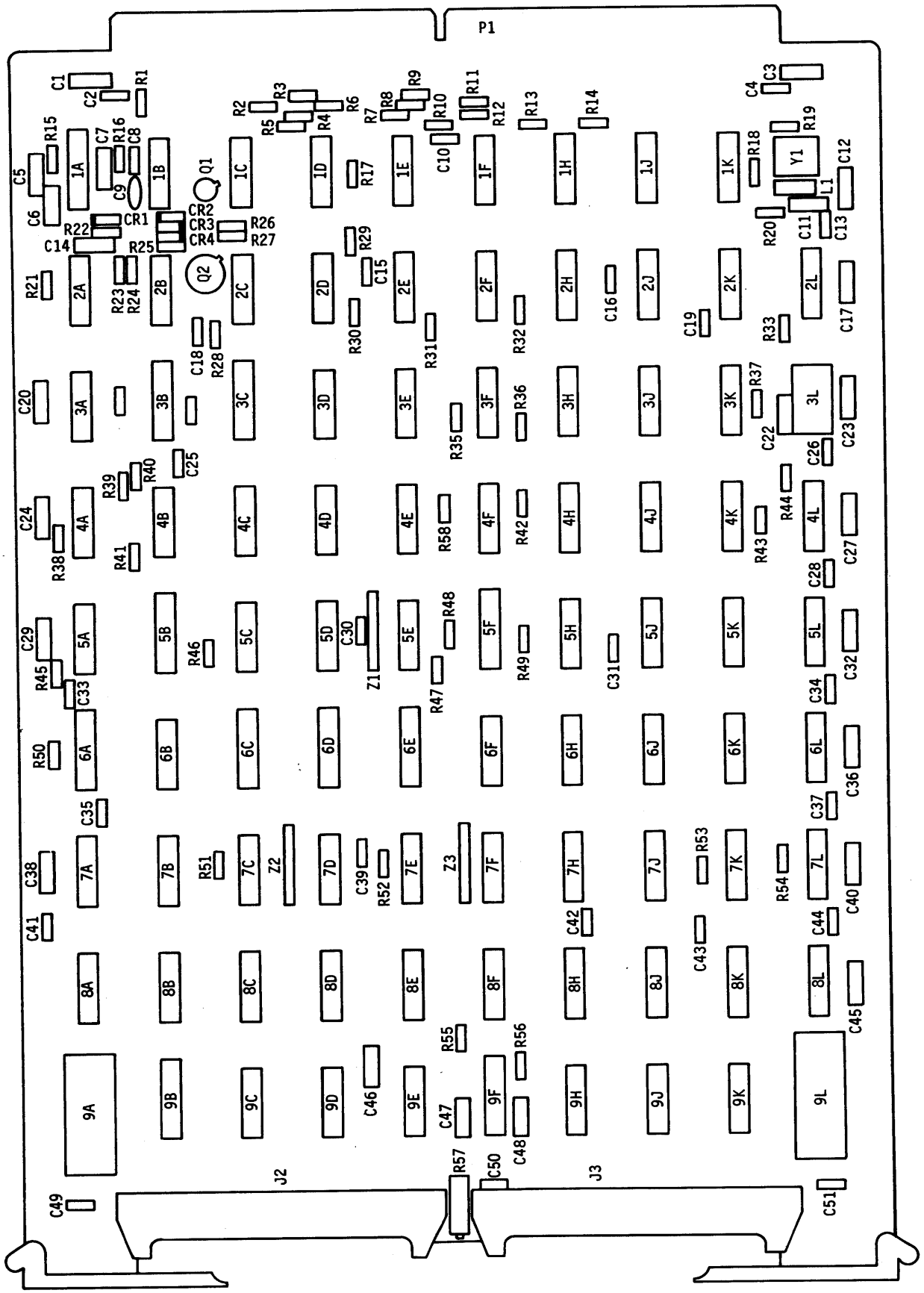


Figure 5-6. Control Card

5.7 CONTROL CARD PARTS LIST (FIGURE 5-6) (cont)

Reference Designator	Part Number	Nomenclature	Qty
Y1	MSC1167	Crystal, 20 MHz	1
Z1,Z2,Z3	CS20000129-1	Resistor module	3
1A,9F	U7B960259X	Integrated circuit	2
1B	SN7413N	Integrated circuit	1
1C,1K,5E,5H,6L,6H,8A, 8L,9K	SN74H04N	Integrated circuit	9
1D,1E,1F,2B,4F,7C	SN7438N	Integrated circuit	6
1H,5L	SN74H51N	Integrated circuit	2
1J,4C	SN7408N	Integrated circuit	2
2A,3F,4A	SN74H05N	Integrated circuit	3
2C,2D,2E,2F,3K,4H,4L, 4K,5C	SN74H74	Integrated circuit	9
2H,9H	SN7437N	Integrated circuit	2
2J,2K,6H,9E	SN74H11N	Integrated circuit	4
2L,5A	SN74H108N	Integrated circuit	2
3A	SN74164N	Integrated circuit	1
3B,3C,5B,5F,6A	U7B902259X	Integrated circuit	5
3D,3E,4D,5K,6F,6K,7F,8C	SN7400N	Integrated circuit	8
3H,4B,4E,6J,9J	SN74H20	Integrated circuit	5
3J,8F,8H	SN74H53	Integrated circuit	3
4J,5D,5J,6B,7B,8E	SN74H10N	Integrated circuit	6
6C,6D,6E	SN74161N	Integrated circuit	3
7A,8J	SN74H30N	Integrated circuit	2
7D,7E	SN74H102N	Integrated circuit	2
7J,7K,7L	SN74H22N	Integrated circuit	3
8B,8K	SN74S00N	Integrated circuit	2
8D	SN74H08N	Integrated circuit	1
9A,9L	SN74154N	Integrated circuit	2
9B	U7B932259X	Integrated circuit	1
9C,9D	SN74510N	Integrated circuit	2

5.8 8K MEMORY CARD PARTS LIST (FIGURE 5-7)

Reference Designator	Part Number	Nomenclature	Qty
A02	10001002	8K memory stock	1
C1,C7	DM15-681J	Capacitor, 680 pf	2
C2,C12,C13,C14,C31,C32, C33,C40,C41,C42,C43, C44,C45,C56,C57,C58	T362C226M025AS	Capacitor, 22 µf, 25V	16
C4,C5,C23 through C30	DM10-101J	Capacitor, 100 pf	10
C6,C11,C15,C17,C18, C21,C22,C34,C35,C38, C39,C46,C47,C48,C49, C50,C51,C54	TG-S10	Capacitor, 0.01 µf, 100V	18
C9	DM10-470J	Capacitor, 47 pf	1
C16,C19	DM15-331J	Capacitor, 330 pf	2
C20,C36,C37,C52,C53	150D475X0010A2	Capacitor, 4.7 µf, 10V	5
C55	DM10-221J	Capacitor, 220 pf	1
CR1 through CR163,CR165, CR166	FDH600	Diode	165
CR164	1N751A	Diode, zener 5.1V	1
CR167	1N5239B	Diode, zener	1
Q1,Q2,Q5,Q6,Q13,Q14, Q18,Q19,Q110,Q111	2N2905	Transistor	10
Q3,Q4,Q9,Q10,Q15,Q16, Q17,Q22,Q23,Q26,Q27, Q92	2N2369A	Transistor	12
Q7,Q8,Q20,Q21,Q28,Q31, Q32,Q35,Q39,Q40,Q43, Q44,Q47,Q48,Q51,Q52, Q55,Q56,Q59,Q60,Q63, Q64,Q67,Q68,Q71,Q72, Q75,Q76,Q79,Q80,Q83, Q84,Q87,Q88,Q91	2N2906	Transistor	36
Q11,Q12,Q24,Q25,Q29,Q30, Q33,Q34,Q37,Q38,Q41, Q42,Q45,Q46,Q49,Q50, Q53,Q54,Q57,Q58,Q61, Q62,Q65,Q66,Q70,Q73, Q74,Q77,Q78,Q81,Q82, Q85,Q86,Q89,Q90,Q93	2N4013	Transistor	37
Q94 through Q109	2N3725	Transistor	16
R1,R15,R47	RC07GF103J	Resistor, 10K, 1/4W, ±5%	3
R2,R3,R10,R16,R25,R33	RC07GF(type)J	Resistor, selected	6
R4,R5,R8,R9,R26,R27, R31,R32	M10	Resistor, W.W., 21.5 ohm, 1W, ±1%	
R6,R7,R22,R28,R30,R35, R46,R133,R134,R142, R144,R146	RC07GF102J	Resistor, 1K, 1/4W, ±5%	12

5.8 8K MEMORY CARD PARTS LIST (FIGURE 5-7) (cont)

Reference Designator	Part Number	Nomenclature	Qty
R11,R12,R34,R36	RC07GF221J	Resistor, 220 ohm, 1/4W, ±5%	4
R13,R14,R23,R24,R37,R38, R48,R49,R54,R55,R80, R81	RC07GF151J	Resistor, 150 ohm, 1/4W, ±5%	12
R17,R20,R41,R43,R56,R58, R59,R61,R62,R64,R65, R67,R68,R70,R71,R73, R74,R76,R77,R79,R82, R84,R85,R87,R88,R90, R91,R93,R94,R96,R97, R99,R100,R102,R103, R105	RC07GF152J	Resistor, 1.5K, 1/4W, ±5%	36
R18,R19,R21,R42,R44,R45, R52,R106,R108,R109, R145	RC07GF471J	Resistor, 470 ohm, ±5%	11
R29,R53	RN55D6810F	Resistor, 681 ohm, ±5%	2
R39	3009P-1-103	Potentiometer, 10K	1
R40	RC07GF472J	Resistor, 4.7K, 1/4W, ±5%	1
R50	RN55D2001F	Resistor, 2K, 1/8W, ±1%	1
R51	RC07GF330J	Resistor, 33 ohm, 1/4W, ±5%	1
R57,R60,R63,R66,R69,R72, R75,R78,R83,R86,R89, R92,R95,R98,R101, R104	RC07GF222J	Resistor, 2.2K, 1/4W, ±5%	16
R107,R110,R112,R113, R115,R116,R118,R119, R121,R122,R124,R125, R127,R128,R130,R131	RC07GF390J	Resistor, 39 ohm, 1/4W, ±5%	16
R111,R114,R117,R120, R123,R126,R129,R132	M30	Resistor, W.W., 17.4 ohm, 4.5W, ±1%	8
R135	M16	Resistor, W.W., 68 ohm, 1.75W, ±1%	1
R136	M9	Resistor, W.W., 511 ohm, 0.3W, ±1%	1
R137	M9	Resistor, W.W., 2 ohm, 0.3W, ±1%	1
R138,R139	RC07GF331J	Resistor, 330 ohm, 1/4W, ±5%	2
R140,R141	RC07GF681J	Resistor, 680 ohm, 1/4W, ±5%	2
R143	RC07GF301J	Resistor, 300 ohm, 1/4W, ±5%	1
RM1 through RM8	10001023	Resistor module	4
T1 through T4	PA2332	Transformer	4
Z1,Z2	U6B960259X	Integrated circuit	2
Z3	U6A901659X	Integrated circuit	1
Z4	SN74H11N	Integrated circuit	1
Z5	U6B932259X	Integrated circuit	1
Z6	N8242A	Integrated circuit	1
Z7	U6A900259X	Integrated circuit	1
Z8 through Z11	U6B930159X	Integrated circuit	4
Z12 through Z19	QC7520J	Integrated circuit	4
Z20 through Z25	SN7438N	Integrated circuit	4

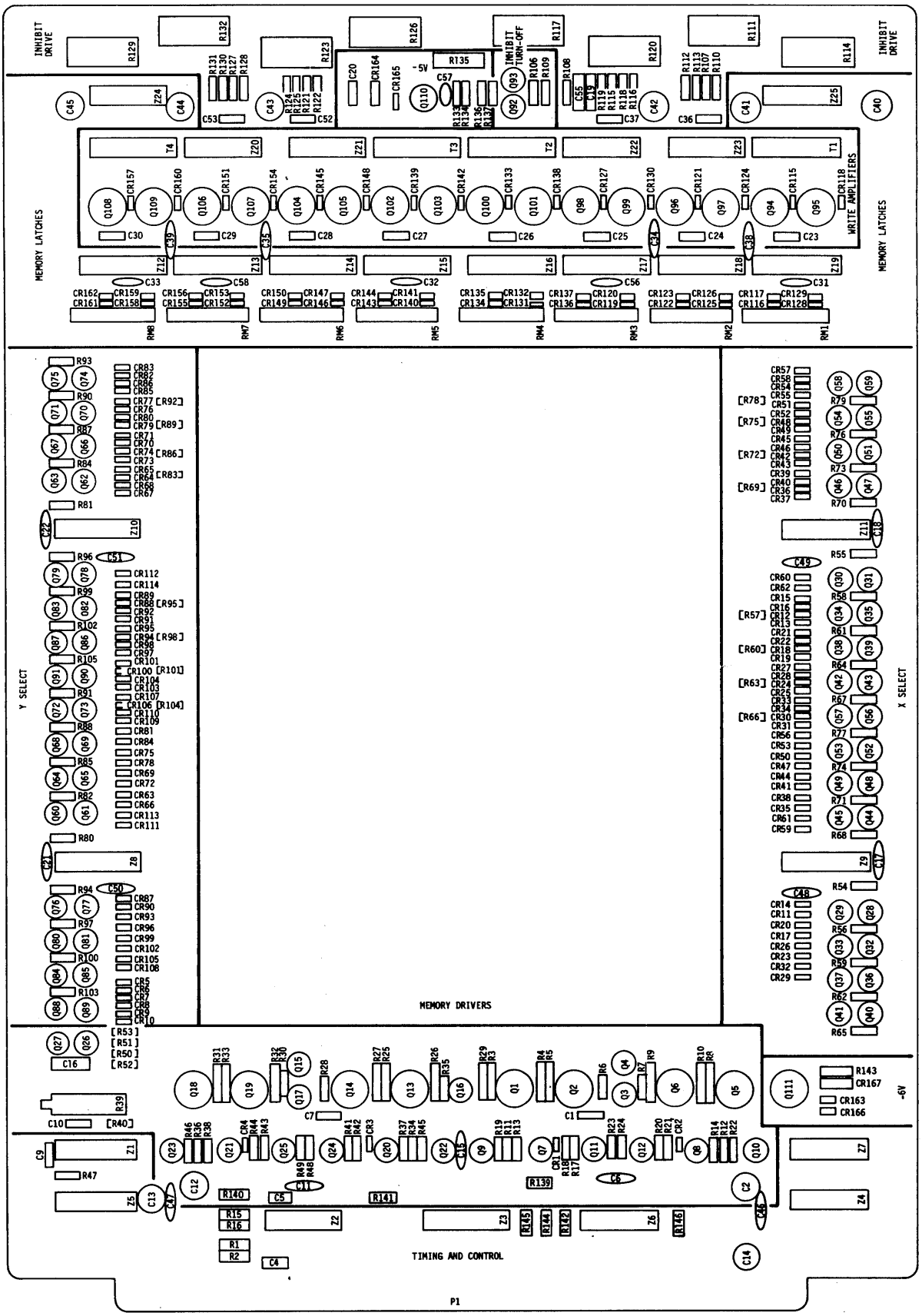


Figure 5-7. 8K Memory Card

5.9 POWER SUPPLY PARTS LIST (FIGURES 5-8 AND 5-9)

Reference Designator	Part Number	Nomenclature	Qty
A1	A200001221	Regulator board	1
A2	A200001220	Failure detection board	1
B1	4500A	Fan	1
C1	CGS133U025R4C3PL	Capacitor, 13K µf, 25V	1
C2	CGS134U015X4C3PL	Capacitor, 130K µf, 15V	1
C3	CGS253U040W4C3PL	Capacitor, 25K µf, 40V	1
*C4	VK105	Capacitor, 1 µf, 3V	1
*C5,C6,C7	225P1059R75	Capacitor, 1 µf, 75V	3
*C9	225P3349R75	Capacitor, 0.33 µf, 75V	1
*CR1,CR2	1N1183	Diode W/mtg kit	2
*CR3,CR4,CR5,CR6	1N1200A	Diode W/mtg kit	4
F1	313005	Fuse, 5A, 3AG, SB	1
J1	7486	Connector, power	1
J2	SAC22S/1-2	Connector, 22 pin	1
*J3	5411-31	Connector, 15 pin	1
K1	A2000119	Relay circuit assembly	1
L1	7382	Choke	
P1	7484	Connector, power	1
*Q1 through Q8	2N3055	Transistor	8
*R1	PW-5	Resistor, 0.45 ohm, 5W	1
*R2,R3,R4,R8	PW-5	Resistor, 0.22 ohm, 5W	4
*R5,R9	PW-5	Resistor, 0.33 ohm, 5W	2
R10	PW-5	Resistor, 100 ohm, 5W	1
R11	PW-5	Resistor, 400 ohm, 5W	1
*R12	PW-10	Resistor, 0.05 ohm, 10W	1
S1	756-K5	Switch	1
*S2	20700 series	Switch, thermal, open 300°F, close 260°F	1
S3	201	Switch	1
*SCR1	C20U	SCR W/mtg kit	1
T1	7383A	Transformer	1
TB1	699-3/4ST-13	Terminal block	1

*These line entries (items) are mounted on heatsink assembly and are not individually illustrated in figure 5-10.

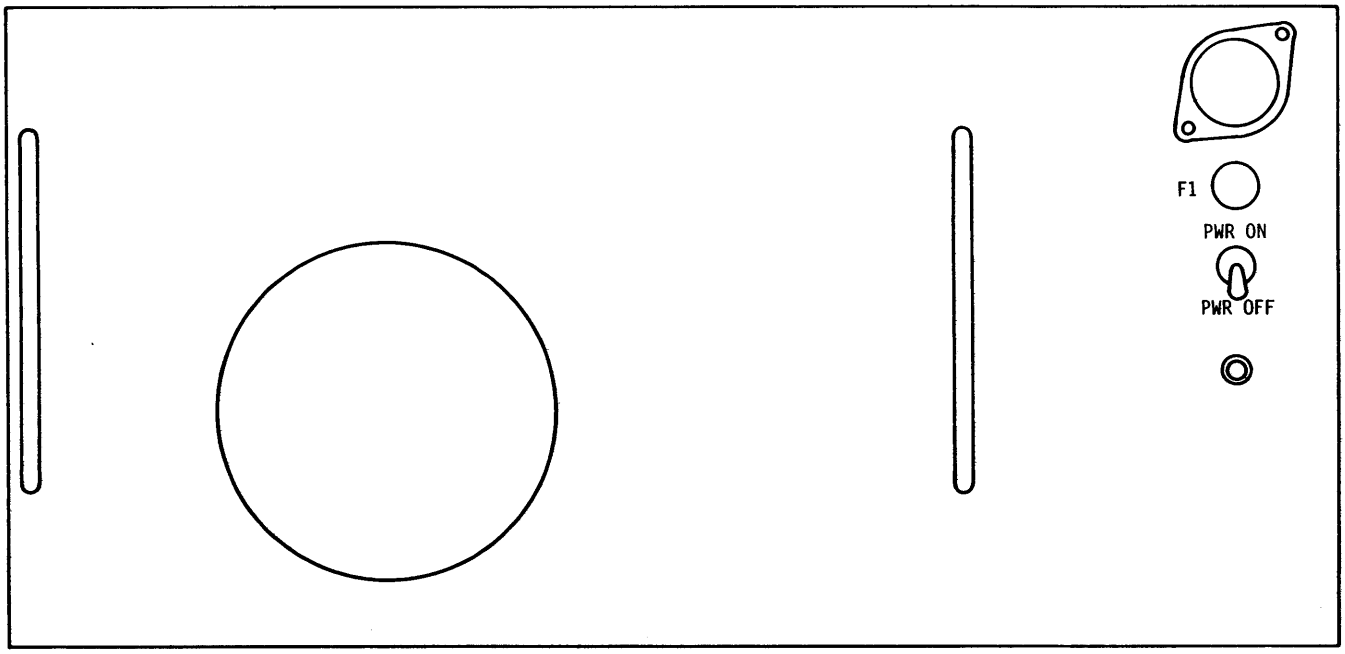


Figure 5-8. Power Supply (Front)

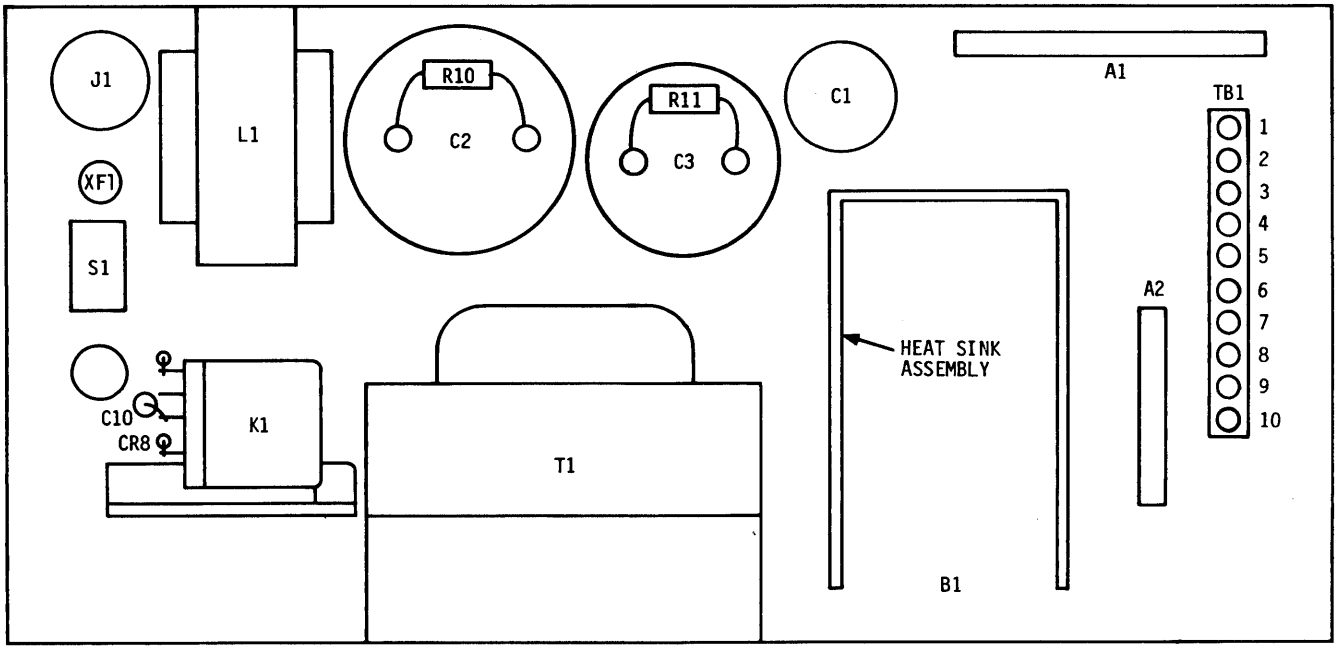


Figure 5-9. Power Supply (Back)

5.9 POWER SUPPLY (LATER MODEL) PARTS LIST (FIGURES 5-8 AND 5-9)

Reference Designator	Part Number	Nomenclature	Qty
A1	10001152	Regulator board	1
A2	10001154	Failure detection board	1
B1	4500A	Fan	1
C1	CG5822V025R4C3PL	Capacitor, 8.2K uf, 25V	1
C2	CGS134U015X4C3PL	Capacitor, 130K uf, 15V	1
C3	CGS253U040W4C3PL	Capacitor, 25K uf, 40V	1
*C4	VK105	Capacitor, 1 uf, 3V	1
*C5, C6, C7	225P1059R75	Capacitor, 1 uf, 75V	3
*C9	225P3349R75	Capacitor, 0.33 uf, 75V	1
C10	220P-6PS-10	Capacitor, 0.1 uf, 600V	1
C11	6-85ASD100-15	Capacitor, 100 uf, 15V	1
*CR1, CR2	1N1183	Diode W/mtg kit	2
*CR3, CR4, CR5, CR6	1N1200A	Diode W/mtg kit	4
CR7, CR9, CR10, CR11	1N40002	Diode	4
CR8	1N40005	Diode	1
F1	313005	Fuse, 5A 3AG, SB	1
J1	7486	Connector, power	1
J2	SAC22S/1-2	Connector, 22 pin	1
*J3	5411-31	Connector, 15 pin	1
K1	1R-1225-3C-12D	Relay	1
L1	7382	Choke	
P1	7484	Connector, power	1
*Q1 through Q8	2N3055	Transistor	8
*R1	PW-5	Resistor, 0.45 ohm, 5W	1
*R2, R3, R4, R8	PW-5	Resistor, 0.22 ohm, 5W	4
*R5, R9	PW-5	Resistor, 0.33 ohm, 5W	2
R10	PW-5	Resistor, 100 ohm, 5W	1
R11	PW-5	Resistor, 400 ohm, 5W	1
*R12	PW-10	Resistor, 0.05 ohm, 10W	1
R20	RC20GF103J	Resistor, 10K, ±5%, ½W	1
S1	7561-K5	Switch	1
*S2	H1-D	Switch, thermal, open 300°F, close 260°F	1
S3	201	Switch	1
*SCR1	C20U	SCR W/mtg kit	1
T1	7383A	Transformer	1
T2	7444	Transformer	1

Reference Designator	Part Number	Nomenclature	Qty
TB1	699-3/4ST-13	Terminal block	1
XC1	VR3	Capacitor clamp	1
XC2	VR12	Capacitor clamp	1
XC3	VR10	Capacitor clamp	1
XF1	342014	Fuseholder	1
	10001169	Panel	1
	10001207	Heat Sink Assembly	1
	774	Clamp, cable 5/16 diameter nylon	1
	17409-B	Cable, power	1
	2051 1013-12	Handle	2
	2372	Standoff #4-40 THD x½lg	2
	500 series 3.5" LG	Card guide	2
	007900	Connector key	1
	1461	Terminal lug, insulated	3
	R41495	Terminal lug	4
	10001171	Input voltage labed	1
	D20000389	Decal-tap and jumper table	1
	D20000321	Cover-protection	1

*These line entries (items) are mounted on heatsink assembly and are not individually illustrated in figure 5-10.

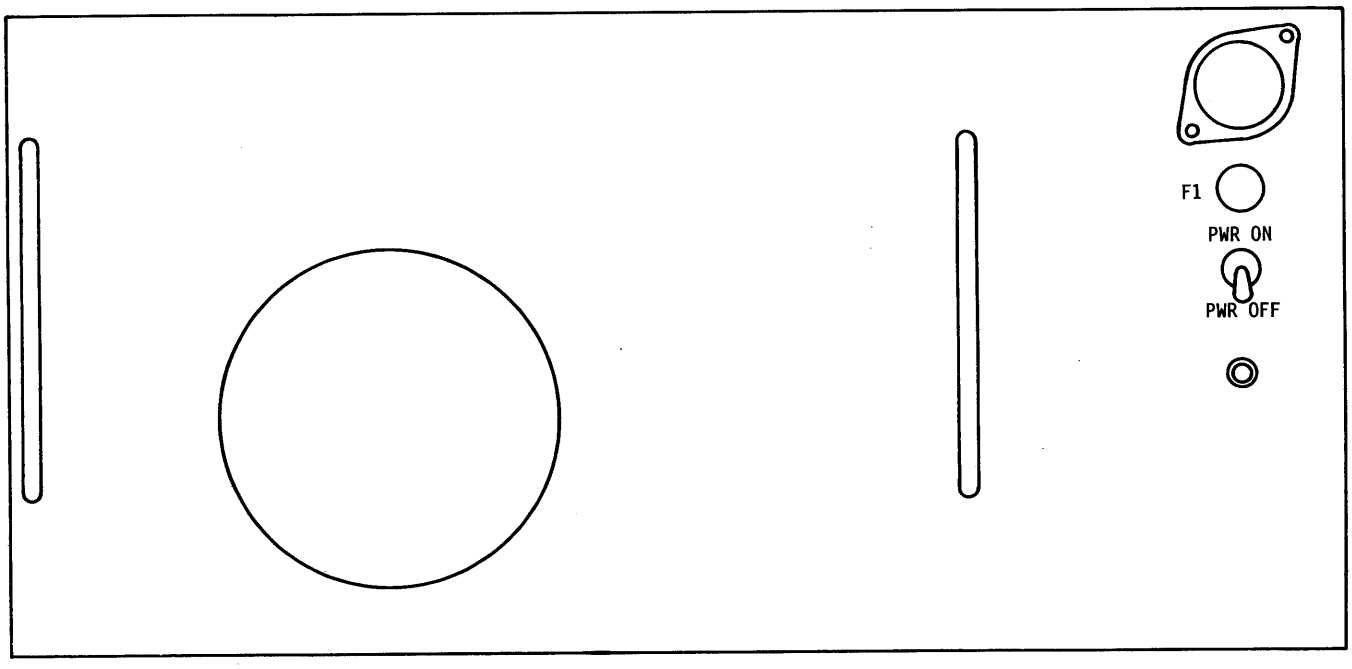


Figure 5-8.1. Power Supply (Later Model) (Front)

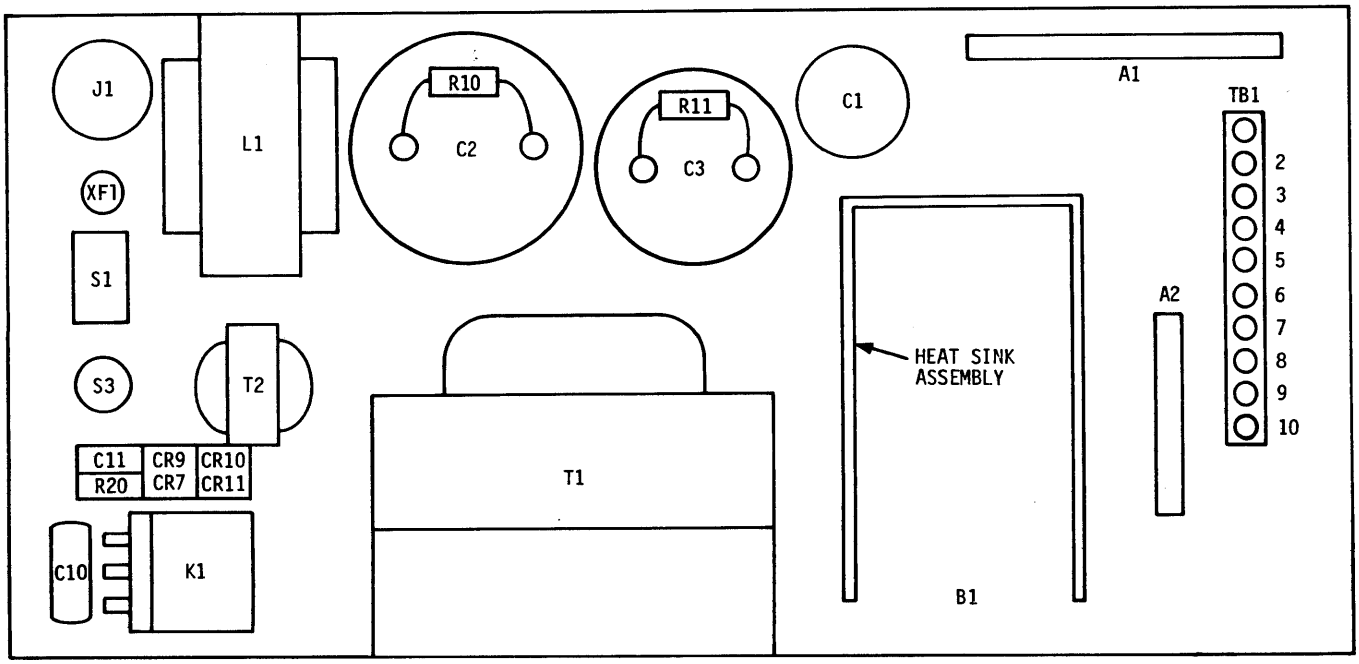


Figure 5-9.1. Power Supply (Later Model) (Back)

5.10 REGULATOR BOARD PARTS LIST (FIGURE 5-10)

Reference Designator	Part Number	Nomenclature	Qty
C1, C3, C11, C21	5R6UF-35V	Capacitor, 5.6 uf, 35V	5
C2	5GA-T56	Capacitor, 560 pf, 1000V	1
C12	5GA-T30	Capacitor, 300 pf, 1000V	1
C14	225P10491WD3	Capacitor, 0.1 uf, 100V	1
C22	225P10391WD3	Capacitor, 0.01 uf, 100V	1
C23	5GA-D18	Capacitor, 0.0018 uf, 1000V	1
C24	T310B226K015	Capacitor, 22 uf, 15V	1
C25	UK474	Capacitor, 0.47 uf, 3V	1
C26	MTA402T006J2DP	Capacitor, 4000 uf, 6V	1
C27	DM15F151M	Capacitor, 150 pf	1
C28	TGS-10	Capacitor, 0.01 uf, 100V	1
CR21	1N751	Diode, zener	1
IC1, IC11, IC21	LM305A	Integrated circuit	3
R1, R28	RC20GF101K	Resistor, 100 ohm, ½W ±10%	2
R2	RC20GF470K	Resistor, 47 ohm, ½W, ±10%	1
R3, R3	RC20GF471K	Resistor, 470 ohm, ½W, ±10%	2
R4	RC20GF183K	Resistor, 18K, ½W, ±10%	1
R5, R15	89PR2K	Potentiometer, 2K	2
R11, R21	RC20GF560K	Resistor, 56 ohm, ½W, ±10%	2
R12	RC20GF270K	Resistor, 27 ohm, ½W, ±10%	1
R13	RC20GF271K	Resistor, 270 ohm, ½W, ±10%	1
R14	RC20GF123K	Resistor, 12K, ½W, ±10%	1
R16	RC20GF122K	Resistor, 1.2K, ½W, ±10%	1
R22	RC20GF820K	Resistor, 82 ohm, ½W, ±10%	1
R23	RC20GF331K	Resistor, 330 ohm, ½W, ±10%	1
R24	RC20GF472K	Resistor, 4.71K, ½W, ±10%	1
R25	89PR1K	Potentiometer, 1K	1
R26	RC20GF222K	Resistor, 2.2K, ½W, ±10%	1
R27	89PR100	Potentiometer, 100 ohm	1
R29	RC20GF120K	Resistor, 12 ohm, ½W, ±10%	1
Q1, Q21	2N374	Transistor	2
Q11	2N2905A	Transistor	1
	10001178	P. C. Board	1
	2230C-5	Heatsink, to 5	1
		Screw 4.40 x 3/8 pan hd.	4
		Washer, lock #4 int. star	4
		Nut, hex 4.40	4

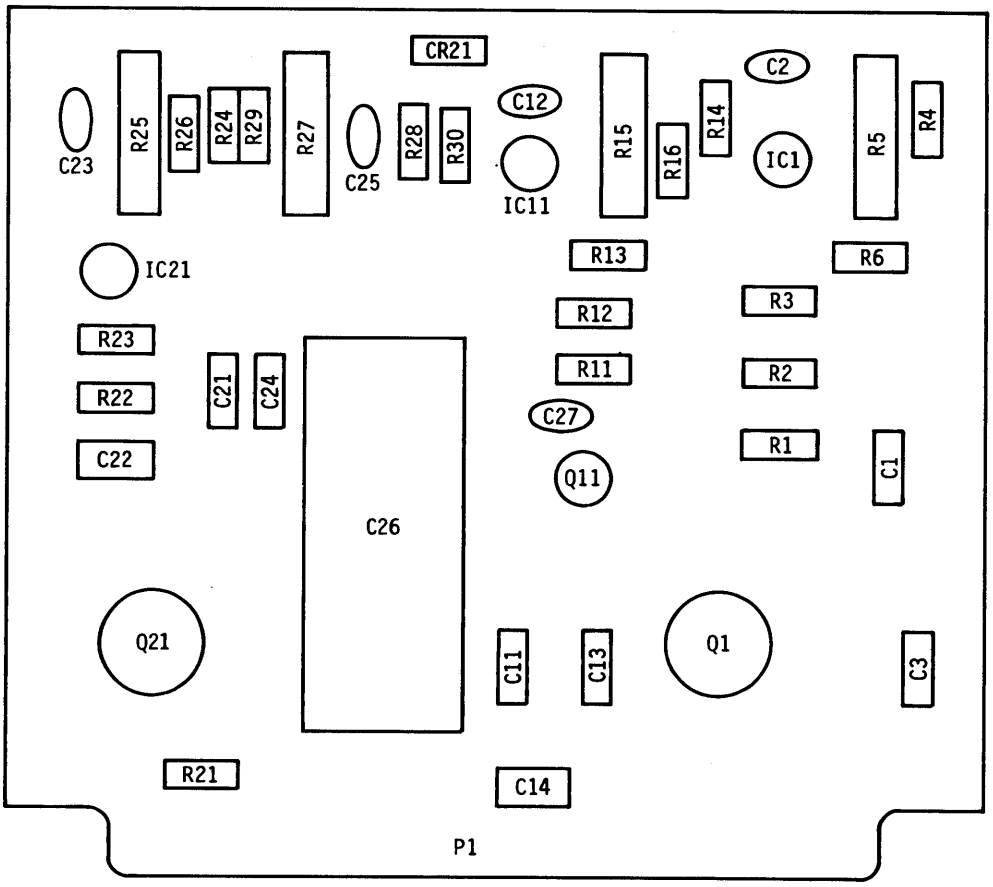


Figure 5-10. Regulator Board

5.11 FAILURE DETECTION BOARD PARTS LIST (FIGURE 5-11)

Reference Designator	Part Number	Nomenclature	Qty
C1	225P2249R75203	Capacitor, 0.22 µf, 75V	1
C2	TA396K020P1F	Capacitor, 39 µf, ±20%, 20V	1
C3	9-85ARD1200-25-1	Capacitor, 1200 µf, 25V	1
C6	8-85ASD290-12-1	Capacitor, 290 µf, +100% -10%, 12V	1
CR1,CR2,CR3,CR4,CR5,CR10	1N4005	Diode	6
CR6	1N751A/1N4733A	Diode, zener	1
CR7		Diode (selected)	1
CR8,CR9	1N4720	Diode	2
CR11	1N4536	Diode	1
IC1	LM311	Integrated circuit	1
Q1,Q2,Q4,Q5	2N2222	Transistor	4
Q6	2N3053	Transistor	1
R1,R3	RC20GF153K	Resistor, 15K, 1/2W, ±10%	2
R2,R4,R11,R14	RC20GF472K	Resistor, 4.7K, 1/2W, ±10%	4
R5,R6,R18,R20	RC20GF101K	Resistor, 100 ohm, 1/2W, ±10%	4
R7	RC20GF392K	Resistor, 3.9K, 1/2W, ±10%	1
R8	89PR100K	Potentiometer, 100K	1
R9	RC20GF123K	Resistor, 12K, 1/2W, ±10%	1
R12	RC20GF103K	Resistor, 10K, 1/2W, ±10%	1
R15	RC20GF152K	Resistor, 1.5K, 1/2W, ±10%	1
R17	RC20GF104K	Resistor, 100K, 1/2W, ±10%	1
R19	RC20GF471K	Resistor, 470 ohm, 1/2W, ±10%	1
R21	PW-5	Resistor, 15 ohm, 5W, ±10%	1
R22	RC20GF224K	Resistor, 220K, 1/2W, ±10%	1

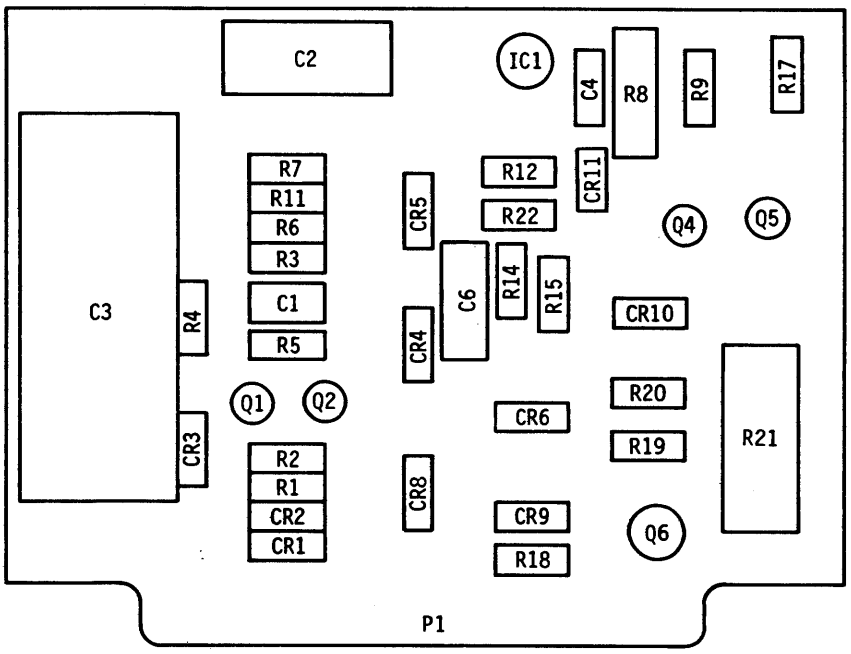


Figure 5-11. Failure Detection Board

Supplement 2

READ ONLY MEMORY

TABLE OF CONTENTS

List of Illustrations

List of Tables

Supplement 2

READ ONLY MEMORY

S2.1 INTRODUCTION

The Read Only Memory (ROM) provides storage and high-speed readout of program commands and constants. The printed circuit board can accommodate a memory capacity of 256 sixteen-bit words up to 2,048 sixteen-bit words in selectable increments of 256 words.

The memory elements are bipolar integrated circuits contained in 16-pin dual in-line packages and have a maximum access time of 50 nanoseconds. The IC packages are organized in groups of four, each group capacity being 256 sixteen-bit words. The addressing scheme permits access to any one, several contiguous, or all of the 16 possible 256-word groups. The addressing scheme involves dedicated IC package installation and jumper wire programming.

A simplified logic block diagram of the ROM is shown in figure S2.1-1. A 12-bit address word is used to access the memory locations. A control bit (CPEN/) enables the addressing and readout operations. The least-significant eight bits (L00X through L07X) select one of 256 locations. L08X and L09X are decoded into four lines to select one of the four basic 256-word memory groups (pages) if only half of board capacity (1,024 words) is to be used. If both halves (2,048 words) are to be used, L10A is decoded with L09X and L08X into one of eight lines to select one of the eight 256-word memory groups. L11A is used for addressing above 2,048 words.

The jumpers at the output of the decoder can be used to select starting addresses for the eight 256-word pages on the board. Alternately, the IC memory packages can be located in dedicated positions on the board to achieve starting address selection.

S2.2 THEORY OF OPERATION

Logic drawings LD 0380 through LD 0389, contained in Logic Diagrams Manual LD 1010 illustrate the internal circuits within the ROM. The logic circuits on these

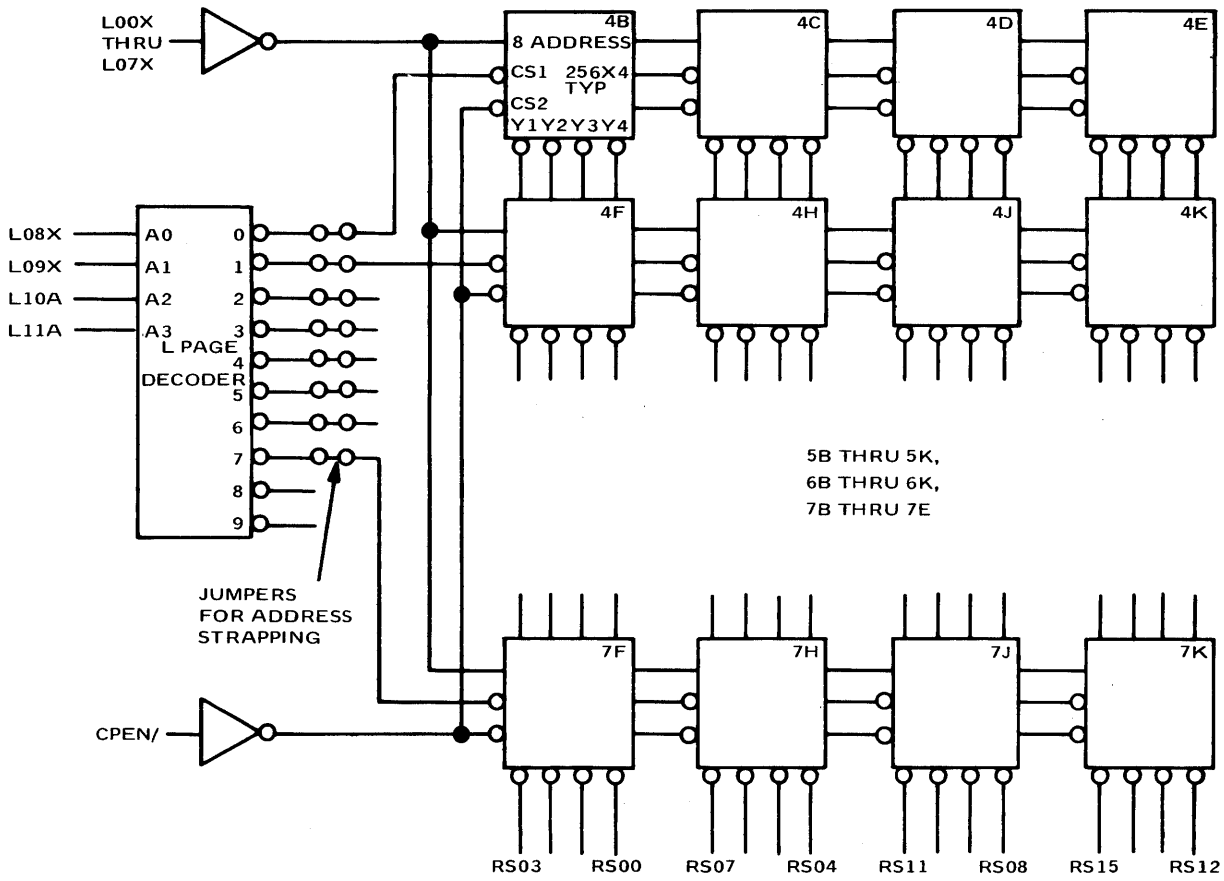


Figure S2.1-1. ROM Block Diagram

drawings are divided into three basic sections; address buffering and decoding, and memory extension (if used), and memory.

S2.2.1 MEMORY ORGANIZATION

Figure S2.2-1 is a ROM locator diagram. As shown, the memory is organized in eight 256-word ROM pages. Each ROM page can be addressed independently through the decoding logic, so that starting addresses of 000_{10}, 511_{10}, 767_{10}, 1023_{10}, 1279_{10}, 1535_{10}, or 1790_{10}, can be implemented.

S2.2.2 ADDRESS BUFFERING AND DECODING

Address bits L00X through L11X are received from the CPU, together with control panel enable term CPEN/. Address bit complements L00X/ through L07X/ perform a selection on one of 256 words from any of the eight ROM pages.

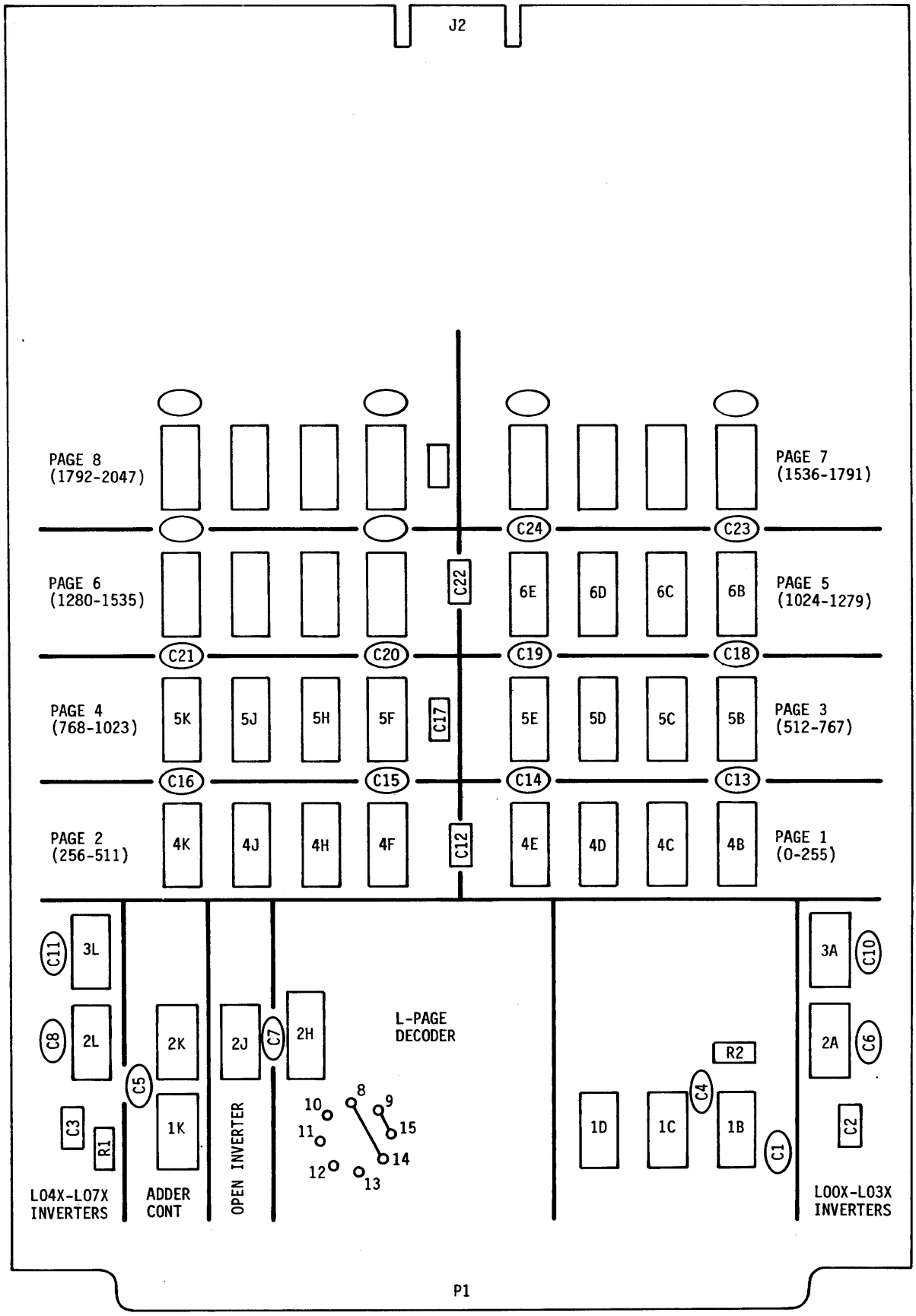


Figure S2.2-1. ROM Locator Diagram

Selection of a specific 256-word ROM page is implemented with a decimal decode of bits L08X, L09X, and either L10A or L11A. The decode of these three bits drives one of the eight output lines of decoder 2H low to enable one of the eight 256-word groups. Which group depends on the output jumper wire configuration.

With a specific word address defined by bits L00X/ through L07X, and a group selector line from decoder 2H driven low, one word out of the eight 256-word ROM pages is selected for readout depending on the decoder output jumper wire configuration. With the jumpers installed as shown in 0720, for example, a binary progression through the modulus of L08X•L09X•L10X will energize decimal output lines 0 through 7 in succession. Consequently, a starting address of 000 activates the starting word in ROM page 1, a starting address of 255_{10} activates the starting word in ROM page 2, and so on. By rearranging the decoder output jumpers, different starting addresses can be selected for all possible memory sizes in increments of 256. The jumpering required for all possible starting addresses and memory sizes is listed in table S2.2-1.

When CPEN/ is true (control panel not enabled), the ROM outputs are enabled. In the control panel mode, CPEN/ is false to disable the ROM outputs and allow the control panel switches to supply the data for the R_1, R_2 registers.

S2.3 PARTS LIST

The physical location of the electronic parts on the ROM board is shown in figure S2.2-1, and the parts list is on page S2-6.

TABLE S2.2-1. STARTING ADDRESS SELECTION

MEMORY SIZE	START ADDRESS	JUMPER ADDRESSING REMOVE JUMPERS	JUMPER ADDRESSING INSTALL JUMPERS	ROM LOCATION ADDRESSING
256	000	--	--	4B-4E
256	256	0	A1-B0	4F-4K
256	512	0	A2-B0	5B-5E
256	768	0	A3-B0	5F-5K
256	1024	0	A4-B0	6B-6E
256	1280	0	A5-B0	6F-6K
256	1536	0	A6-B0	7B-7E
256	1792	0	A7-B0	7F-7K
512	000	--	--	4B-4K
512	256	0,1	A1-B0,A2-B1	4F-5E
512	512	0,1	A2-B0,A3-B1	5B-5K
512	768	0,1	A3-B0,A4-B1	5F-6E
512	1024	0,1	A4-B0,A5-B1	6B-6K
512	1280	0,1	A5-B0,A6-B1	6F-7E
512	1536	0,1	A6-B0,A7-B1	7B-7K
768	000	--	--	4B-5E
768	256	0,1,2	A1-B0,A2-B1,A3-B2	4F-5K
768	512	0,1,2	A2-B0,A3-B1,A4-B2	5B-6E
768	768	0,1,2	A3-B0,A4-B1,A5-B2	5F-6K
768	1024	0,1,2	A4-B0,A5-B1,A6-B2	6B-7E
768	1280	0,1,2	A5-B0,A6-B1,A7-B2	6F-7K
1024	000	--	--	4B-5K
1024	256	0,1,2,3	A1-B0,A2-B1,A3-B2,A4-B3	4F-6E
1024	512	0,1,2,3	A2-B0,A3-B1,A4-B2,A5-B3	5B-6K
1024	768	0,1,2,3	A3-B0,A4-B1,A5-B2,A6-B3	5F-7E
1024	1024	0,1,2,3	A4-B0,A5-B1,A6-B2,A7-B3	6B-7K
1280	000	--	--	4B-6E
1280	256	0,1,2,3,4	A1-B0,A2-B1,A3-B2,A4-B3,A5-B4	4F-6K
1280	512	0,1,2,3,4	A2-B0,A3-B1,A4-B2,A5-B3,A6-B4	5B-7E
1280	768	0,1,2,3,4	A3-B0,A4-B1,A5-B2,A6-B3,A7-B4	5F-7K
1536	000	--	--	4B-6K
1536	256	0,1,2,3,4,5	A1-B0,A2-B1,A3-B2,A4-B3,A5-B4,A6-B5	4F-7E
1536	512	0,1,2,3,4,5	A2-B0,A3-B1,A4-B2,A5-B3,A6-B4,A7-B5	5B-7K
1792	000	--	--	4B-7E
1792	256	0,1,2,3,4,5,6	A1-B0,A2-B1,A3-B2,A4-B3,A5-B4,A6-B5,A7-B6	4F-7K
2048	000	--	--	4B-7K

Parts List for Read Only Memory Board (figure S2.2-1)

Ref. Designation	Part Number	Description	Qty	Vendor
C1,C4 thru C8,C10,C11, C13 thru C16, C18 thru C21,C23,C24	TG-S10	Capacitor, 0.01uf, 100v	18	Sprague
C2,C3,C12,C17,C22	150D475X0010A2	Capacitor, 4.7uf, 10v	5	Sprague
R1	RC07GF102J	Resistor, 1K, ¼w, 5%	1	
R2	RC07GF471J	Resistor, 470, ¼w, 5%	1	
1B	SN74H10N	Integrated Circuit	1	TI
1C	SN74H30N	Integrated Circuit	1	TI
1D,2K	SN74H04N	Integrated Circuit	2	TI
1K	SN74H74N	Integrated Circuit	1	TI
2A,2J,2L,3A,3L	SN74H40	Integrated Circuit	5	TI
2H	U7B930159X	Integrated Circuit	1	Fairchild
4B	0980010-01	Semiconductor ROM	1	Microdata
4C	0980010-02	Semiconductor ROM	1	Microdata
4D	0980010-03	Semiconductor ROM	1	Microdata
4E	0980010-04	Semiconductor ROM	1	Microdata
4F	0980010-05	Semiconductor ROM	1	Microdata
4H	0980010-06	Semiconductor ROM	1	Microdata
4J	0980010-07	Semiconductor ROM	1	Microdata
4K	0980010-08	Semiconductor ROM	1	Microdata
5B	0980010-09	Semiconductor ROM	1	Microdata
5C	0980010-10	Semiconductor ROM	1	Microdata
5D	0980010-11	Semiconductor ROM	1	Microdata
5E	0980010-12	Semiconductor ROM	1	Microdata
5F	0980010-13	Semiconductor ROM	1	Microdata
5H	0980010-14	Semiconductor ROM	1	Microdata
5J	0980010-15	Semiconductor ROM	1	Microdata
5K	0980010-16	Semiconductor ROM	1	Microdata
6B	0980010-17	Semiconductor ROM	1	Microdata
6C	0980010-18	Semiconductor ROM	1	Microdata
6D	0980010-19	Semiconductor ROM	1	Microdata
6E	0980010-20	Semiconductor ROM	1	Microdata
--	E-26-126	Wire, 26 AWG (Grn)	A/R	Southco

Supplement 3

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Supplement 3

8K MEMORY MODULE

S3.1 INTRODUCTION

This supplement provides coverage for the 8K Memory Module, hereinafter referred to as the 8K memory. Logic diagrams for the module are contained on sheets 0460 through 0468 in Logic Diagrams Manual, LD 1010.

S3.1.1 PURPOSE AND FUNCTION

The 8K memory serves as the main storage device for data and programs in the Central Processing Unit (CPU), and provides storage for a device connected to an optional direct memory access (DMA) channel. Storage is accomplished by a core memory module that contains a storage capacity of 8192 bytes. Each byte contains eight bits. Memory locations are accessed at the byte level by means of a 16-bit address received from either the CPU or the DMA. The three most significant bits of the 16-bit address contain a memory bank select code that selects any one of up to eight 8K memory modules installed in the CPU. The 13 least significant bits of the address select a specific memory location within the selected 8K memory module.

Control and timing signals from the CPU specify the type of operation to be performed on the addressed memory location. The types of operations that can be performed are as follows:

a. Half-Cycle Read. All cores in the addressed location are switched to the one state. If the core was in the zero state, a flux reversal is produced. If the core was already in the one state, no flux reversal is produced. The presence or absence of a flux reversal from each core (bit position) is sensed, and the resulting binary signals are stored in memory data latches. The output from these latches are supplied over the bidirectional memory data bus to the CPU and the DMA.

b. Full-Cycle Read. The first portion of the full-cycle read operation is identical to the half-cycle read operation. During the second half of the cycle, the data stored in the memory data latches is written back into the memory location from which the data was read.

c. Half-Cycle Write. A half-cycle write operation requires that all cores in the addressed memory location must have been previously switched to the one state by performing a half-cycle read operation. During the half-cycle write operation, an 8-bit byte received over the bidirectional memory data bus from the CPU or the DMA is written into the addressed memory location.

d. Full-Cycle Write. During the first half of the full-cycle write operation, all cores in the addressed memory location are switched to the one state. Then, during the second half of the cycle, an 8-bit byte received from the CPU or the DMA is written into the addressed memory location.

S3.1.2 PHYSICAL DESCRIPTION

The 8K memory is contained on a single printed circuit board having dimensions of 8.575 X 12.5 inches. The logic circuits are implemented with medium-scale integration (MSI) integrated circuits. The core memory is implemented with two 4K memory stacks, utilizing lithium ferrite cores, that have an inside diameter of 22 mils. Printed circuit connector P1 installs in the mainframe of the CPU chassis.

S3.1.3 OPERATION SUMMARY

The interface block diagram for the 8K memory is shown in Figure 2-6 of the basic CPU manual. Signals WTXX/, RTXX/, and READ from the CPU initiate and select the four types of read/write operations that can be performed. Detailed timing diagrams showing these signals are provided in Figures 2-23 and 2-24 of the basic CPU manual.

In the 8K memory module, the three most significant bits (M07A/ through M05A/) of the 16-bit address received from the CPU or the DMA, is compared to the states of the three memory select lines (MS3 through MS1). The states of MS3 through MS1 are hardwired into the CPU backplane to contain the code (0 through 7) that is assigned to eack 8K memory module. Signal MS3 is compared with M07A/, MS2 is compared with M06A/, and MS1 is compared with M05A/. If each of these three comparisons indicates that the signals are in opposite states, a bank select match exists, indicating that the address in bits M07A/ through M05A/ is selecting the 8K memory bank. The remaining 13 address bits (M04A/ through M00A/ and N07A/ through N00A/) are used to select one 8-bit location within the 8192-byte memory. During a read operation (full- or half-cycle), data from the accessed memory location is placed on the bidirectional memory data bus to the CPU and the DMA. During a write operation (full- or half-cycle), data from the CPU or the DMA is placed on the bidirectional memory data bus, and is written into the accessed memory location.

The following points are common to the bidirectional memory data bus:

a. The output lines from the memory data (MD) register in the CPU
b. The output lines from the write data register in the DMA channel

c. The output lines from the memory data latches in the 8K memory
d. The input lines to the inhibit current drivers in the 8K memory
e. The input lines to the memory data gates in the CPU
f. The input lines to the read data register in the DMA channel

The eight lines to or from each of these points have the common designation MD00 through MD07. During the read portion of a read or full-cycle write operation, the cores in the addressed memory location are switched to the one state. If a read operation is being performed, the resulting outputs from the point c are supplied to points e and f.

During the write portion of a full-cycle read operation, the outputs from point c are written back into the memory via the lines to point d. During the write portion of a write operation, the data to be written into memory is supplied from point a or b to point d.

S3.2 THEORY OF OPERATION

Figure S3.2-1 is a block diagram of the 8K memory. Four-digit numbers (LD 0460 through LD 0468) referenced in the subparagraphs that follow are logic drawings contained in Logic Diagrams Manual, LD 1010.

Note

Unless otherwise specified, a logical one is represented by a relatively high voltage level, and a logical zero is represented by a relatively low level.

S3.2.1 CORE MEMORY OPERATION

The core memory utilizes a three-wire, coincident current technique. As shown in Figure S3.2-2, three wires are threaded through each core in the 8K memory array. The X and Y address lines are threaded through the core at right angles, and the inhibit/sense line is threaded through the core parallel to the Y address line. Only the eight cores in the addressed memory location receive current on both the X and Y address lines. The vector sum of the two currents is sufficient to switch the core.

All other memory locations receive no current on either address lines, or receive current that is not sufficient to switch the cores.

During the read portion of a read/write operation, the inhibit current to the inhibit/sense line is turned off, and the direction of current flow in the X and Y address lines causes all cores in the addressed memory location (that were previously in the zero state) to be switched to the one state. Those cores which were in the one state remain unchanged. The flux reversal produced when a core is switched from the zero to the one state induces a voltage in the corresponding inhibit/sense line, which is sensed by a sense amplifier. The state of the sense amplifier is then strobed into a memory data latch, whose output is gated to the memory data bus.

During the write portion of a read/write operation, the direction of current flow in the X and Y address lines cause the cores in the addressed memory location to be switched to the zero state, provided that no inhibit current is flowing in the associated inhibit/sense line. If a one is to be written into a core, an inhibit current flow is enabled through the inhibit/sense line for that core. The direction of inhibit current flow is opposite to the current flow in the Y address line, which is threaded through the core parallel to the inhibit/sense line. The effect of the inhibit current is to cancel the current in the Y address line, thus the current is insufficient to switch the core. Since a write operation must be preceded by a read operation that causes all cores in the addressed location to be in the one state, the core remains in the one state at the completion of the write operation if an inhibit current is produced.

As mentioned previously, only cores in the memory location that receive half-currents on both the X and Y address lines will be switched during a read or write operation. In addition, however, the direction of the current vector produced by the vector sum of the two-half currents must be 180 degrees out of phase with the direction of the current vector used to switch the core to the opposite state. If the current vector is only 90 degrees out of phase, the magnitude and direction of the current vector will not be sufficient to switch the cores. This feature permits discrimination between the upper and lower core memory stacks.

Memory address lines N05A/ through N00A/ select one of 64 possible X address current paths through the upper and lower stacks. Memory address lines M03A/ through M00A/,

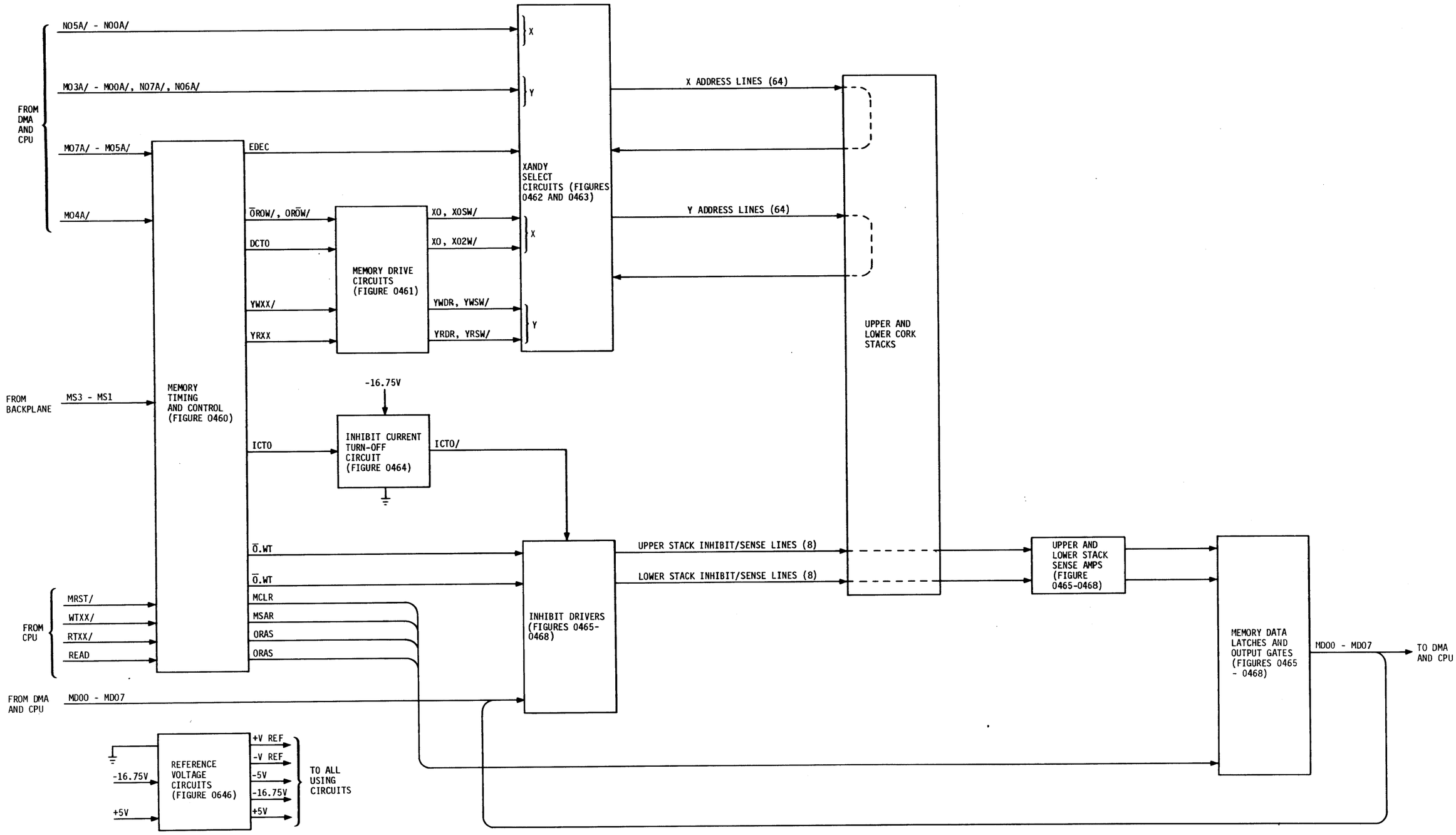


Figure S3.2-1. 8K Memory Block Diagram

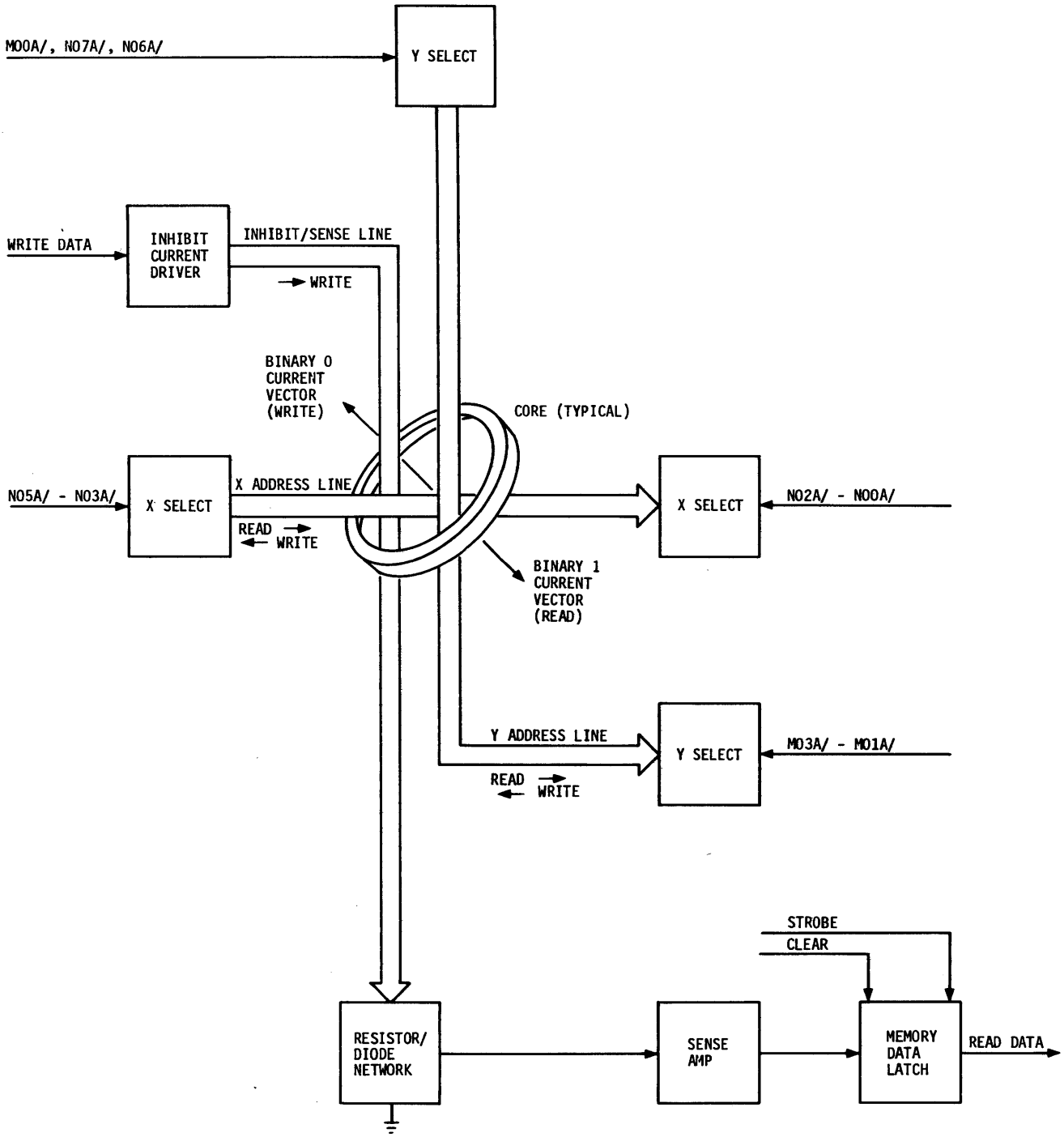


Figure S3.2-2. Core Memory Simplified Block Diagram

N07A/, and N06A/ select one of 64 possible Y address current paths through the upper and lower stacks. Consequently, there are 4096 possible memory locations that can be accessed by these 12 address bits. Since, however, the X and Y address lines are threaded through both the upper and lower stacks, a given address code will actually address one location in the upper stack and one location in the lower stack. However, the current direction in the selected X address current path is controlled by upper/lower stack select line M04A/ such that the resulting current vector is in the proper direction only in the selected stack. In the other stack, the direction of the X address current produces a current vector that is only 90 degrees out of phase with the current vector that places the cores in the opposite state, and the cores in this location are not switched.

Eight inhibit/sense lines (one line for each bit in a memory byte) are threaded through the cores for the corresponding bit positions (MD00 through MD07) in the lower stack, and eight separate inhibit/sense lines are threaded through the cores for the corresponding bit positions in the upper stack. A separate inhibit current driver and sense amplifier is provided for each of the 16 inhibit/sense lines. The sense amplifiers associated with the same bit in both stacks, feed a single memory data latch that is common to both stacks. The read strobe that is produced during a read operation selects the sense amplifiers in the selected stack as the source of the binary data fed to the memory data latch. At the beginning of a read operation, the memory data latch is set to contain a binary one. During the read operation, the latch is set to contain a binary zero if a flux reversal (indicating that the core contained a binary zero) is detected by the selected sense amplifier.

S3.2.1.1 MEMORY TIMING AND CONTROL (Logic Diagram 0460). The occurrence of a low (active) MASTER RESET (MRST/) signal from the CPU produces EDEC/ = 0, which enables the address decoder (LD 0462 and LD 0463). When the MRST/ line is high (inactive), read and write operations are enabled whenever the three memory bank select lines (M07A/ through M05A/) from the CPU or DMA are in the states established by the three hard-wired MEMORY SELECT (MS3 through MS1) lines. Table S3.2-1 contains the states of the three memory bank select lines that select an 8K memory module for each possible state combination of the hard-wired MEMORY SELECT lines.

TABLE S3.2-1. MEMORY BANK SELECT CODES

BANK NO.	HARD-WIRED STATES OF MEMORY SELECT LINES*			CORRESPONDING STATE OF MEMORY BANK SELECT LINES*		
	MS3	MS2	MS1	M07A/	M06A/	M05A/
0	L	L	L	H	H	H
1	L	L	H	H	H	L
2	L	H	L	H	L	H
3	L	H	H	H	L	L
4	H	L	L	L	H	H
5	H	L	H	L	H	L
6	H	H	L	L	L	H
7	H	H	H	L	L	L

*High = relatively high (or open-circuit) voltage; L = relatively low voltage (0 V).

When the correct memory bank select code is not present, MEN = 0 and MOS = 0 are both present, the read and write timing one-shots (YRX and YWX) are both held in the reset state, and read/write operations are disabled.

When the correct memory bank select code is present, MOS = 1 and MEN = 1 both occur, and the following conditions exist:

a. The write timing one-shot (YWX) is enabled.

b. The read timing one-shot (YRX) is enabled.

c. ENABLE DECODER (EDEC/) = 1 exists, disabling the X and Y address decoders (LD 0462 and LD 0463).

d. DRIVE CURRENT TURN-OFF (DCTO) = 1 occurs. The resulting positive-going transition is differentiated, producing a positive pulse that shuts off the drive current to the memory drive circuits (LD 0461) until the positive pulse terminates. The shut-off provides a switching delay.

e. INHIBIT CURRENT TURN-OFF (ICTO) = 1 exists.

With these conditions existing, the 8K memory is prepared for executing a memory read or write operation, as described in the following subparagraphs.

S3.2.1.2 SEQUENCE OF ACTIONS FOR A MEMORY READ OPERATION. When a memory read operation is to be performed, the CPU produces a 350-nanosecond-wide, negative-going READ PULSE (RTXX/). The occurrence of this pulse produces the following sequence of actions:

a. The read timing one-shot is triggered by RTXX/ = 0, producing a 345-nanosecond-wide, negative-going YRXX/ pulse. On the negative-going loading edge of this pulse, the following actions occur:

   1. EDEC/ = 0 occurs, enabling the X and Y address decoders (LD 0462 and LD 0463), which decode the memory address.

   2. If M04A/ = 1 exists (select lower stack), multiplexer Z05 produces X WRITE/READ PULSE ($\overline{\text{O}}$ROW/) = 0. If M04A/ = 0 exists (select upper stack), multiplexer Z05 produces X READ/WRITE PULSE (OR$\overline{\text{O}}$W/) = 0. The occurrence of $\overline{\text{O}}$ROW/ = 0 completes a current path from X driver line X$\overline{\text{O}}$ (LD 0461) through the X Select circuit (LD 0462) and the two core stacks to X sink line X$\overline{\text{O}}$SW/ (LD 0461) to select a location in the lower core stack. The occurrence of OR$\overline{\text{O}}$W/ = 0 completes a current path from X driver line X0 (LD 0461) through the X Select circuit (LD 0462) and the two core stacks to X sink line XOSW/ (LD 0461) to select a location in the upper core stack.

   3. DCTO = 0 occurs.

   4. A current path is completed from Y driver line YRDR (LD 0461) through the Y Select circuit (LD 0463) and the two core stacks to Y sink line YRSW/ (LD 0461).

   5. The memory data latch clear (MCLR) one-shot is triggered, producing a 200-nanosecond-wide, negative-going MCLR/ pulse that sets all memory data latches (LD 0465 through LD 0468) to the one state.

b. When the negative-going pulse from the MCLR one-shot terminates, the read amplifier strobe (RAS) one-shot is triggered, producing an adjustable 50- to 70-nanosecond-wide, positive-going RAS pulse (RAS = 1).

c. While RAS = 1 exists, multiplexer Z05 produces either $\overline{\text{O}}$RAS = 1 (READ AMP STROBE UPPER STACK) or ORAS = 1 (READ AMP STROBE LOWER STACK), depending upon the state of stack select line M04A/. The occurrence of the strobe causes the outputs from the sense amplifiers (LD 0465 through LD 0468) in the selected stack to be strobed into the memory data latches.

d. When the YRXX/ pulse terminates, DCTO = 1 again occurs. The resulting positive-going transistor is differentiated, producing a positive pulse that shuts off the drive current to the memory current drivers (LD 0461) until the positive pulse terminates.

If memory read is being performed as part of a full- or half-cycle read operation, as opposed to a full-cycle memory write operation, the CPU also produces READ ENABLE

(READ) = 1. The occurrence of READ = 1 produces MEMORY SELECT AND READ (MSAR) = 1, which allows the contents of the memory data latches to be gated to the bidirectional memory data bus (MD00 through MD07), as shown in logic diagrams 0465 through 0468. If a full-cycle memory read operation is being performed, the memory read is followed by a memory write, during which the binary data contained in the memory data latches is written back into the addressed memory location.

S3.2.1.3 SEQUENCE OF ACTIONS FOR A MEMORY WRITE OPERATION. When a memory write operation is to be performed, the CPU produces a 350-nanosecond-wide, positive-going WRITE PULSE (WTXX/). The occurrence of this pulse triggers the write timing one-shot, producing a 345-nanosecond-wide, positive-going YWXX pulse. On the positive-going edge of this pulse, the following actions occur:

a. EDEC/ = 0 occurs, enabling the X and Y address decoders (LD 0462 and LD 0463), which decode the memory address.

b. DCTO = 0 and ICTO = 0 both occur.

c. If M04A/ = 1 exists (select lower stack), multiplexer Z05 produces X READ/WRITE PULSE ($\text{OR}\overline{\text{OW}}$) = 0. If M04A/ = 0 exists (select upper stack), multiplexer Z05 produces X WRITE/READ PULSE ($\overline{\text{O}}$ROW/) = 0. The occurrence of $\text{OR}\overline{\text{OW}}$ = 0 completes a current path from X sink line XOSW/ (LD 0461) through the X Select circuit (LD 0462) and the two stacks to X driver line X0 (LD 0461) to select a location in the lower core stack. The occurrence of $\overline{\text{O}}$ROW/ completes a current path from X sink line $\overline{\text{XO}}$SW through the X Select circuit (LD 0462) and the two core stacks to X driver line $\overline{\text{XO}}$ (LD 0461) to select a location in the upper core stack.

d. A current path is completed from Y driver line YWDR (LD 0461) through the Y Select circuit (LD 0463) and the two core stacks to Y sink line YWSW/.

e. Depending on the state of stack select line M04A/, either $\overline{\text{O}}$,WT = 1 (write lower stack) or O.WT = 1 (write upper stack) is produced, gating the data on the memory data bus (MD00 through MD07) to the inhibit current drivers for the selected stack.

When the YWXX pulse terminates, DCTO = 1 and ICTO = 1 both occur. These positive-going trannsitions are differentiated, producing positive pulses that shut off the drive current to the memory current drivers (LD 0461) and to the inhibit current drivers (LD 0465 through LD 0468) until the positive pulses terminate.

S3.2.2 MEMORY DRIVE CIRCUITS (Logic Diagram 0461)

The memory drive circuits consist of current drivers for the X and Y select drivers in logic diagrams 0462 and 0463. Except when a positive-going transition of the

DCTO signal produces a differentiated positive pulse input to stage Q27, drive current for the drive circuits is enabled. The occurrence of $\overline{OROW}$ = 0 activates the driver stages for lines $\overline{XO}$ and $\overline{XOSW}$/. The occurrence of $OR\overline{OW}$ = 0 activates the driver stages for lines XO and XOSW/. The occurrence of YWXX/ = 0 activates the driver stages for lines YWDR and YWSW/. The occurrence of YRXX/ = 0 activates the driver stages for lines YRDR and YRSW/.

S3.2.3 X AND Y SELECT CIRCUITS (Logic Diagrams 0462 and 0463)
When EDEC/ = 0 occurs, the X and Y address decoders are enabled. The X address decoder selects one of 64 possible X current paths, and the Y address decoder selects one of 64 possible Y current paths. The direction of current flow in the selected X address current path is determined by whether the $\overline{XO}$ - $\overline{XOSW}$/ or the XO - XOSW/ memory driver pair is activated in the memory drive circuits (LD 0461). The direction of current flow in the selected Y address current path is determined by whether the YWDR - YWSW/ or the YRDR - YRSW/ memory driver pair is activated in the memory drive circuits. Current in the selected X and Y current paths flows through the cores in the upper and lower stacks. As discussed in Paragraph 5.1, these X and Y currents will select a single memory location in either the upper or lower core stack.

S3.2.4 PRIORITY/SELECT, MEMORY CLAMP AND POWER (Logic Diagram 0464)
This logic diagram sheet contains miscellaneous reference voltage circuits and filter capacitors, and also includes the inhibit current turn-off (ICTO) circuit and the PRIORITY IN (PRIN/) and SELECT IN (SELI/) jumpers to the backplane.

When ICTO/ = 1 occurs, the resulting positive voltage transition is differentiated, producing a positive-going pulse that disables the inhibit current drivers (LD 0465 through LD 0468) by forcing the ICTO/ line to a negative voltage. When the positive pulse terminates, the ICTO/ line returns to 0 volt, and the inhibit current drivers are enabled.

S3.2.5 MEMORY READ/WRITE CIRCUITS (Logic Diagrams 0465 through 0468)
Logic diagrams 0465 through 0468 show eight identical read/write circuits for the eight bits of a memory byte. In the descriptions that follow, the read/write circuit for bit 0 (MD00) is described.

S3.2.5.1 INHIBIT DRIVER CIRCUIT. During a memory write operation (refer to Paragraph 5.2.2), signal $\overline{0}$.WT = 1 (write low stack) or 0.WT = 1 (write lower stack) occurs, gating bit MD00 to the inhibit driver circuit for the selected stack. Inhibit drivers Q94 and Q95 are enabled when the ICTO/ line is at 0 volt. When signal ICTO/ is negative, the inhibit drivers are cut off. If MD00 = 1 exists, gate Z23 goes low, and a current pulse is produced in the primary winding of transformer T1, causing a voltage to be induced in a secondary winding. If the inhibit drivers are enabled, the voltage induced in the secondary winding turns on inhibit driver Q94 (lower stack driver) or Q95 (upper stack driver), producing a current pulse in the inhibit/sense line for bit zero of the selected stack. The current pulse prevents the bit zero core (in the addressed memory location) from being switched from the one state to the zero state during the write operation.

S3.2.5.2 SENSE AMPLIFIER. During a memory read operation (refer to Paragraph 5.2.1), cores in the addressed memory location are switched to the one state. If the core was previously in the zero state, the resulting flux reversal in the inhibit/sense line produces a differential voltage at the input of the sense amplifier associated with the selected stack, and the sense amplifier output goes high. If the core was already in the one state, no flux reversal occurs, and the sense amplifier output remains low.

S3.2.5.3 MEMORY DATA LATCH. At the beginning of a memory read operation (refer to Paragraph 5.2.1), MCLR/ = 0 occurs, setting the memory data latch to the one state. Then, while the sense amplifier outputs are still active, either $\overline{\text{ORAS}}$ = 1 (read amp strobe lower stack) or ORAS = 1 (read amp strobe upper stack) is produced, selecting the output of the sense amplifier associated with the selected stack as the input to the memory data latch. If the output of the selected sense amplifier is high, the memory data latch is reset. If the sense amplifier output is low, the memory data latch remains in the set state.

S3.2.5.4 MEMORY DATA GATE. During a half- or full-cycle read operation, MSAR = 1 occurs, gating the output of the memory data latch to the bidirectional memory data bus (MD00 through MD07). If the memory data latch is set, MD00 = 1 is produced. If the memory data latch is reset, MD00 = 0 is produced.

S3.3 PARTS LIST

All electronic parts on the 8K memory board are illustrated in Figure 5-7 and are listed and described in the parts list, section 5.

Supplement 4

TABLE OF CONTENTS

Supplement 4

REMOTE POWER SUPPLY

S4.1 INTRODUCTION

This supplement provides coverage for the Remote Power Supply. Schematic diagrams for the two power supplies are contained on sheets 2010 through 2030 in Logic Diagrams Manual, LD 1010.

S4.1.1 PURPOSE AND FUNCTION

The remote power supply is designed for rack mounting outside the CPU chassis. The power supply provides the following regulated dc voltages to the circuits installed in the CPU chassis:

a. +12 volts dc (adjustable ±5%) @ 3.0 amperes maximum
b. +5 volts dc (adjustable from 4.75 volts to 5.5 volts) @ 20.0 amperes maximum
c. -16.75 volts dc (adjustable ±2%) @ 5.0 amperes maximum

The power supply is capable of maintaining these regulated dc voltages within an envelope of ±4 percent while subject to the worst simultaneous conditions of line voltage, line frequency, temperature, dc load, and ripple. Overload protection is provided, which limits the current output to no more than 120 percent of the rated maximum output current for each supply voltage. The +5 volts dc supply is also provided with an overvoltage shutdown device that shuts down the supply when the voltage rises to an excessive value.

The power supply is capable of operating from an ac line voltage of 105, 115, 210, or 230 volts rms, with a frequency range of 47 Hz to 63 Hz. The regulated dc voltages are maintained within their specified limits while the ac line voltage is varied ±10 percent from its nominal value. The ac power input to the power supply is switched on by a relay when the following conditions are satisfied:

a. For the remote power supply only, the front panel PWR ON/OFF switch must be set to PWR ON.

b. The key-lock switch on the CPU front panel must be set to either the ON or LOCK position.

The power switching relay is also equipped with an auxiliary set of normally open contacts (rated at 10 amperes, 120 volts ac non-inductive) to permit power switching control of an auxiliary power supply or peripheral device. An ac convenience outlet is also provided for auxiliary equipment.

The remote power supply is equipped with an auxiliary RESET switch that, when actuated, applies a master reset pulse to the CPU.

The power supply also includes an ac power failure detection circuit that is used with the power failure detection and automatic restart feature of the CPU. When the ac input voltage falls more than 10 percent below the nominal input voltage, a power fail indication is supplied to the CPU. When the ac voltage returns to normal, the power fail indication terminates to notify the CPU that power has been restored.

S4.1.2 PHYSICAL DESCRIPTION

The remote power supply has dimensions of 8.75 X 19 X 10.5 inches, and weighs approximately 25 pounds. A terminal board (TB1), located on the left side of the unit, provides the connection point for the power supply interface cable (part number A20001089). The other end of this cable connects to a printed circuit extender card (part number A20001057) that installs into the left-most slot of the backplane in the CPU chassis. The interface cable and extender card provide the means for interconnecting the remote power supply to the backplane. The rear of the unit contains the ac convenience outlet for connecting auxiliary equipment. The front of the unit contains the ac power receptacle, which is connected to the ac power source by means of a power cable, and also contains the main power fuse, the PWR ON/OFF switch, and the auxiliary RESET switch. The front panel includes two handles to permit easy removal 8and installation in the rack. The inlet for the blower, which provides cooling air for the power supply electronics, is also located on the front panel.

Mounted to the rear of the front panel are the power supply electronics, including two heat sinks and two printed circuit boards (A1 and A2). Screwdriver adjustments for the power fail circuit and for the three supply voltages are mounted on the outer edge of printed circuit boards A1 and A2.

S4.2 THEORY OF OPERATION

S4.2.1 INPUT POWER DISTRIBUTION

Ac power from receptacle J1 is applied through main power fuse F1 to PWR ON/OFF switch S1. When this switch is set to PWR ON, the ac power is supplied to the primary winding of transformer T2 and to the normally open contacts of power switching relay K1. Transformer T2, in conjunction with rectifier diode CR7 and filter capacitor C11, develops the 12-volt dc power for switching relay K1. When a ground is applied to the RPSS/ line from the key-lock switch on the CPU front panel, the current path for relay K1 is completed, and the relay energizes, applying power to blower B1, the ac convenience outlet, and power transformer T1. Thermal protection switch S2, mounted on the heat sink, interrupts ac power to transformers T1 and T2 when the temperature of the heat-sink-mounted transistors becomes excessive.

Terminal board TB2 is connected to the taps in the primary winding of power transformer T1, permitting jumper connections to be made for operation with a 105-, 115-, 210-, or 230-volt ac power source. The jumper connections for each voltage are as follows:

	Connections							
AC Input Voltage	TAP to:	TB2-2 to:	TB2-3 to:	TB2-5 to:	T2-1 to:	T2-2 to:	T2-3 to:	T2-3 to:
105	TB2-6	TB2-1	None	TB2-6	S1-3	T2-1	T2-4	TB2-1
115	TB2-4	TB2-1	TB2-4	None	S1-3	T2-1	T2-4	TB2-1
210	TB2-6	TB2-5	None	TB2-2	S1-3	T2-3	T2-2	TB2-1
230	TB2-4	TB2-3	TB2-2	None	S1-3	T2-3	T2-2	TB2-1

S4.2.2 +12-VOLT POWER SUPPLY

The full-wave-rectified voltage produced by rectifier diodes CR5 and CR6 is filtered by capacitor C1 and supplied through series regulator Q1 as the +12-volt output. The current through series regulator Q1 is controlled by the +12-volt regulator circuit on printed circuit board A1. Variable resistor R15 on board A1 permits adjustment of the regulated +12-volt output level.

S4.2.3 +5-VOLT POWER SUPPLY

The full-wave-rectified voltage produced by rectifier diodes CR1 and CR2 is filtered by inductor L1 and capacitor C2, and is supplied through a series regulator circuit

as the +5-volt output. The series regulator circuit consists of transistors Q2 through Q4 and Q7, which are controlled by transistor Q5. Transistor Q5, in turn, is controlled by the +5-volt regulator circuit on printed circuit board A1. Variable resistor R25 on board A1 permits adjustment of the regulated +5-volt output level. Variable resistor R27 on board A1 permits adjustment of the overvoltage protection circuit. When the +5-volt output rises above a level set by R27, a crowbar circuit is activated, shutting off the +5-volt output.

S4.2.4 -16.75-VOLT POWER SUPPLY

The full-wave-rectified voltage produced by rectifier diodes CR3 and CR4 is filtered by capacitor C3 and supplied to the shunt regulator. The shunt regulator circuit, consisting of transistors Q6 and Q8, regulates the -16.75-volt output level by varying the current in the shunt path. The shunt regulator is controlled by the -16.75-volt regulator circuit on board A1. Variable resistor R5, on board A1, permits adjustment of the regulated -16.75-volt output level.

S4.2.5 POWER FAIL DETECTION CIRCUIT

Board A2 contains the power fail protection circuit. This circuit monitors the magnitude of the ac voltage present in the secondary winding of power transformer T1. When this voltage drops more than 10 percent below its nominal value, the PWRD/ output supplied to the CPU switches from +5 volts to +0.25 volts, notifying the CPU that a power failure has occurred. When the ac voltage is restored to its normal value, the PWRD/ output returns to +5 volts.

S4.3 PARTS LIST

The physical location of the electronic parts on the power supply is shown in figures 5-8 and 5-9; the parts list is in section 5.

Supplement 6

LOCAL REMOTE SINGLE CHANNEL CONTROLLER

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Supplement 6

LOCAL/REMOTE SINGLE CHANNEL CONTROLLER

S6.1.1 INTRODUCTION

This supplement describes the Local/Remote Single Channel Controller (L/RSCC) shown in figure S6.1-1. The L/RSCC is a plug-in printed circuit board assembly which may be used in any of the four locations from J13 through J16, as shown on page iv of LD 1010.

S6.1.2 PURPOSE AND FUNCTION

The L/RSCC is interconnected to the backplane through one of the four 65-pin, double-row connectors J13 through J16. It also connects one local or remote input device through a cable attached to either J2 or J3 (but not both J2 and J3. When operating as a local SCC, the L/RSCC may interface with any RS-232-C compatible device (such as a Video Display Terminal) or any teletype that operates in a 20-milliampere current loop. As a remote SCC, the L/RSCC may interface with a Dedicated Asynchronous Communications Channel, a type 103 Modem, or a type 202 Modem.

Serial data from the communications device is converted to an 8-bit parallel form for transmittal to the CPU via the byte I/O interface. Parallel data received from the CPU via the byte I/O interface is converted into a serial form for transmittal to the communications device. Data transfers to and from the CPU can be executed in either the program controlled mode or the concurrent (block transfer) mode. The controller is capable of full-duplex (simultaneous sending and receiving) operation. Upon command from the CPU, the controller also supplies controller and modem status information to the CPU.

S6.1.3 PHYSICAL DESCRIPTION

The L/RSCC is rectangular, 12.5 inches long by 8.75 inches wide. Components and jumpers are mounted only on one side of the board. Figure S6.1-1 shows areas of the board blocked out to show the location of circuit elements used for the

functions performed by this assembly. Occasionally there are isolated circuit elements located outside the area indicated and the integrated circuits located in the area marked "miscellaneous" are used for several functions. Locating components on the L/RSCC is facilitated by a number/letter matrix: 1 through 9, by A through L (excluding the letters G and I). Symbols on the logic diagrams in LD 1010 identify matrix location for integrated circuit elements.

S6.1.4 FUNCTIONAL SPECIFICATIONS

Table S6.1-1 lists the electrical and functional specifications of the L/RSCC.

S6.1.5 OPERATION SUMMARY

Figure S6.1-2 is an interface block diagram showing the interface exchange between the L/RSCC and the CPU through the backplane and the communications device through a cable connected to J2 or J3. The output data bus from the CPU is used to send device orders, function commands, and data for transmission from the CPU to the controller. The input data bus to the CPU is used to send address bytes (to confirm receipt of a CPU transmission), status bytes, and received data bytes from the controller to the CPU.

Three I/O mode control lines, a clock line, and three control lines control communications between the CPU and the controller. Transmitted and received data to and from the communications device is in a serial form. Four control lines from the controller to the modem provide control of modem operation. Six status lines to the controller indicate modem status, which is transmitted (upon request) to the CPU.

The controller contains separate registers for storing data received from the communications device, and for storing data to be transmitted to the communications device. Because separate registers are provided, data transfers between the controller and the communications device can be accomplished in either the full- or half-duplex mode.

Data transfers between the controller and the CPU can be accomplished in either the program-controlled mode, or in the concurrent (block transfer) mode. In the program-controlled mode, the transfer of each data byte from the CPU to the controller must be initiated by the CPU under program control.

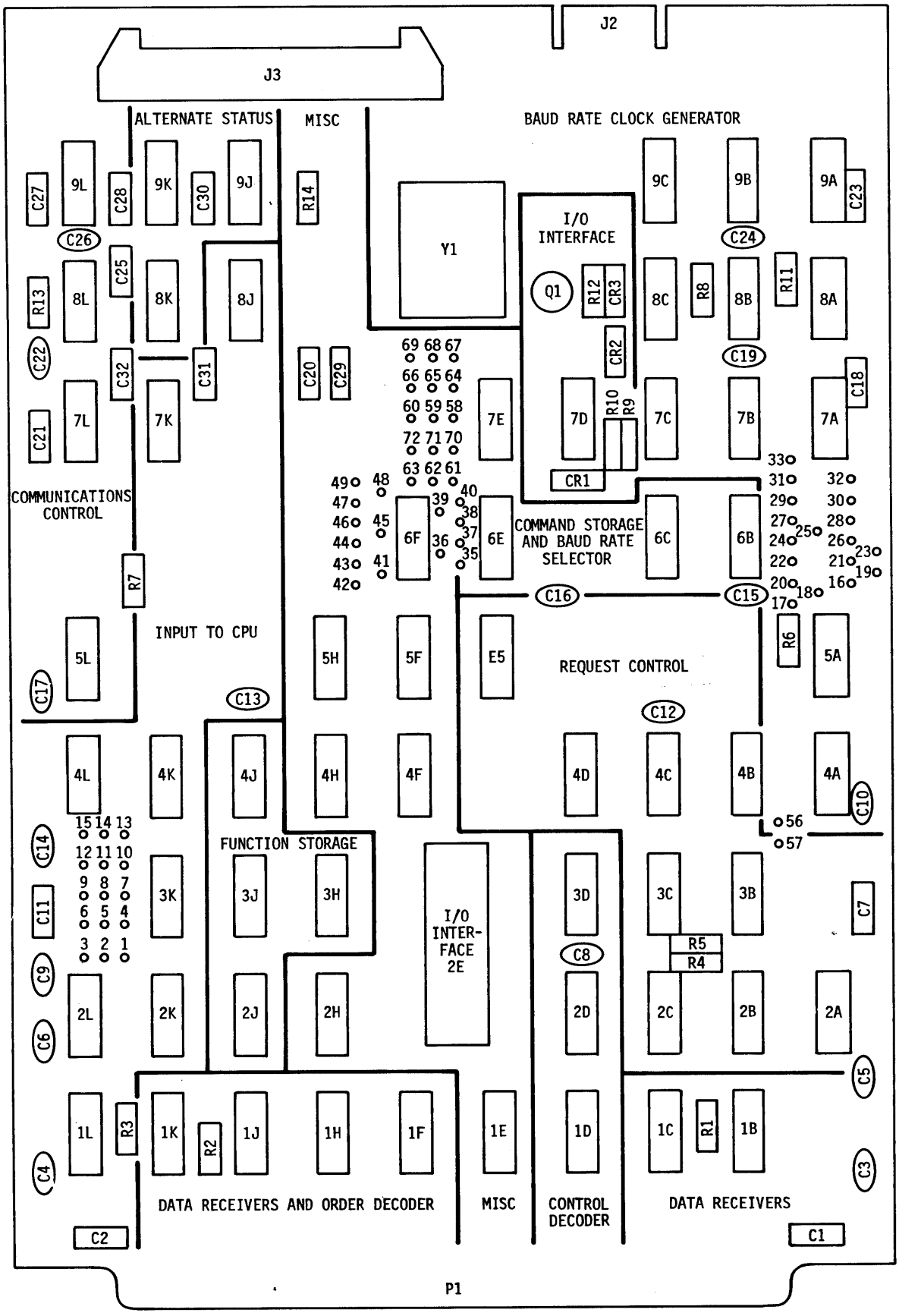


Figure S6.1-1. Local/Remote Single Channel Controller Circuit Functions

Table S6.1-1. Specification Data

Parameter	Specification
Input Power Requirements	+5v at 750 ma; -16.75v at 250 ma; +12v at 200 ma.
Temperature Ranges:	Operation: 0°C to 50°C. Storage: -40°C to 80°C.
Communications Device Interface	RS-232-C or 20-ma current loop
Character Format	May be selected under program or may be fixed, as selected by jumper connections.
Character Length	5, 6, 7 or 8 data bits
Stop Bits	1 or 2 stop bits
Parity Bit	May be enabled or disabled. If enabled, either odd or even parity may be selected.
Data Rate	May be selected under program, or may be fixed as selected by jumper connections. Any of the following baud rates may be selected: 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, or 9600.
Received Data Error Detection	The following types of errors are detected in data received from the communications device: Parity Error, Framing Error, Overrun Error.
I/O Modes	Program-controlled or concurrent (block transfer) mode.
Line Modes	Asynchronous, full- or half-duplex, with completely independent transmit and received circuits.
Line Control	Auto-Answer, Line Turnaround, and Supervisory Reverse Channel.
Interrupts	Interrupt on received character ready in program-controlled mode. Interrupt on line conditions in any mode. Interrupt on disconnect in concurrent mode.
Logic Levels:	Internal: Binary 1 = +5v; binary 0 = 0v CPU Interface: Binary 1 = 0v; binary 0 = +3v
Communications Device Interface Levels	Binary 1 (mark) = -12v; binary 0 (space) = +12v

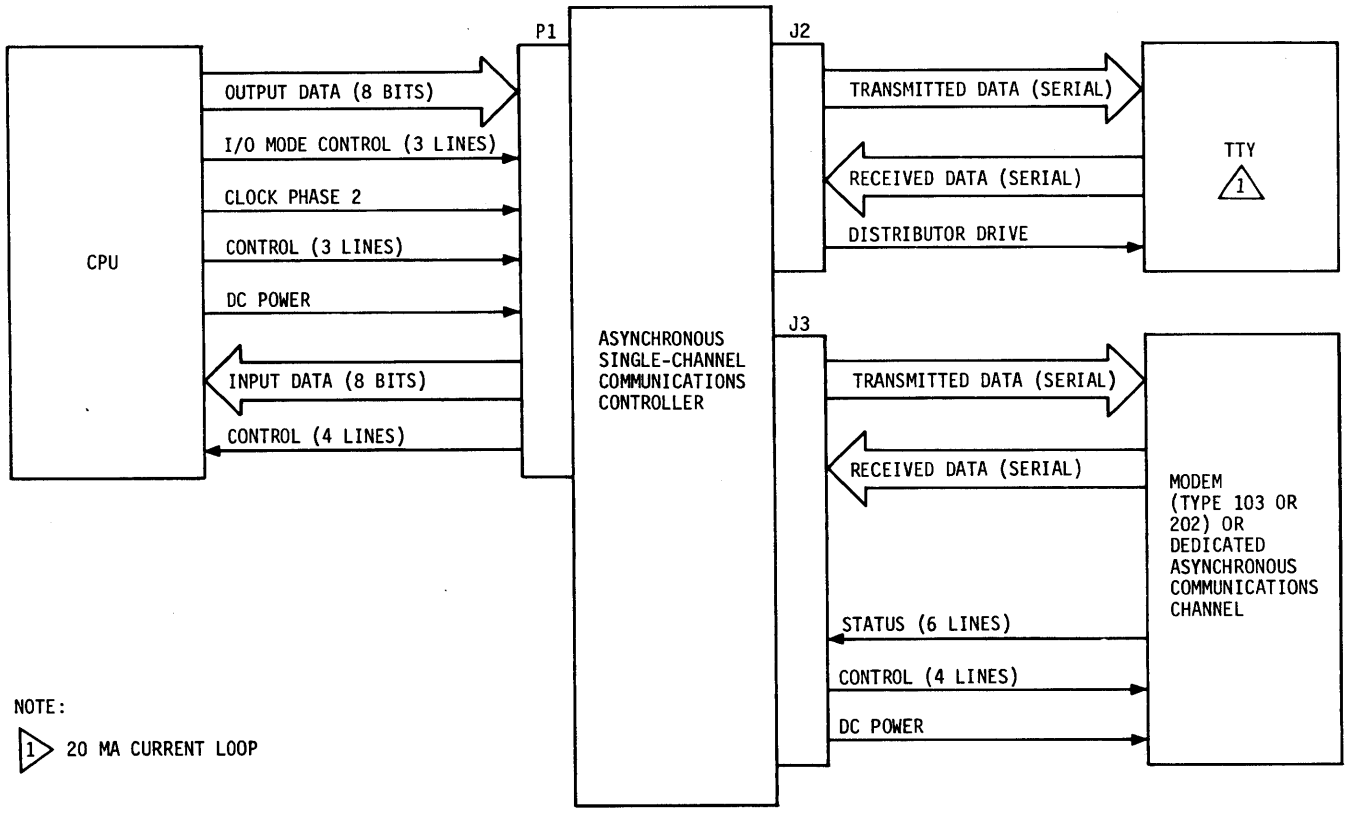


Figure S6.1-2. Interface Block Diagram

Normally, the transfer of each data byte from the controller to the CPU must be initiated by an interrupt request from the controller to the CPU.

When data transfers are executed in the concurrent mode, the CPU sets up a block of memory storage locations within the CPU, corresponding to the block of data, and commands the controller to operate in the concurrent mode. After these steps are accomplished, data transfers may be executed without program intervention. In the concurrent mode, each data transfer to or from the controller must be preceded by a concurrent I/O request from the controller. When the controller is operating in the concurrent output mode, data words from the assigned memory block are transferred in sequence to the controller for transmittal to the communications device. When the controller is operating in the concurrent input mode, data words from the communications device are transferred to the CPU, and stored in the assigned memory block.

S6.1.5 CPU INPUT/OUTPUT DATA FORMATS AND I/O MODE CONTROL CODES

Figure S6.1-3 defines the I/O mode control codes supplied from the CPU to the controller over the three-line I/O mode control bus. Figure S6.1-4 shows the for formats of all types of bytes transferred from the CPU to the controller over the output data bus. Figure S6.1-5 shows the formats of all types of bytes transferred from the controller to the CPU over the input data bus.

S6.1.6.1 I/O MODE CONTROL

The I/O mode of the controller is determined by the three-bit code on the I/O mode control lines (IO1X through IO3X). The definition and function of each applicable code is given in figure S6.1-3. Each transfer over the input or output data bus must be preceded by the appropriate I/O mode control code, as follows:

a. Control Out (1_8) - This code (COXX) precedes each device order byte placed on the output data bus.

b. Data Out (2_8) - This code (DOXX) precedes any byte except a device order byte that is placed on the output data bus in either the program-controlled or concurrent mode of operation.

c. Data In (6_8) - This code (DIXX) precedes any byte that is placed on the input data bus.

d. Concurrent Acknowledge (4_8) - When the controller is operating in the concurrent mode, a concurrent I/O request must be issued to the CPU for each data transfer. The CPU responds by issuing concurrent acknowledge (CACK). If no higher-priority device is requesting concurrent mode service, the controller responds to concurrent acknowledge by placing an identifying address byte on the input data bus. The CPU responds to the address byte by issuing data in (concurrent input mode) or data out (concurrent output mode).

e. Interrupt Acknowledge (5_8) - When the controller issues an interrupt request to the CPU, the CPU responds by issuing interrupt acknowledge (IACK). If no higher-priority device is requesting interrupt service, the controller responds to interrupt acknowledge by placing an identifying address byte on the input data bus. The CPU responds by performing the interrupt processing routine for the controller.

S6.1.6.2 DEVICE ORDER BYTES

The device order byte (figure 6.1-4) contains a device order code in bits 7 (MSB) through 5 (LSB), and a device address in bits 4 (MSB) through 0 (LSB). The controller responds to the device order byte only if the device address code matches

IO3X	IO2X	IO1X	OCTAL EQUIVALENT	I/O MODE
0	0	1	1	CONTROL OUT - NOTIFIES CONTROLLER THAT A DEVICE ORDER IS ON OUTPUT DATA BUS
0	1	0	2	DATA OUT - NOTIFIES CONTROLLER THAT A DATA, FUNCTION COMMAND, OR COMMUNICATIONS CONTROL BYTE IS ON OUTPUT DATA BUS
1	0	0	4	CONCURRENT ACKNOWLEDGE - INDICATES THAT CPU HAS ACKNOWLEDGED RECEIPT OF A CONCURRENT I/O REQUEST FROM CONTROLLER
1	0	1	5	INTERRUPT ACKNOWLEDGE - INDICATES THAT CPU HAS ACKNOWLEDGED RECEIPT OF AN INTERRUPT REQUEST FROM CONTROLLER
1	1	0	6	DATA IN - COMMANDS CONTROLLER TO PLACE STATUS DATA OR RECEIVED DATA ON INPUT DATA BUS TO CPU

Figure S6.1-3. I/O Mode Control Codes

OUTPUT DATA WORD TYPE	OUTPUT DATA BIT NO. 0 (LSB)	1	2	3	4	5	6	7 (MSB)
	LSB	2ND LSB	3RD LSB	2ND MSB	MSB	LSB	2ND LSB	MSB
DEVICE ORDER BYTE (PRECEDED BY I/O MODE CONTROL CODE 1_8 - CONTROL OUT)	1	0	0	0	0	DEVICE ORDER: 0_8 = DATA TRANSFER; 1_8 = FUNCTION COMMAND; 2_8 = CONCURRENT INPUT; 3_8 = ARM INTERRUPT; 4_8 = CONCURRENT DISCONNECT; 5_8 = CONCURRENT OUTPUT; 6_8 = DISARM INTERRUPT; 7_8 = ALTERNATE FUNCTION COMMAND		
	STANDARD DEVICE ADDRESS (MAY BE CHANGED BY CHANGING JUMPERS)							
	LSB	2ND LSB	MSB	P1	SBS	WLS2 (LSB)	WLS1 (MSB)	EPE
FUNCTION COMMAND (PRECEDED BY FUNCTION COMMAND DEVICE ORDER BYTE AND I/O MODE CONTROL CODE 2_8 - DATA OUT)	BAUD RATE: 0_8 = 110 BAUD (STD) OR 75 BAUD; 1_8 = 150 BAUD; 2_8 = 134.5 BAUD; 3_8 = 300 BAUD (STD) OR 600 BAUD; 4_8 = 1200 BAUD; 5_8 = 1800 BAUD; 6_8 = 2400 BAUD (STD) OR 4800 BAUD; 7_8 = 9600 BAUD			PARITY ENABLE: 1 = ENABLE, 0 = DISABLE	NO. OF STOP BITS: 0 = 1 BIT, 1 = 2 BITS	CHARACTER LENGTH: 00 = 5 BITS, 10 = 6 BITS, 01 = 7 BITS, 11 = 8 BITS		ODD/EVEN PARITY SELECT: 0 = ODD PARITY, 1 = EVEN PARITY
	LSB							MSB
TRANSMIT DATA (PRECEDED BY I/O MODE CONTROL CODE 2_8 - DATA OUT)	DATA							

Figure S6.1-4. CPU Output Data Formats

INPUT DATA WORD TYPE	INPUT DATA BIT NO. 0 (LSB)	1	2	3	4	5	6	7 (MSB)
ADDRESS BYTE (PRECEDED BY SELECT IN AND EITHER I/O MODE CONTROL CODE 4_8-CONCURRENT ACK. OR 5_8-INTERRUPT ACK.)	0	1 (STD)	0 (STD)	0 (STD)	0 (STD)	0 (STD)	0	1 = CONCURRENT OUTPUT TRANSFER; 0 = NOT CONCURRENT OUTPUT TRANSFER
		THE CODE IN THESE BITS CAN BE CHANGED BY CHANGING JUMPERS						
	BLKM	DR	THRE	OE	FE	PE	-	TAST
CONTROLLER STATUS BYTE (PRECEDED BY FUNCTION COMMAND DEVICE ORDER AND I/O MODE CONTROL CODE 6_8 - DATA IN)	BLOCK TRANSFER MODE: 1 = NOT BLOCK MODE, 0 = BLOCK MODE	INPUT WORD READY: 1 = READY, 0 = NOT READY	READY TO RECEIVE TRANSMIT WORD: 1 = READY, 0 = NOT READY	OVERRUN ERROR: 1 = ERROR, 0 = NO ERROR	FRAMING ERROR: 1 = ERROR, 0 = NO ERROR	PARITY ERROR: 1 = ERROR, 0 = NO ERROR	0 (NOT USED)	TEST ALTERNATE STATUS: 1 = TEST, 0 = DON'T TEST
	LSB							MSB
RECEIVED DATA (PRECEDED BY I/O MODE CONTROL CODE 6_8 - DATA IN)	DATA							

Figure S6.1-5. CPU Input Data Formats

the wired-in address assigned to the controller. The standard address is 00001, which may be changed by changing jumper connections on the circuit board.

The functions of the device order codes are as follows:

a. Data Transfer (0_8) - Used to initiate a one-byte input or output data transfer in the program-controlled mode. The decode is labeled DATX.

b. Function Command (1_8) - Notifies the controller that either a function command is to be stored, or a controller status byte is to be issued. The decode is labeled FUNC. If the function command device order is followed by Data Out, the function command on the output data bus is stored in the controller. If the function command device order is followed by Data In, the controller status byte is placed on the input data bus.

c. Concurrent Input (2_8) - Commands the controller to enter the concurrent input mode, in which a block transfer of data from the controller to the CPU is performed. Also, the interrupt is armed. The decode is labeled CCIX.

d. Arm Interrupt (3_8) - This device order is used in the program-controlled mode of data transfer when interrupt requests are to be enabled. The decode is labeled ARM.

e. Concurrent Disconnect (4_8) - When the controller is in the concurrent input or concurrent output mode and the block transfer has been completed, the CPU issues the concurrent disconnect device order. The controller responds by terminating the concurrent mode. If the interrupt is armed, an interrupt request is issued to the CPU. The decode is labeled DISC.

f. Concurrent Output (5_8) - Commands the controller to enter the concurrent output mode, in which a block transfer of data from the CPU to the controller is performed. Also, the interrupt is armed. The decode is labeled CCOX.

g. Disarm Interrupt (6_8) - This device order follows a concurrent input or concurrent output device order when interrupt operation is not desired. The decode is labeled DSM.

h. Alternate Function Command (7_8) - Notifies the controller that a communications control byte is to be stored, or that a modem status byte is to be issued. The decode is labeled DC07. If the alternate function command device order is followed by Data Out, the communications control byte on the output data bus is stored in the controller. If the alternate function command is followed by Data In, the modem status byte is placed on the input data bus.

S6.1.6.3 FUNCTION COMMAND BYTE

As shown in figure S6.1-4, the function command byte establishes the format and data transfer rate of the data transferred between the controller and the communications device. A three-bit code in bits 2 (MSB) through 0 (LSB), selects one of

eight baud rates for the data transfers. Bits 3 and 7 define the parity checking (if any) performed in the controller. When bit 3 is a 1 (parity enable), bit 7 determines whether odd (bit 7 = 0) or even (bit 7 = 1) parity checking is performed. Bits 6 (MSB) and 5 (LSB) contain a two-bit code that defines the character length (5, 6, 7 or 8 bits) of the characters transferred between the controller and the communications device. Bit 4 determines whether each character is followed by 1 (bit 4 = 0) or 2 (bit 4 = 1) stop bits.

The function command byte is stored in the controller, and establishes the data rate and character structure until another function command byte is received.

S6.1.6.4 TRANSMIT DATA BYTES

In either the program-controlled mode or the concurrent mode, transmit data bytes (figure S6.1-4) are sent over the output data bus to the controller for transmittal to the communications device.

S6.1.6.5 ADDRESS BYTE

When the CPU acknowledges an interrupt request or a concurrent I/O request from the controller, and if no higher-priority device is requesting service, the controller places an address byte (figure S6.1-5) on the input data bus to the CPU. The state of bit 7 of the address byte is determined by whether or not the controller is in the concurrent output mode. If the controller is in the concurrent output mode (block transfers from the CPU to the controller), bit 7 is a 1; if not, bit 7 is a 0. Bit 6 of the address byte is always 0. Bits 5 (MSB) through 0 (LSB) contain an address code that is twice the device order address (refer to paragraph S6.1.6.2). The standard address in bits 5 through 0 is 000010, which may be changed by changing jumper connections on the circuit board.

S6.1.6.6 CONTROLLER STATUS BYTE

In response to a function command device order followed by Data In, the controller places a controller status byte (figure S6.1-5) on the input data bus to the CPU. The definitions of the status bits are as follows:

a. Test Alternate Status (bit 7) - This bit is a 1 when any of the following conditions occur:

   1. Data Set Ready from the modem goes low (not ready).

2. The loss of carrier mask is set, and the carrier on signal from the modem drops, indicating a loss of carrier.

3. The data terminal is not ready and a ringing indication is received from the modem.

4. The supervisory received data mask is set, and a space is detected in the supervisory received data from the modem.

5. The clear to send mask is set, and one of the following two conditions (selected by jumpers) occurs:

   (a) A clear to send signal is received from the modem (standard).

   (b) A mark is present in the supervisory received data from the modem (optional).

b. Bit 6 - This bit is always 0 (not used).

c. Parity Error (bit 5) - This bit is a 1 when the controller detects a parity error in the character received from the communications device.

d. Framing Error (bit 4) - This bit is a 1 when a stop bit is not present following the last data bit of a received character.

e. Overrun Error (bit 3) - This bit is a 1 when the previously received character has not been transmitted to the CPU before the current character is loaded into the receiver holding register.

f. Ready to Receive Transmit Word (bit 2) - This bit is a 1 when the controller is ready to receive the next transmit character from the CPU.

g. Input Word Ready (bit 1) - This bit is a 1 when the controller contains a received character that is ready for transmittal to the CPU.

h. Block Transfer Mode (bit 0) - This bit is a 1 when the controller is in the concurrent input or concurrent output mode.

S6.1.6.7 RECEIVED DATA BYTE

In either the program-controlled mode or the concurrent mode, data bytes (figure S6.1-5) received from the communications device are supplied over the input data bus to the CPU.

S6.1.7 INTERRUPTS

An interrupt is armed when any of the following device orders are issued by the CPU:

a. Arm Interrupt

b. Concurrent Input

c. Concurrent Output

Once armed, an interrupt request is issued to the CPU when any of the following conditions occur:

a. The test alternate status bit is a 1 (refer to paragraph S6.1.6.6).

b. In the concurrent input or output mode, when a disconnect device order is issued.

The CPU responds to the interrupt request by issuing Interrupt Acknowledge. If no higher-priority device is requesting service, the controller responds by placing the address byte (refer to paragraph S6.1.6.5) on the input data bus.

S6.1.8 JUMPER OPTIONS

S6.1.8.1 BAUD RATE

Baud rates of 110, 134.5, 150, 300, 1200, 1800, 2400 and 9600 are programmable by the CPU. The following fixed baud rates are jumper selectable.

a. 75 Baud - Cut etch from E32 to E33, and add jumper from E34 to E33.

b. 600 Baud - Cut etch from E26 to E27, and add jumper from E25 to E27.

c. 4800 Baud - Cut etch from E19 to E20, and add jumper from E18 to E20.

S6.1.8.2 CHARACTER FORMAT OPTIONS

The etched-circuit jumpers provide for a programmable character format, as described in paragraph S6.1.6.3. If a fixed format is desired, the etched-circuit jumpers must be cut, and the jumpers given in table S6.1-2 must be added.

Table S6.1-2. Character Size

Number of Bits	Eyelet 68 to Eyelet			Eyelet 65 to Eyelet		
	67	69	N/C*	64	63	N/C*
Programmable	X			X		
8 bits			X			X
7 bits		X				X
6 bits			X		X	
5 bits		X			X	

* No Connection

S6.1.8.3 ADDRESS OPTIONS

The etched circuit jumpers provide a standard device address of 00001. The address strapping in table S6.1-3 provides jumper connection information for all possible addresses. If an address other than the standard address is desired, the address strapping table provides the necessary jumper information.

Table S6.1-3. Address Strapping

Hex Address	Jumper Eyelet Numbers									
	48 To		45 To		39 To		36 To		41 To	
	47	49	46	44	38	40	37	35	43	42
	13 To		10 To		4 To		1 To		7 To	
	15	14	12	11	6	5	3	2	9	8
0*		X		X		X		X		X
1		X		X		X		X	X	
2		X		X		X	X			X
3		X		X		X	X		X	
4		X		X	X			X		X
5		X		X	X			X	X	
6		X		X	X		X			X
7		X		X	X		X		X	
8		X	X			X		X		X
9		X	X			X		X	X	
A		X	X			X	X			X
B		X	X			X	X		X	
C		X	X		X			X		X
D		X	X		X			X	X	
E		X	X		X		X			X
F		X	X		X		X		X	
10	X			X		X		X		X
11	X			X		X		X	X	
12	X			X		X	X			X
13	X			X		X	X		X	
14	X			X	X			X		X
15	X			X	X			X	X	
16	X			X	X		X			X
17	X			X	X		X		X	
18	X		X			X		X		X
19	X		X			X		X	X	
1A	X		X			X	X			X
1B	X		X			X	X		X	
1C	X		X		X			X		X
1D	X		X		X			X	X	
1E	X		X		X		X			X
1F	X		X		X		X		X	

Set 1

Set 2

*NOTE: Address "0" is normally assumed and is jumpered by etch. If another address is used the etch between the eyelets that are changed must be cut.

S6.2.1 BLOCK DIAGRAM DESCRIPTION

Figure S6.2-1 is a detailed block diagram of the controller, which contains a separate block symbol for each sheet of logic diagrams 1010 through 1021 in LD 1010. Logic diagrams 1010 through 1021 are the detailed logic diagrams, which are referenced in the text that follows. The paragraphs that follow describe each element shown in figure S6.2-1.

S6.2.2 DATA RECEIVERS AND DEVICE ORDER DECODER (figure 1010)

Output data bits OD00/ through OD07/ from the CPU are supplied through data receivers, and are distributed throughout the controller. Bits 0 through 4 are supplied through the address jumpers to gate 6F-8. When all bits are in the states that correspond to the device address of the controller, the device order decoder is enabled. The device order code in bits 5 through 7 is then decoded, producing the device order decodes described in paragraph S6.1.6.2.

S6.2.3 MASTER RESET AND CONTROL DECODER (figure 1011)

I/O mode control lines I01X/ through I03X/ from the CPU are applied through data receivers to the control decoder. When one or more of the three lines goes low, the control decoder is enabled at the next positive-going transition of the clock phase 2 (CPH2/) line. The I/O mode control code is then decoded, producing the I/O mode control decodes described in paragraph S6.1.6.1.

When the Data In (DIXX), Data Out (DOXX), or Control Out (COXX) of the I/O mode control codes are decoded, or when ANXX/ (answer) is low occurs, clock KIXX = 1 is produced. When the code for either concurrent acknowledge (CACK) or interrupt acknowledge (IACK) is decoded, ACK (acknowledge) is high. When the master reset (MRST/) line from the CPU goes low, MRST (high), MRST/ (low) and MRSTA/ (low) are all produced, producing a general reset action in the controller.

S6.2.4 FUNCTION STORAGE REGISTER (figure 1012)

Each time DIXX/, DOXX/, COXX/, or ANXX/ go low, the function storage register is strobed on the trailing edge of the KIXX clock. Each state of the function storage register is described in the following paragraphs. The storage register is cleared when MRSTA/ goes low.

S6.2.4.1 XFER FLIP-FLOP

The occurrence of CCAN/ (concurrent answer) or DATX/ (data transfer device order) going low sets the XFER flip-flop. If DOXX/ (data out) goes low, DAOT/ (output transfer) goes low. If DIXX/ goes low, DAIN (input transfer) goes high. The flip-flop is reset at the KIXX clock that follows the occurrence of DATX/ = 1.

S6.2.4.2 FUNC FLIP-FLOP

When a function device order is decoded (FUNC is high), the FUNC flip-flop is set. If DOXX goes high, FNCD/ (function command coming) is low. If DIXX/ is low, STAT (output controller status) goes high. The flip-flop is reset at the KIXX clock that follows the occurrence of FUNC/ going high.

S6.2.4.3 CCIN FLIP-FLOP

When a concurrent input device order is decoded (CCIX high), the CCIN flip-flop is set, producing CCIN (concurrent input) high and BLKM/ (block mode) low. The flip-flop is reset at the KIXX clock that follows the occurrence of DISC (disconnect device order) going high.

S6.2.4.4 CCOT FLIP-FLOP

When a concurrent output device order is decoded (CCOX high), the CCOT flip-flop is set, producing CCOT (concurrent output) high and BLKM/ (block mode) low. The flip-flop is reset at the KIXX clock that follows the occurrence of DISC (disconnect device order) going low, and is also reset when RINT (test alternate status interrupt) is high.

S6.2.4.5 AFUN FLIP-FLOP

When an alternate function command device order is decoded (DC07 high), the AFUN flip-flop is set. If DOXX is high, ACLD/ (communications control byte coming) is low. If DIXX/ goes low, ASTA (output modem status) goes high. The flip-flop is reset at the next KIXX clock that follows the occurrence of DC07/ going high.

S6.2.5 COMMUNICATIONS CONTROL BYTE STORAGE REGISTER (figure 1013)

The communications control byte storage register is cleared when MRST/ goes low. When ACLD/ (communications control byte coming) goes low, the register is loaded

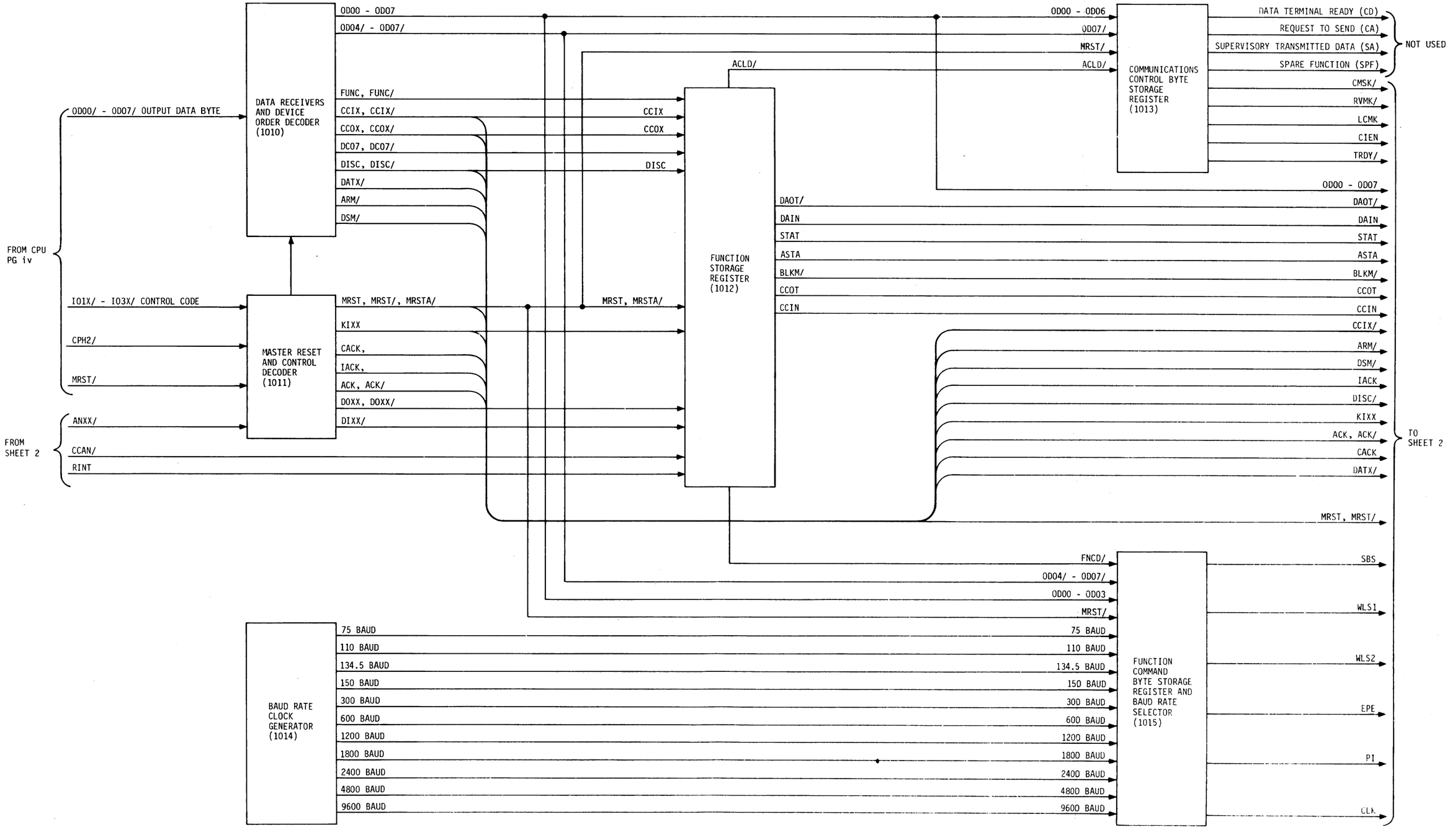


Figure S6.2-1. Detailed Block Diagram (sheet 1 of 2)

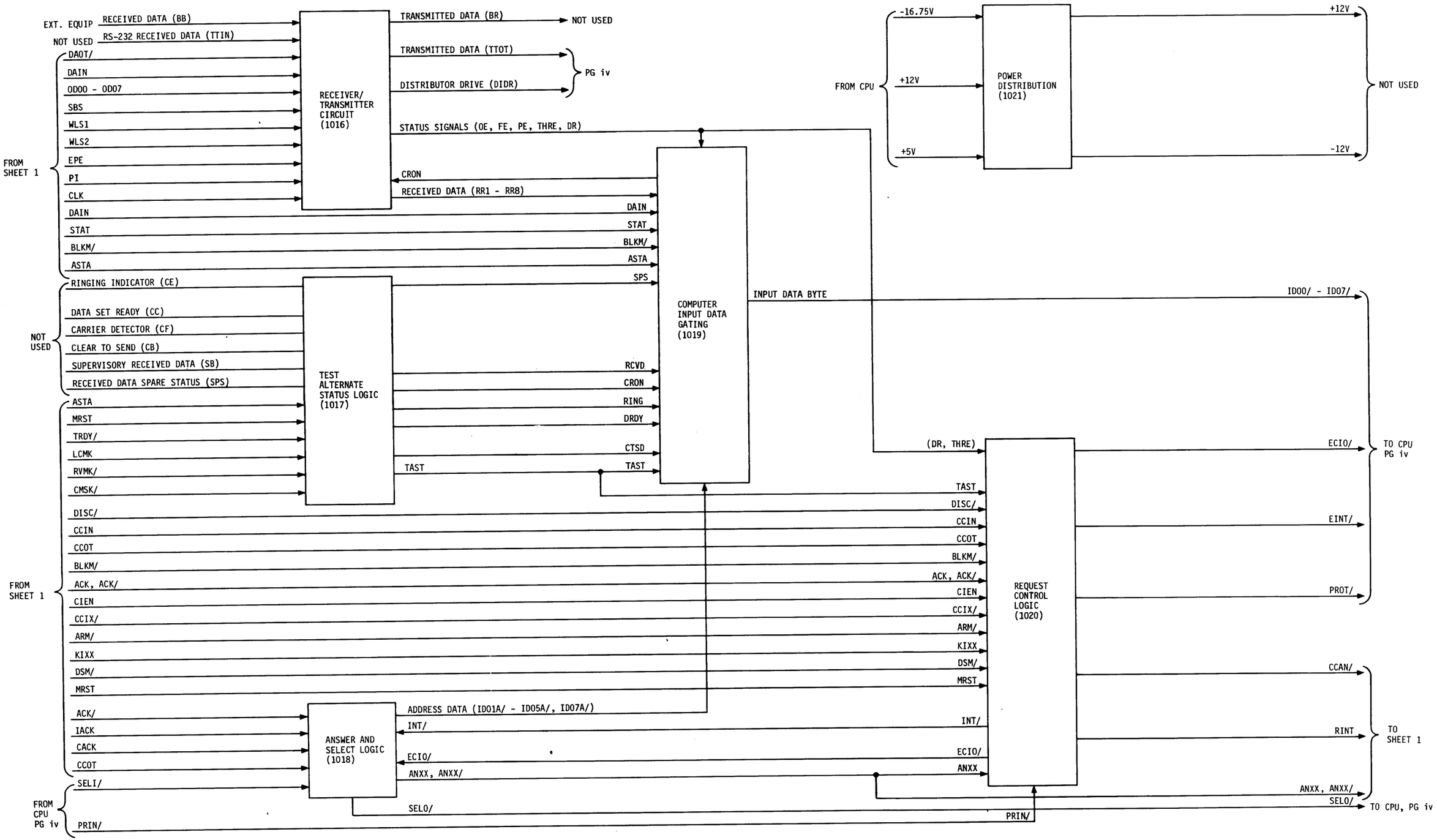


Figure S6.2-1. Detailed Block Diagram (sheet 2 of 2)

with the communications control byte. Stages 0 through 3 and 7 are loaded with the complemented bits, and stages 4 through 6 are loaded with the uncomplemented bits. The outputs of stages 0 through 3 are not currently being used.

S6.2.6 BAUD RATE CLOCK GENERATOR (figure 1014)

The baud rate clock generator consists of a 3.686-MHz crystal-controlled oscillator (Y1) and a series of counter stages that count down the 3.686-MHz clock to produce clock derivatives whose frequencies are 16 times the following baud rates:

a. 9600 baud
b. 4800 baud
c. 2400 baud
d. 1800 baud
e. 1200 baud
f. 600 baud
g. 300 baud
h. 150 baud
i. 134.5 baud
j. 110 baud
k. 75 baud

Each time a full count is reached in one of the counter sets, the associated counter stages are pre-loaded with the count value established by the voltage (0 volt or +5 volts) applied to the preset inputs.

Counter stage 9C serves as a divide-by-5 counter for the 3.686-MHz clock, and produces a 737.2-kHz square wave output for the following three counter sets:

a. Counter stages 5A and 4A produce the 9600, 4800, 2400, 1200, 600, 300, and 150 baud clocks. The 150-baud clock is applied to a divide-by-2 flip-flop, producing the 75-baud clock.

b. Counter stages 8B and 8A, in conjunction with a divide-by-2 flip-flop, produce the 110-baud clock.

c. Counter stages 9B and 9A, in conjunction with a divide-by-2 flip-flop, produce the 134.5-baud clock.

Counter stages 8C and 7C count down the 3.686-MHz clock to produce the 1800-baud clock.

S6.2.7 FUNCTION COMMAND BYTE STORAGE REGISTER AND BAUD RATE SELECTOR (figure 1015)

The function command byte storage register is cleared when MRST/goes low. When FNCD/ (function command byte coming) goes low, the register is loaded with the function command byte described in Paragraph S6.1.6.3.

Stages 0 through 3 are loaded with the uncomplemented bits, and stages 4 through 7 are loaded with the complemented bits. The code in stages 0 through 2 is applied to the baud rate selector, which is a multiplexer that selects the baud rate clock designated by the 3-bit code. The selected clock output (CLK) of the baud rate selector, and the outputs of function byte storage register stages 3 through 7 (WLS1, WLS2, SB5, P1, and EPE) are applied to the modem input/output data interface on figure 1016.

S6.2.8 RECEIVER/TRANSMITTER CIRCUIT (figure 1016)

The receiver/transmitter is a large-scale integration (LSI) integrated circuit that performs the parallel-to-serial conversion of data to be transmitted to the communications device, and the serial-to-parallel conversion of data received from the communications device. Table S6.2-1 describes each input to and output from the receiver/transmitter.

S6.2.8.1 PROCESSING OF SERIAL DATA INPUTS

Serial data inputs may be received over the received data line (BB) from the modem, or over the received data line (TTIN) from the teletype. If J3-28 is jumpered to J3-27, CRON, high, (carrier on) must exist before the received data from the modem is accepted. The received data from the connected source is applied to the RI input of element 2E. The number of valid data bits in the resulting parallel data output produced on lines RR1 through RR8 is determined by the states of the WLS1 and WLS2 lines from the function command byte storage register, as described in Paragraph S6.1.6.3. The framing error detection criterion (number of stop bits) is established by the state of the SBS line from the function byte storage register, as described in Paragraph S6.1.6.3. The parity error detection mode (if any) is established by the P1 and EPE lines from the function byte storage register, as described in Paragraph S6.1.6.3. A high-level input on the RI line corresponds to a mark, and a low-level input corresponds to a space. When no character is being received, a start bit detect circuit continually searches for the mark-to-space

transition that defines the start bit. When this transition is detected, a counter is reset, and is allowed to count (under the control of the CLK signal applied to the RRC line) at a rate that is 16 times the selected baud rate. If a space is still present when the counter detects the center of the start bit interval, a valid start bit is assumed, and the counter continues to count to locate the center of each subsequent bit in the received data. Each detected bit is entered into a receiver register. After the number of bits established by the states of lines WLS1 and WLS2 has been received, the next bit is checked to assure that it is a mark (stop bit). If a stop bit is not detected, a framing error (FE high) indication is produced.

If a stop bit is detected in the proper position, the assembled character (less the start and stop bits) is transferred to a receiver holding register, and DR = 1 (data received) is produced, indicating that a data byte is ready for transmittal to the CPU. The received data is made available on lines RR1 through RR8. Subsequently, when DAIN = 1 (input transfer) is produced, the DR line is reset. If the next character is transferred to the receiver holding register before the DR line is reset, the overrun error (OE high) indication is produced, indicating that an overrun (lost character) error has occured.

S6.2.8.2 PROCESSING OF DATA TO BE TRANSMITTED

When the transmitter holding register is empty, THRE goes high, indicating that the unit is ready to receive a data character from the CPU. Subsequently, when the CPU sends the data character, the occurrence of DAOT/, low, (output transfer) activates the THRL (transmitter holding register load) line, and the output data byte on lines TR1 through TR8 is strobed into the transmitter holding register. Also, the THRE line is reset. Then, when transmission of the previous character is completed, the data from the transmitter holding register is transferred to the transmitter register, and the THRE line goes high again. The desired number of data bits, stop bits, and the state of the parity bits (if any) is established by the WLS1, WLS2, SB3, P1, and EPE lines, as described in Paragraph S6.1.6.3. The bits of the character in the transmitter register are shifted out serially over the TRO line. A high output on the TRO line corresponds to a mark, and a low output corresponds to a space. The TRO output is applied through drivers to the connected communications device.

The rate at which the bits are clocked out is 1/16 the frequency of the CLK input applied to the TRC line. After all bits of the character have been shifted out, the next character (if any) from the transmitter holding register is transferred to the transmitter register, and the process is repeated.

Table S6.2-1. Receiver/Transmitter Functions

Signal	Mnemonic	Function
Transmitter Holding Register Empty	THRE	High-level output voltages on this line indicate the Transmitter Holding Register has transferred its contents to the Transmitter Register and can be loaded with a new character
Word Length Select	WLS2-WLS1	These two lines select the character length (5, 6, 7 or 8 bits).
Parity Inhibit	PI	High-level input voltages on this line inhibit the parity generation and verification circuits. The stop bit(s) immediately follow the last data bit on transmission if parity is inhibited. Low-level input voltages enable the parity generation and verification circuits. PI will, when a high-level input voltage is applied, also clamp the PE line (pin 13) to a low-level output voltage.
Even Parity	EPE	This line selects either even or odd parity to be generated by the transmitter and checked by the receiver. High-level input voltages select even parity and low-level input voltages select odd parity.
Stop Bit(s) Select	SBS	This line selects the number of stop bits generated after the parity bit during transmission. High-level input voltages on this line select two stop bits, and low-level input voltages select a single stop bit.
Control Register Load	CRL	High-level input voltages on this line load the Control Register with the control bits (WLS1, WLS2, EPE, PI, SBS). This line is wired to a high-level input voltage.
Status Flags Disconnect	SFD	High-level input voltages applied to this pin disconnect the PE, FE, OE, DR and THRE circuit outputs.
Receiver Input	RI	Serial input data received on this line enters the Receiver Register at a point determined by the character length, parity, and the number of stop bits. High-level input voltages must be present when data is not being received.

Table S6.2-1. Receiver/Transmitter Functions (continued)

Signal	Mnemonic	Function
Data Received	DR	High-level output voltages indicate an entire character has been received and transferred to the Receiver Holding Register.
Data Received Reset	DRR	Low-level output voltages applied to this line reset the DR line.
Receiver Holding Register Data	RR1-RR8	The contents of the Receiver Holding Register appear on these lines in parallel when a low-level input voltage is applied to RRD. Selection of a word length less than 8 bits causes the MSB of the character to be forced to a low-level output voltage. The character is right justified. RR1 (pin 12) is the LSB of the character.
Receiver Register Clock	RRC (CLK)	This clock is 16 times faster than the desired receiver shaft rate.
Receiver Register Disconnect	RRD (SFD)	High-level input voltages applied to this line disconnect Receiver Holding Register outputs from the RR8 thru RR1 data outputs (pins 5-12).
Parity Error	PE	The status of the parity verification circuit appears on this line when a low-level input voltage is applied to the Status Flags Disconnect (pin 16) control line. Wired-OR capability is provided on this line allowing PE lines from other R/T's to be OR-tied. High-level output voltages on this line (under the conditions above) indicate a Parity error in the recieved parity bit as programmed by Even Parity Enable control line (pin 39). The status is updated each time a character is transferred from Receiver Register to Receiver Holding Register.
Framing Error	FE	Status of the stop bit detection circuit appears on this line when a low-level input voltage is applied to Status Flag Disconnect (pin 16) control line. Wired-OR capability is provided on this line allowing FE lines from other R/T's to be OR-tied. High-level output voltage indicate received character has no valid stop bit (i.e., the bit following the parity bit is not a high-level input voltage). The status is updated each time a character is transferred from Receiver Register to Receiver Holding Reg.
Overrun Error	OE	Status of Data Received circuit appears on this line when a low-level input voltage is applied to Status Flag Disconnect (pin 16) control line. Wired-OR Capability is provided on this line allowing OE

Table S6.2-1. Receiver/Transmitter Functions (continued)

Signal	Mnemonic	Function
		lines from other R/T's to be OR-tied. High-level output voltages indicate previously received character was not read (DR line not reset) before present character was transferred to the Receiver Holding Register.

S6.2.9 TEST ALTERNATE STATUS LOGIC (figure 1017)

The test alternate status logic is not currently being used.

S6.2.10 ANSWER AND SELECT OUT LOGIC (fugure 1018)

If the select in line (SELI/) is low, indicating that no higher-priority device is requesting service, the occurrence of either of the following conditions will produce an answer action:

a. IACK is high (interrupt acknowledge), and the controller has an interrupt pending ($\overline{IS01}$ + $\overline{IS02}$ is low). The occurrence of IACK high produces ACK/ low.

b. CACK is high (concurrent acknowledge), and the controller has a concurrent I/O request pending (ECIO/ is low). The occurrence of CACK high produces ACK/ low.

When either condition occurs, the select out line (SELO/) is made high, indicating that select in has been captured, and ANXX (answer) goes high. The occurrence of ANXX, high, causes the address byte to be produced (refer to Paragraph S6.1.6.5).

S6.2.11 COMPUTER INPUT DATA GATING (figure 1019)

The computer input data gating serves as a multiplexer that selects the type of data placed on the input data bus (ID00/ through ID07/) to the CPU. The data on the input data bus may be any of the following:

a. When DAIN (input transfer) is high, the states of RR1 through RR8 from the receiver/transmitter are supplied via the data gates to the input data bus.

b. When STAT is high, the controller status byte (refer to Paragraph S6.1.6.6) is supplied via the controller status gates to the input data bus.

c. When ASTA is high, the modem status byte is supplied via the modem status gates to the input data bus.

d. When ANXX is high, the address byte (from the answer and select out logic) is supplied to the input data bus.

S6.2.12 REQUEST CONTROL LOGIC (figure 1020)

The request control logic controls the generation of interrupt request and concurrent I/O requests. Before either request can be generated, the priority in line (PRIN/) from the CPU must be low. When either request is produced, the priority out line (PROT/) goes low for the duration of the request, indicating that priority in has been captured.

S6.2.12.1 INTERRUPT REQUESTS

The occurrence of CCIX/, CCOX/, or ARM/ being low arms the interrupt by setting the IS01 flip-flop on the trailing edge of the next KIXX clock. At the same time, the IS02 flip-flop is reset. Once IS01 has been set, an interrupt request is issued when flip-flop IS02 is set by any of the following events:

a. DISC/ (disconnect device order) goes low, which sets IS02 on the trailing edge of the next KIXX clock.

b. CIEN (input character interrupt enable) goes high in the program-controlled mode (BLKM/ high), and a character is ready for transmittal to the CPU (DR = 1 exists). Under these conditions, LINT (high) produces STIN/ (low) which direct sets the IS02 flip-flop.

c. The TAST flip-flop is set by the occurrence of TAST (test alternate status) being high. This condition produces RINT (high) (test alternate status interrupt), which in turn produces STIN/ (low). The occurrence of STIN/ direct sets the IS02 flip-flop. The TAST flip-flop then remains set until either ANXX or MRST goes high.

When IS01 and IS02 are both set, EINT/ (interrupt request) goes slow, and continues until the IS01 flip-flop is reset on the high-to-low transition of the ANXX signal. When DSM/ (disarm device order) goes low, the IS01 and IS02 flip-flops are both reset.

S6.2.12.2 CONCURRENT I/O REQUEST

A concurrent I/O request (ECIO/ low) is produced when either of the following conditions exists:

a. The controller is in the concurrent input mode (CCIN high), and DR high occurs (data byte ready for transmittal to CPU).

b. The controller is in the concurrent output mode (CCOT high), and THRE goes high, indicating that the controller is ready to accept the next data byte from the CPU.

Following the concurrent I/O request, ANXX goes high (answer), producing CCAN/ , (concurrent answer) low. The concurrent I/O request terminates when either DR (concurrent input mode) or THRE (concurrent output mode) goes low.

S6.2.13 POWER DISTRIBUTION (figure 1021)

The controller receives -16.75V, +12V, and +5V power from the CPU, and supplies +12V and -12V power to the modem. Also, a derivative (DIDR) of the -16.75V voltage is developed through resistor R9, and supplied to the Teletype as the DIDR signal (distributor drive for the 20-ma current loop).

S6.3.1 PARTS LIST

All electronic parts of the controller are illustrated on figure S6.1-1 and are identified in the following list.

Parts List for Local/Remote Single Channel Communications Controller (figure S6.1-1)

Reference Designation	Part Number	Description	Qty	Vendor
	D20002610	Printing Wiring Board	1	Microdata
C1, 2, 7, 11, 18, 21, 23, 27	150D475X0010A2	Capacitor, 4.7uf, 10v	8	Sprague
C3, 4, 5, 6, 8, 9, 10, 12, 13, 14, 15, 16, 17, 19, 22, 24, 26	TG-S10	Capacitor, .01uf, 100v	17	Sprague
C20	150D105X0035A2	Capacitor, 1.0uf, 35v	1	Sprague
C25, 28-32	150D155X0035A2	Capacitor, 1.5uf, 35v	6	Sprague
CR1	1N751A	Diode	1	
CR2	1N759	Diode	1	
CR3	FDH600	Diode	1	Fairchild
J3	D2000307-35	Header, 50 Pin	1	Microdata
1B	SN74L04N	Integrated Circuit	1	T.I.
1C, 1F, 1H, 1K, 2H, 3C, 5F, 7B, 7E, 8K, 8L	SN7404N	Integrated Circuit	11	T.I.
1D, 1J	SN7442N	Integrated Circuit	2	T.I.
1E, 1L, 2K, 2L, 3K, 4F, 7K, 8J	SN7438N	Integrated Circuit	8	T.I.

Parts List for Local/Remote Single Channel Communications Controller (figure S6.1-1) (continued)

Reference Designation	Part Number	Description	Qty	Vendor
2A, 2J, 3J, 4C, 4J, 7A	SN74107N	Integrated Circuit	6	T.I.
2B, 4K	SN7410N	Integrated Circuit	2	T.I.
2C, 2D, 4B, 4H, 5E	SN7400N	Integrated Circuit	5	T.I.
2E	CS20001120 or TR1 402A 7230	Integrated Circuit	1	Microdata West. Dig.
3B, 3H, 4D, 4L	SN7402N	Integrated Cirucit	4	T.I.
3D	SN7420N	Integrated Circuit	1	T.I.
4A, 5A, 7C, 8A, 8B, 8C, 9A, 9B, 9C	SN7416N	Integrated Circuit	9	T.I.
5H	SN7476N	Integrated Circuit	1	T.I.
5L, 6C, 6E, 7L	U7B931459X	Integrated Circuit	4	Fairchild
6B	U7B931259X	Integrated Circuit	1	Fairchild
6F	SN7430N	Integrated Circuit	1	T.I.
7D, 9L	MC1488L	Integrated Circuit	2	Motorola
9J, 9K	MC1489L	Integrated Circuit	2	Motorola
R1, 6, 7, 8, 11	RC07GF102J	Resistor, 1K, 1/4w, ±5%	5	
R2, 5	RC07GF301J	Resistor, 300Ω, 1/4w, ±5%	2	
R3, 4, 10	RC07GF221J	Resistor, 220Ω, 1/4w, ±5%	3	
R9	RC07GF151F	Resistor, 150Ω, 1/4w, ±5%	1	
R12	RC07GF202J	Resistor, 2K, 1/4w, ±5%	1	
R13	RC07GF103J	Resistor, 10K, 1/4w, ±5%	1	
R14	RC20GF132J	Resistor, 1.3K, 1/2w, ±5%	1	
P3	A20001114-35	Connector	1	Microdata
Q1	2N2906	Transistor T018	1	
(Q1 Ref)	100-000	Transistor Pad	1	Delbert Blinn
Y1	S0-3537 3.686 MHZ	Oscillator 3.686 MHZ	1	Monitor Prod

Parts List for Local/Remote Single Channel Communications Controller (figure S6.1-1) (continued)

Reference Designation	Part Number	Description	Qty	Vendor
	D20001065	Cable Retainer	1 Ref	Microdata
		Screw 4-40 NC X 5/16 Lg, Pan Hd, Cross Rec, Stl, Cad Pl	2 Ref	
		Washer Split Lock #4 Stl Cad Pl	2 Ref	
	90-0-6503-20	Handle, Extractor	2	Southco
	90-0-5858-24	Rivet	2	Southco
	47743	Contact,Female	46	Berg Elect

Supplement 7

SWITCH SELECTABLE BAUD RATE
FOUR-CHANNEL AND EIGHT-CHANNEL CONTROLLERS

TABLE OF CONTENTS

Page

List of Illustrations

List of Tables

Supplement 7

SWITCH SELECTABLE BAUD RATE FOUR-CHANNEL AND EIGHT-CHANNEL CONTROLLERS

S7.1 INTRODUCTION

This supplement describes the two switch selectable baud rate (SSBR) controllers. The logic diagrams for the controllers are in LD 1010. The eight-channel SSBR is exactly the same as the four-channel SSBR except that it has four more input/output (receiver/transmitter) channels. The logic diagrams (figures 1210 through 1224) for the four-channel controller are exactly the same as the logic diagrams for the eight-channel controller (figures 1110 through 1131) except there are baud rate selection switches (figure 1118), receiver/transmitter circuits (figures 1124 through 1127), and output line drivers (figure 1129) for channels 5 through 8 for the eight-channel controller.

S7.1.1 PURPOSE AND FUNCTION

The SSBR controller enables the CPU to communicate with up to four (eight) asynchronous local or remote devices. The controller provides four (eight) channels for servicing terminals having a 20-milliamp current-loop interface or a standard EIA RS-232-C interface. Each channel is capable of simultaneous transmit and receive (full duplex) operation. Each transmitter or receiver operates asynchronously with respect to all other transmitters and receivers. Baud rate, format, and error detection for serial characters exchanged between the controller and each external device are determined by strapping and/or switch selections within the controller. The controller utilizes the Byte I/O interface to exchange eight-bit parallel characters, instructions, and error flags with the CPU. Each transmitter and receiver is double-buffered, allowing a full character time for data transfer between the CPU and controller.

S7.1.2 PHYSICAL DESCRIPTION

The SSBR controllers are plug-in, printed circuit board assemblies. The placement of either controller may be in any of the four locations from J13 through J16. See

page iv of LD 1010 for controller location priority sequence in the CPU. Figure S7.1-1 shows the areas of the SSBR four-channel controller blocked out to indicate the locations of circuit elements used for the functions performed by the assembly. Figure S7.1-2 shows the same information for the SSBR eight-channel controller. These figures indicate functions which correspond to the figure titles in LD 1010:

Address and Function Decoders
Function Control
Channel Select and Mode Decoders
Interrupt Control (includes some timing circuits)
Timing
Receiver/Transmitter Sequencer
Baud Rate Clock Generator
Baud Rate Selection
I/O Line Driver and Line Receivers
Receiver/Transmitter Circuits
Input Data Multiplexer

The baud rate for each channel is switch selectable. The available baud rates are: 75, 110, 134.5, 150, 300, 600, 1200, 2400, 4800, and 6400.

S7.1.3 OPERATION SUMMARY

Figure S7.1-3 is a block diagram of the SSBR controller. As shown in the diagram, the controller interfaces with the CPU via the Byte I/O bus. Standard two-byte I/O commands control data transfer functions. A two-byte command (using the data-in mode) is also used to disarm interrupts. Since control mode commands are used for acknowledge functions. Table S7.1-1 is a summary of CPU interface functions. Since data transfers in either direction are asynchronous, the controller may be performing both data input and data output functions for all channels simultaneously. This requires that the controller have a traffic director which is a sequencer (part of the timing circuits). The transmitter, then the receiver of each channel, is sequenced in order. Data transfers between the CPU and the controller occur when a transmitter is ready to accept another character (byte) and when a receiver has assembled a character into a byte. The scan rate is typically 28 times faster than the character transmission rate for the highest baud rate (9600) which is sufficient to avoid loss of characters as long as the CPU can respond to controller interrupts promptly. However, in the event that a received character is not transferred to the CPU in time, an error flag is generated to notify the CPU of a character loss.

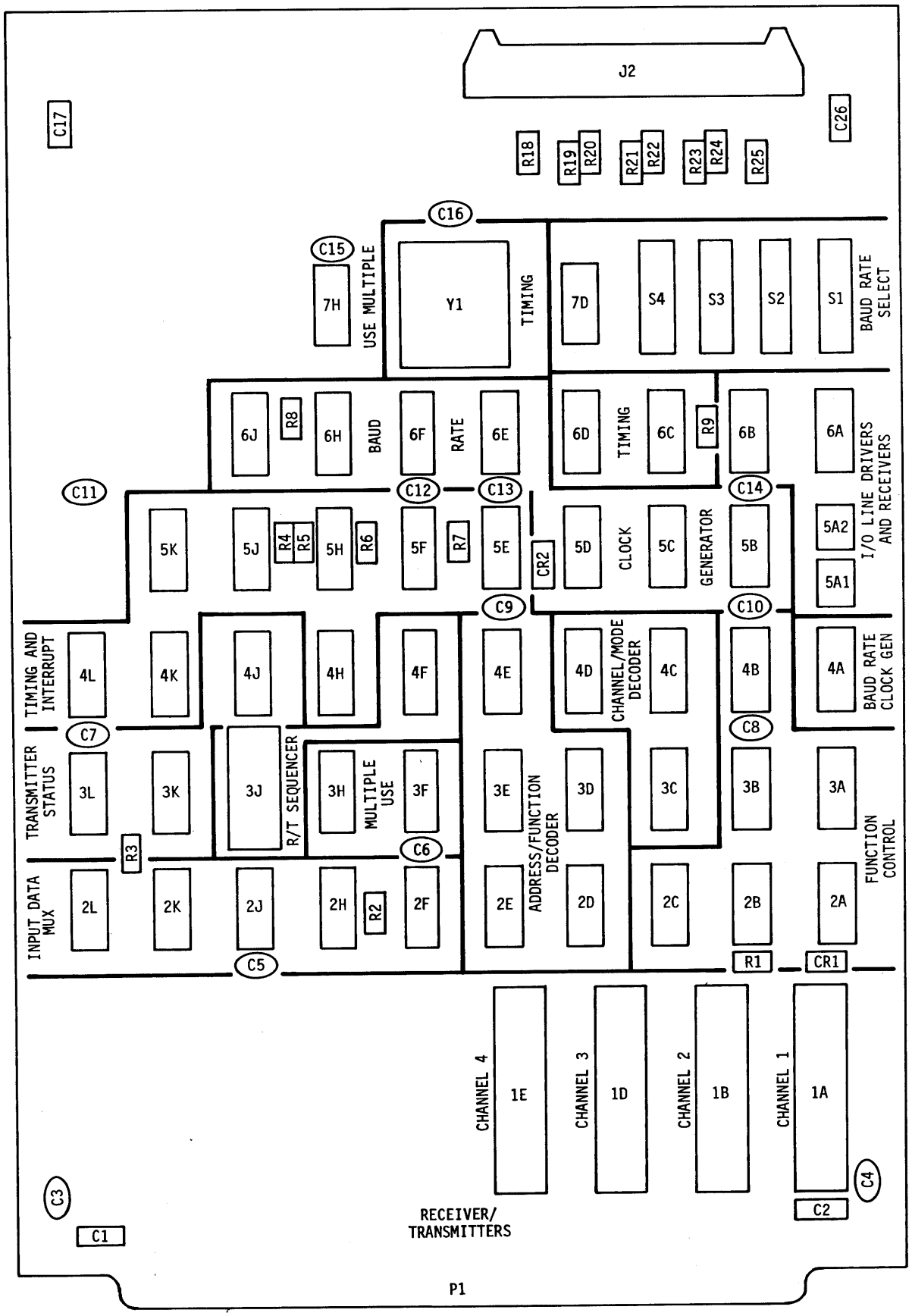


Figure S7.1-1. SSBR Four-Channel Controller Circuit Functions

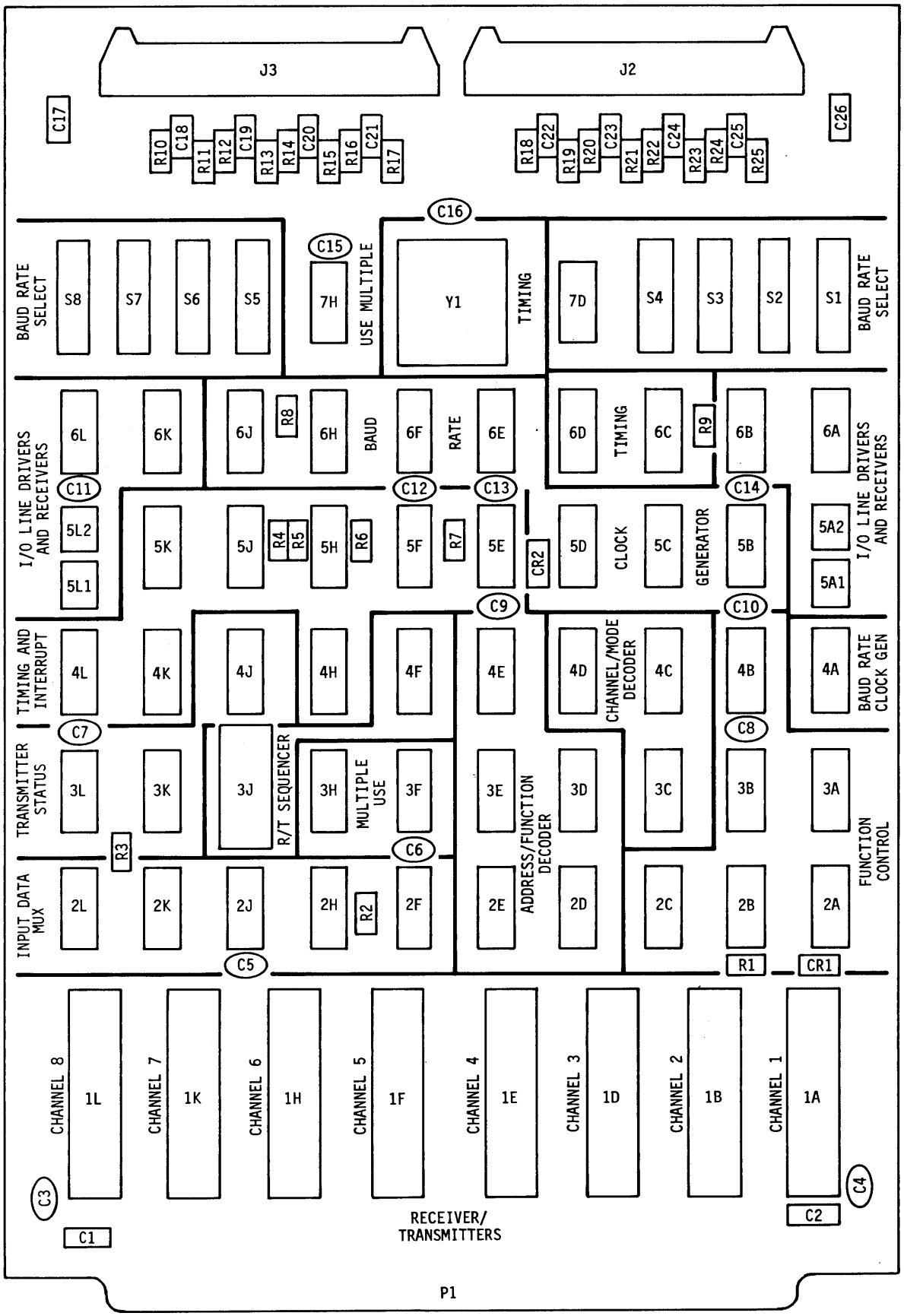


Figure S7.1-2. Eight-Channel Controller Circuit Functions

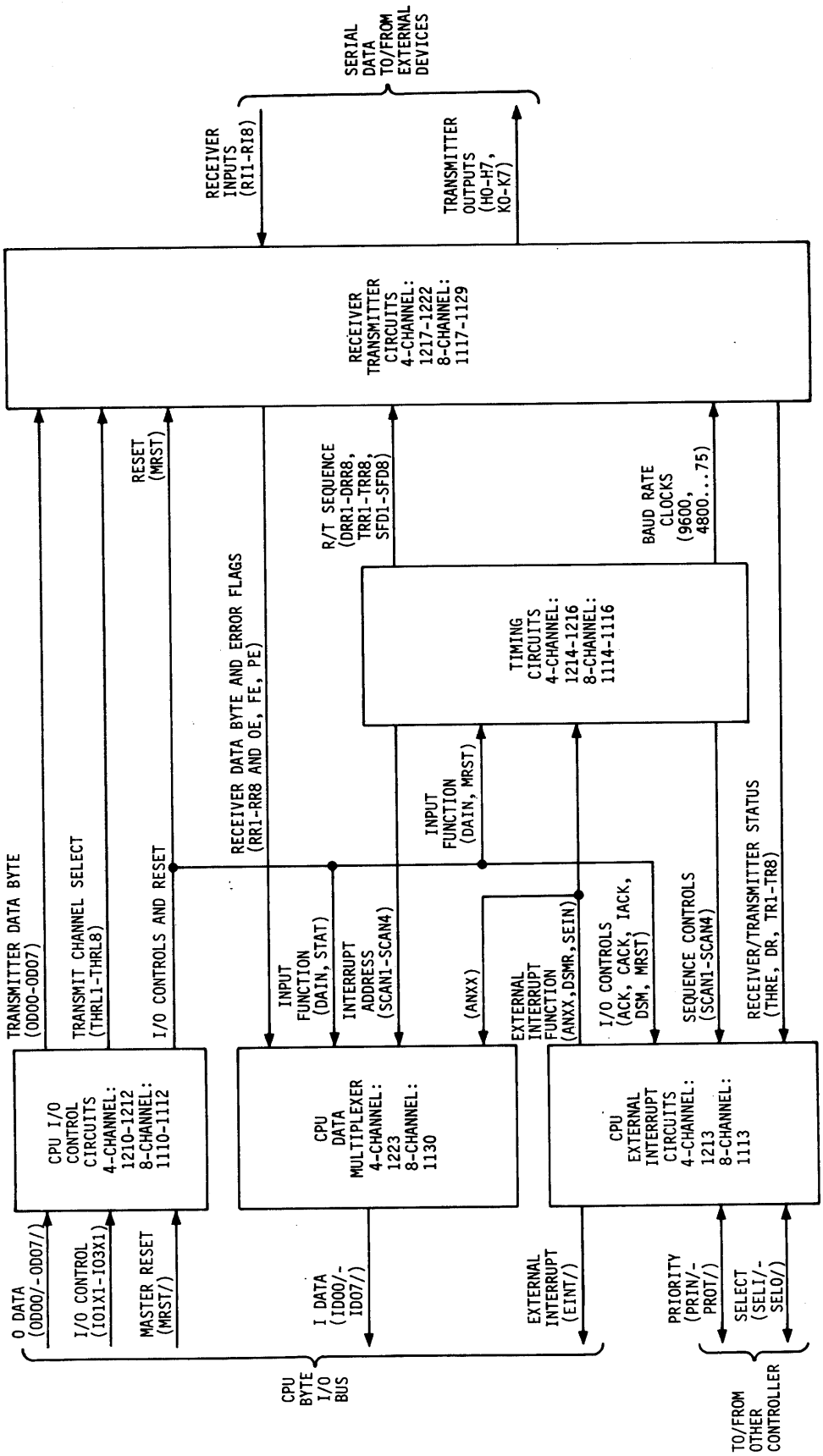


Figure S7.1-3. Eight-Channel Communications Controller Block Diagram

A typical character transmission consists of the controller notifying the CPU that a transmitter holding register is empty. The CPU responds with an interrupt acknowledge command which results in the controller putting the channel number on the data input bus. The CPU responds by addressing the controller with a data out command to that channel. Next the CPU puts the character to be transmitted on the output bus and the controller stores the character in the transmitter holding

Table S7.1-1. CPU-SSBR Controller Interface Functions

Function	Byte	Control Mode	Data Byte
Output Character	First (control)	Function/Device	0 Data = Controller address and Transmit Channel
	Second (output)	Data Out	0 Data = Transmit Character Information Bits
Input Status	First (control)	Function/Device	0 Data = Controller Address and Status Function Code
	Second (input)	Data In	1 Data = Received Character Error Bits
Input Character	First (control)	Function/Device	0 Data = Controller Address and Data Input Function Code
	Second (input)	Data In	1 Data = Received Character Information Bits
Disarm Interrupts	First (control)	Function/Device	0 Data = Controller Address and Disarm Interrupt Function Code
	Second (input)	Data In	None (Controller interrupts are inhibited until a new I/O function is initiated)
Acknowledge Interrupt	--	Interrupt Ackn	1 Data = Interrupt Response Address
Acknowledge Concurrent I/O	--	Concurrent Ackn	None (used only to pass SELECT through the controller)
External Interrupt	Initiated by controller to request output of the next character or input of a received character.		
Priority and Select	Control terms used to establish interrupt priority among CPU interfaces.		

register. As soon as that transmitter empties the transmit register by transmitting the last character, the new character is transferred to the transmit register and the transmitter activates the transmitter holding register empty flag and begins transmitting the character in serial form. The cycle is then ready to repeat.

A typical sequence for a received character consists of the receiver assembling the serial input into the receiver register. When all of the bits for a character have been assembled in the register, the bits are examined for parity (unless parity is inhibited) and for the start and stop bits. When there is a parity error the parity error flag is activated. When the start or stop bits are the incorrect level, the framing error flag is activated. The receiver also activates the data received flag which causes the controller to send an external interrupt to the CPU. The CPU responds by addressing the controller with an interrupt acknowledge command. The controller responds by putting the channel number on the data input bus. The CPU then addresses the controller with a status command for that channel which results in the controller putting the error flags (if any) on the input data bus. If the receiver has started assembling the next character from the input line, the status report will include an overrun error flag. When the status response does not include error flags, the CPU addresses the controller with a data in command for that channel. The controller responds by putting the character byte on the data input bus. The receiver then assembles the next input character in the receiver holding register. Table S7.1-1 is a summary of CPU interface functions used in the transmitter and receiver sequences outlined above.

S7.2 THEORY OF OPERATION

The circuit theory that follows is for either controller. Table S7.2-1 lists the output connectors used (J2 for first four devices; J3 for next four devices) and pin assignments for external devices. Each statement concerning channels (lines) 0 through 3 pertains to both controllers. Statements pertaining to channels (lines) 4 through 7 pertain only to the eight-channel controller.

S7.2.1 OPERATIONAL DESCRIPTION

The function of each of the blocks in figure S7.1-3 is described in paragraphs S7.2.1.1 through S7.2.1.5. The functional processing description in paragraphs S7.2.2 through S7.2.2.5 and the block description describe the overall operation of the SSBR controllers.

Table S7.2-1. External Device Interface Connections, J2 and J3

Type of I/O Operation		Term	Connector Pins				20-MA Jumpers (Notes)
RS-232	20-MA		Line 0 (J2) 4 (J3)	Line 1 (J2) 5 (J3)	Line 2 (J2) 6 (J3)	Line 3 (J2) 7 (J3)	
Output		H	1	13	25	37	
Output Comm.	Output	L	2	14	26	38	
	Output Ret.	K	3	15	27	39	
Input	Term. Input	C	4	16	28	40	Jumper to B
False (+V)	Input	A	5	17	29	41	A and B internally jumpered, connected to +12V via 1.3K.
False (+V)	Term. Input	B	6	18	30	42	
	Filter	F	7	19	31	43	Jumper to E
Input Comm.	Gnd	E	8	20	32	44	Jumper to F
True (-V)	Input Ret.	D	9	21	33	45	(-16.75 via 150)

Composite instruction codes are listed in table S7.2-2 and data byte formats and control codes are defined in tables S7.2-3 and S7.2-4.

S7.2.1.1 CPU I/O Control Circuits. The CPU I/O Control Circuits buffer, decode and store information received on the output lines of the Byte I/O bus. The Master Reset line is activated when power is applied. A buffered Master Reset control term (MRST) is distributed to key controller elements to initialize the controller. The I/O Control lines are decoded to produce individual I/O Controls to key all CPU-related processes. The first 0 Data byte (control byte) is decoded to detect controller (device address and function code. The function code is stored and decoded to produce I/O control terms (DAIN, STAT, DSM, THRL1 through THRL8). The second 0 Data byte (output byte) is buffered and transferred to the designated transmit channel.

S7.2.1.2 CPU Data Multiplexer. The CPU Input Data Multiplexer gates the receiver data byte, error flags, or interrupt response address onto the Byte I/O bus input lines when enabled by the corresponding control term (DAIN, STAT, or ANXX). Each

Table S7.2-2. Composite Instruction Codes

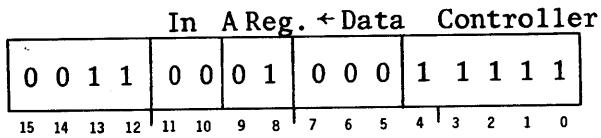
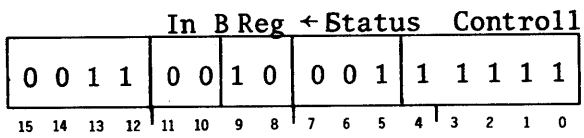
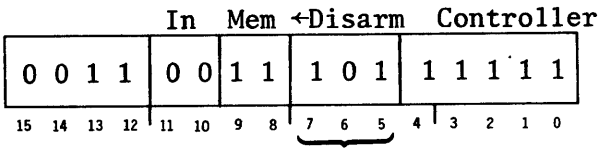
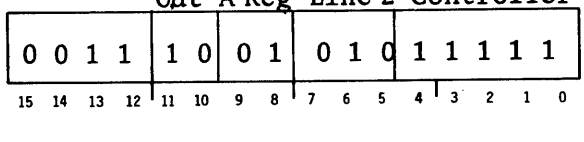
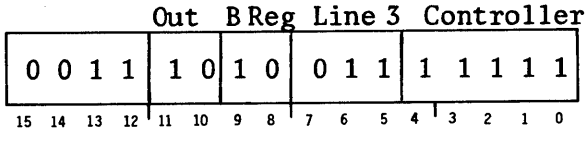
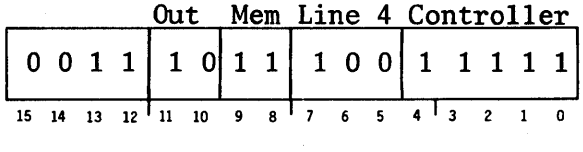
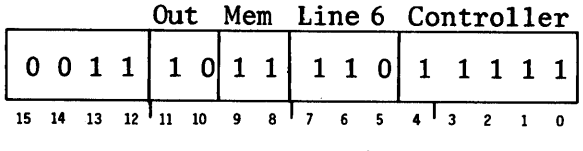
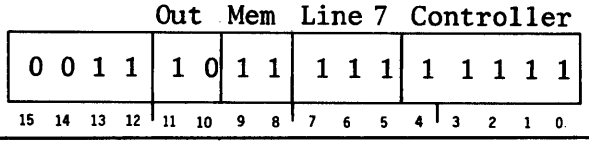
Statements	Codes
Format	
Store data from 8-channel controller in A register	
Store status from 8-channel controller in B register	
Store disarm condition of 8-channel controller in memory	 These function codes would select lines 0, 1 and 5, respectively, for an output operation.
Transmit contents of A register to line 2 of 8-channel controller	
Transmit contents of B register to line 3 of 8-channel controller	
Transmit from memory to line 4 of 8-channel controller	
Transmit from memory to line 6 of 8-channel controller	
Transmit from memory to line 7 of 8-channel controller	

Table S7.2-3. I/O Byte Formats

Function	7	6	5	4	3	2	1	0
Data In or Out	Optional information Bits (Word length 8, 7, or 6)			Information Bits				
Control (Out)	Function Code Output: 000 - In = Line 0-7 Input: 001 = Data Input 010 = Status Input 101 = Disarm Inter			Device Address (all ones)				
Status In	Not Used					Receiver Error Flags Parity Error	Framing Error	Overrun Error
Answer (In)	Interrupt Response Address (table 3-13) Strapped Device Address (Table 3-14) 000 - 111			1=Rec 0=Trans	000 - 111 = Line 0-7			0

Table S7.2-4. I/O Control Codes (Decoded from I01X/-I03X/)

Code 3 2 1	Mnemonic	Name	Process Controlled
0 0 1	COXA/	Function/Device Control	Control Byte
0 1 0	DOXA/	Data Out	Output Data Byte
1 0 0	CAKA/	Concurrent Acknowledge	Select Signal
1 0 1	IAKA/	Interrupt Acknowledge	Interrupt Response
1 1 0	DIXA/	Data In	Input Data Byte

receive line and transmit line has a unique interrupt response address (table S7.2-5) composed of the controller address (strapped, see table S7.2-6) and the designation of the transmitter or receiver initiating the interrupt (SCAN1 through SCAN4).

S7.2.1.3 CPU External Interrupt Circuits. The CPU External Interrupt Circuits produce an interrupt when a receiver indicates that a character is ready for input (DR) or when a transmitter is ready to process a new character (THRE and TR1 through 8). All sixteen lines are checked, in sequence (SCAN1 through SCAN4), for interrupt request status.

S7.2.1.4 Timing Circuits. The Timing Circuits generate basic clock pulses for control timing and ten distinct clock pulse frequencies for baud rate control. A 16-line scanner (SCAN1 - SCAN4) and derivative control terms (DRR1 through DRR8, TRR1 through TRR8, SFD1 through SFD8) multiplex all receive and transmit lines for CPU I/O service.

S7.2.1.5 Receiver/Transmitter Circuits. The Receiver/Transmitter (R/T) circuits perform serial/parallel character conversion. Baud rate and character form for each R/T channel are determined by strapping and switch selects (tables S7.2-7 and S7.2-8). Each received character is checked for framing error (no stop bit received) and parity error (strap selected). An overrun error flag is produced if a new character is received and processed before the CPU has extracted the previous character. To minimize overruns, each processed character is stored separately for extraction while the receiver is processing the next character. Each transmitter is also double-buffered so that the CPU can transfer a new character while the previous character is being transmitted.

S7.2.2 FUNCTIONAL PROCESSING DESCRIPTION

Figures S7.2-1, S7.2-2 and S7.2-3 are flow diagrams describing the sequence of actions performed by the SSBR Communications Controllers. Each diagram illustrates a specific processing function; External Device Input/Output (figure S7.2-1), CPU External Interrupt (figure S7.2-2), CPU Input/Output (figure S7.2-3). Each of these functions is required to complete the overall process of the input or output of a character.

S7.2.2.1 Character Input. To input a character to the CPU from an external device, the following sequence of actions takes place:

1. Serial character is received, assembled and checked for errors. Data Ready (DR) flag is set (figure S7.2-1).

2. When sequenced by the scanner, DR flag causes an interrupt to be produced (figure S7.2-1).

3. CPU acknowledges the interrupt and the interrupt response address is input (figure S7.2-2).

4. CPU addresses a two-byte status input command to the controller. Error flags (if any) are input (figure S7.2-3).

5. CPU addresses a two-byte data input command to the controller. Received character information bits are input (figure S7.2-3).

6. Scanner sequences to next line.

S7.2.2.2 Character Output. To output a character from the CPU to an external device, the following sequence of actions occurs:

1. Transmitter extracts stored character from holding register and begins transmission. Transmitter Holding Register Empty flag (THRE) is set. Transmitter Ready Flag (TR) is set (figure S7.2-1).

2. When sequenced by the scanner, THRE and TR flags cause an interrupt to be produced (figure S7.2-2).

3. CPU acknowledges the interrupt and the interrupt response address is input (figure S7.2-2).

4. Scanner sequences to next line (figure S7.2-1).

5. When data is available, CPU addresses a two-byte data output command to the transmitter. Character information bits are loaded into transmitter holding register for subsequent transmission (figure S7.2-3).

NOTE

If no transmission is currently taking place, the new character is immediately processed and an interrupt is produced on the next scan.

Table S7.2-5. Line Interrupt Response Addresses

Line/Channel	I/O	Address (Hex)							
		18	19	1A	1B	1C	1D	1E	1F
Line 0 (Channel 1)	Output	100	120	140	160	180	1A0	1C0	1C0
	Input	110	130	150	170	190	1B0	1D0	1F0
Line 1 (Channel 2)	Output	102	122	142	162	182	1A2	1C2	1E2
	Input	112	132	152	172	192	1B2	1D2	1F2
Line 2 (Channel 3)	Output	104	124	144	164	184	1A4	1C4	1E4
	Input	114	134	154	174	194	1B4	1D4	1F4
Line 3 (Channel 4)	Output	106	126	146	166	186	1A6	1C6	1E6
	Input	116	136	156	176	196	1B6	1D6	1F6
Line 4 (Channel 5)	Output	108	128	148	168	188	1A8	1C8	1E8
	Input	118	138	158	178	198	1B8	1D8	1F8
Line 5 (Channel 6)	Output	10A	12A	14A	16A	18A	1AA	1CA	1EA
	Input	11A	13A	15A	17A	19A	1BA	1DA	1FA
Line 6 (Channel 7)	Output	10C	12C	14C	16C	18C	1AC	1CC	1EC
	Input	11C	13C	15C	17C	19C	1BC	1DC	1FC
Line 7 (Channel 8)	Output	10E	12E	14E	16E	18E	1AE	1CE	1EE
	Input	11E	13E	15E	17E	19E	1BE	1DE	1FE

Table S7.2-6. Device Address Strapping

Jumper	Address							
	13	19	1A	1B	1C	1D	1E	1F
9 to:	1	2	3	4	5	6	7	8
12 to:	11	10	11	10	11	10	11	10
15 to:	14	14	13	13	14	14	13	13
18 to:	17	17	17	17	16	16	16	16

Note: Standard address 1B is strapped by etch. If another address is to be used, etch must be cut and jumpers installed per table.

Table S7.2-7. Baud Rate Selection

Baud Rate	Application		Jumper*	Switch Setting*
	RS-232	20-MA		
75	X	X	19-21	
110	X	X	19-20	4
134.5	X	X	19-23	
150	X	X	19-20	5
300	X		19-20	6
600	X		19-22	
1200	X		19-20	7
2400	X		19-20	8
4800	X		19-20	9
9600	X		19-20	10

*Jumper eyelet sets are labeled A through J and switches are labeled S1 through S8 to correspond to channels 1 through 8. Eyelets 19-20 are normally jumpered by etch. If 75, 134.5 or 150 baud rate is to be used, etch must be cut and jumpers installed per table.

Table S7.2-8. External Device Format Selection

Parameter		Jumper*	Jumper*	Switch Setting*
Word Length:	5	24-25	26-27	-
	6	24-25	-	-
	7	-	26-27	-
Stop Bits:	1	-	-	1
	2	-	-	2
Parity:	Disable	-	-	-
	Odd	30-31	32-33	-
	Even	-	32-33	-

*Jumper eyelet sets are labeled A through J, and switches are labeled S1 through S8 to correspond to channels 1 through 8. Parity disable and 8-bit word length are assumed. If other parity and/or word lengths are to be used, jumpers must be installed per table.

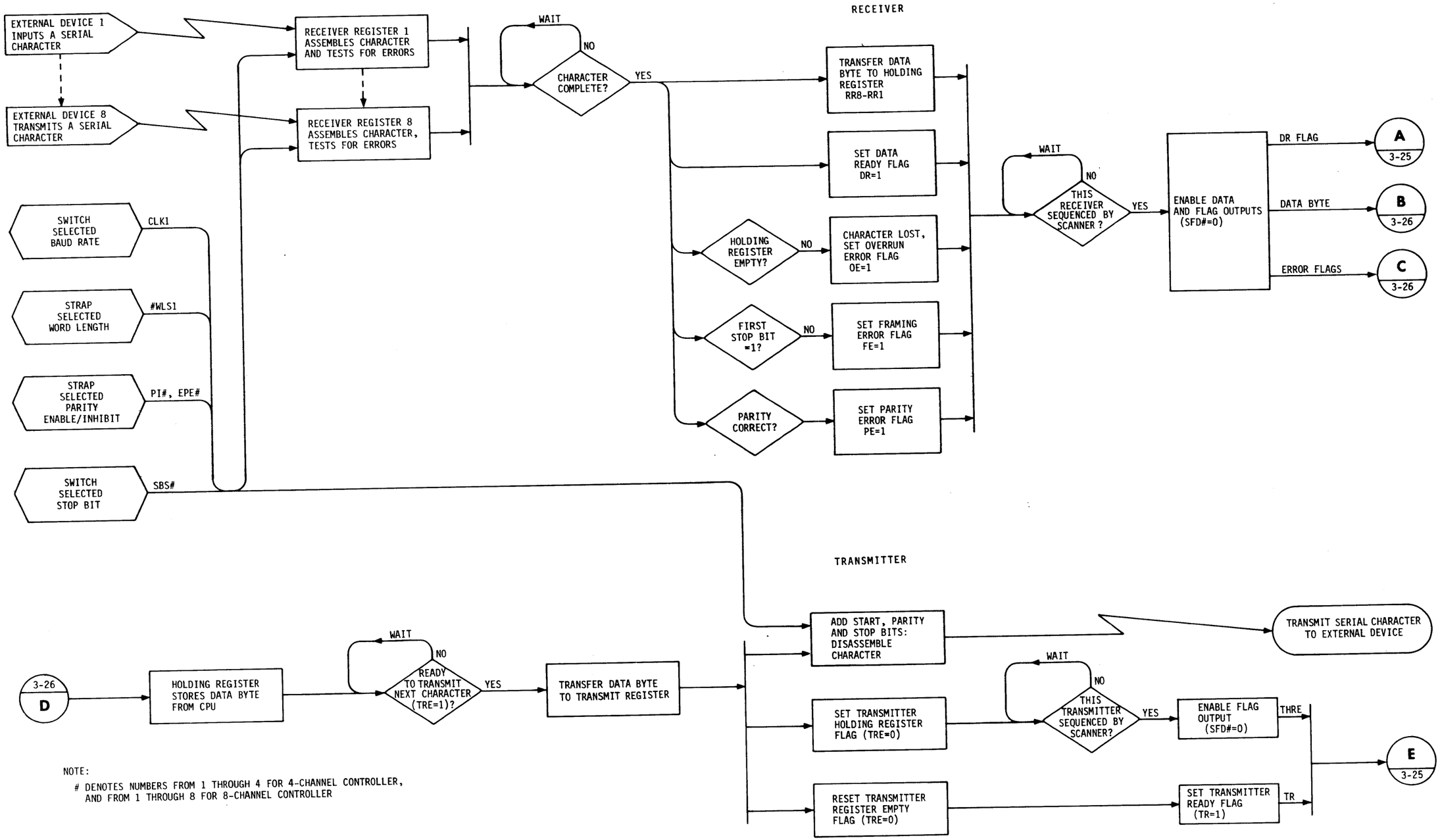


Figure S7.2-1. External Device I/O Flow Diagram, R/T Function

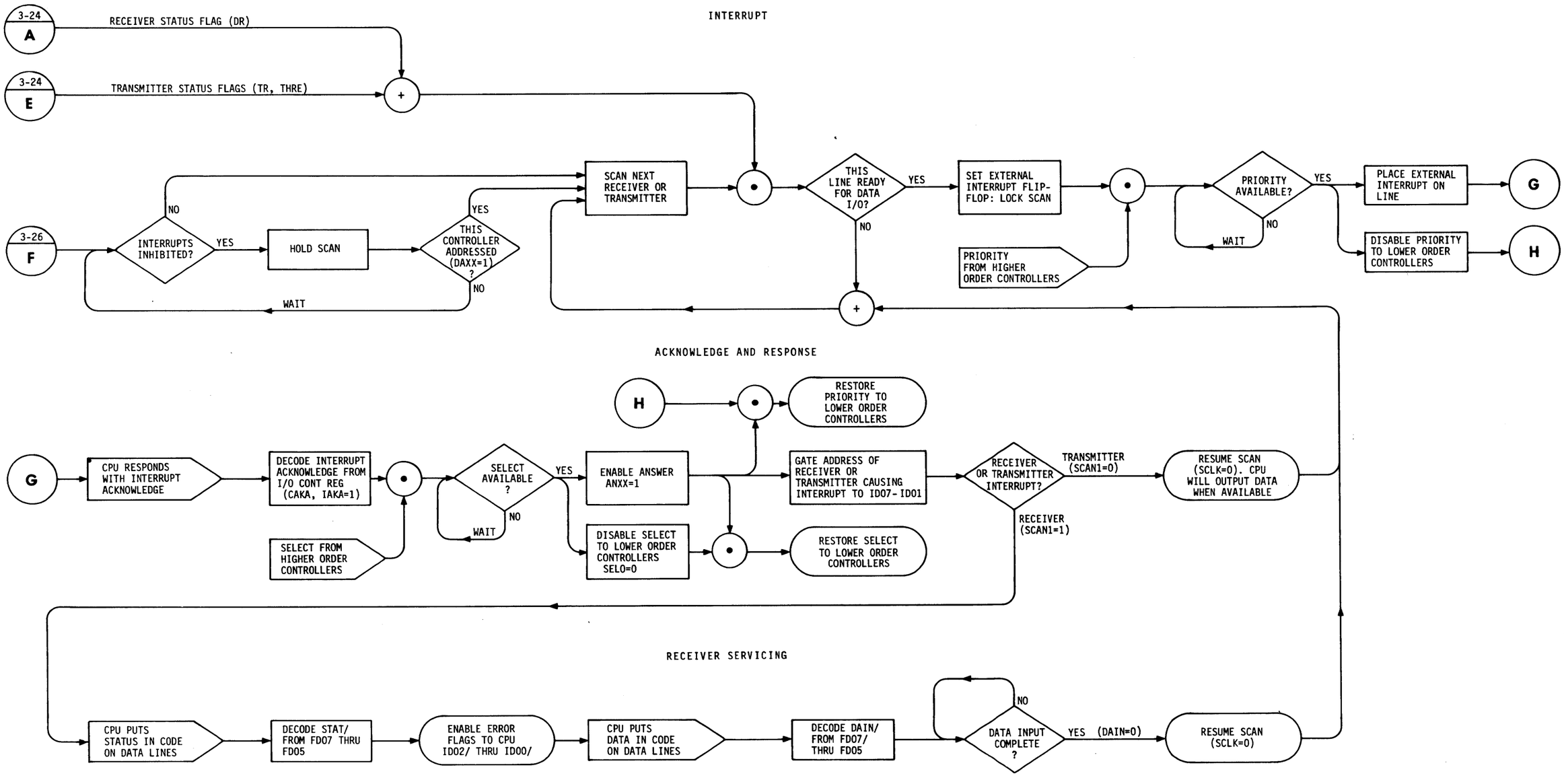


Figure S7.2-2. External Device I/O Flow Diagram, Interrupt and Response

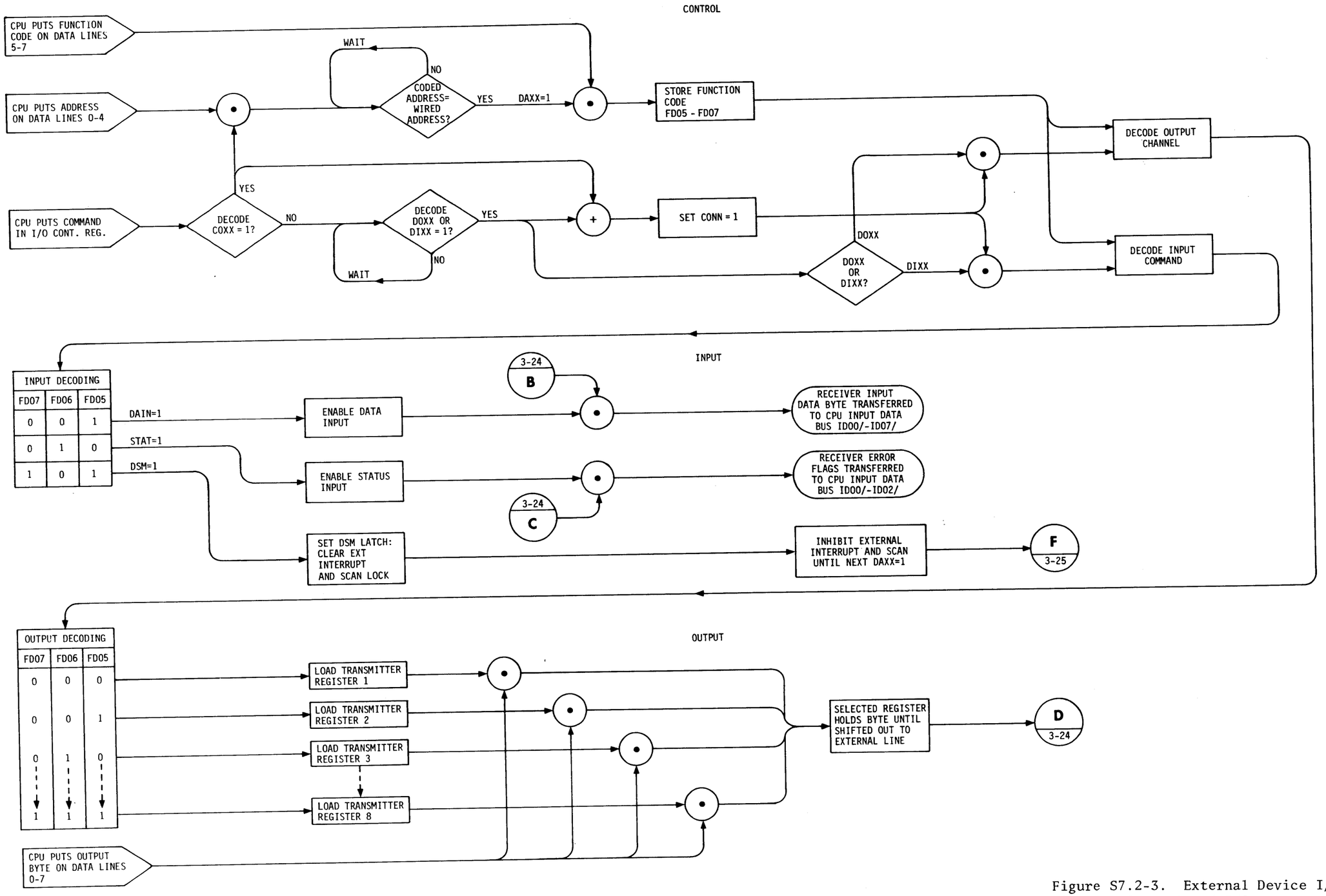


Figure S7.2-3. External Device I/O Flow Diagram, Control and I/O

Original: February 1, 1973

S7.2.2.3 External Device Input/Output. The receives removes the start, stop and parity bits from the incoming serial character and assembles the information bits into an 8-bit parallel data byte (figure S7.2-1). The receiver data byte is immediately transferred to a holding register, allowing the next serial character to be processed. The Data Ready (DR) flag is set and initiates an interrupt the next time the scanner sequences to this receive line. When sequenced, the data byte (RR1 through RR8) and error flag (FE, PE, OE) receiver outputs are enabled. The DR flag is reset when data input to the CPU is initiated. The receiver outputs are disabled when data input is completed.

The transmitter adds start, stop, and parity bits to the information bits set by the CPU. The resulting character is disassembled and serially shifted out to the external device. When transmission of one character is completed, the data byte for the next character is transferred from the holding register to the transmitter register. The TR and THRE flags are set and an interrupt is produced the next time the scanner sequences to this transmit line. The TR flag is reset when the interrupt is acknowledged. The TR flag is not set again until a new character is received from the CPU and has been transmitted (TRE=1, then 0). Therefore, regardless of the number of scan cycles occurring between the initial interrupt and the CPU response, only one interrupt is produced per character.

S7.2.2.4 CPU External Interrupt (figure S7.2-2). Interrupt service is provided for each receiver/transmitter (R/T) line on a sequential basis. The scanner enables status flags from one line at a time. When data service is required for that line (DR or THRE and TR active) an interrupt is produced and the scanner is locked. If interrupts have not been disarmed (DSM inactive) and this controller has priority (PRIN active), the interrupt is sent to the CPU (EINT). The CPU responds with an interrupt acknowledge. If the Select line to this controller is enabled (SELI active), the interrupt response address of the line causing the interrupt is sent to the CPU. When the requesting line is a transmitter, immediate CPU response is not required. The scanner is unlocked and sequences to the next line. When the requesting line is a receiver, the scanner remains locked until the CPU has extracted the receiver error flags (status input) and character information bits (data input).

When interrupts are disarmed (DSM active), the scanner is locked and the interrupt output is disabled. Interrupts are re-enabled by the next CPU command addressed

to the controller. When DSM is reset, the scanner is unlocked and interrupt sequencing resumes from the point at which DSM occurred.

S7.2.2.5 CPU Input/Output (figure S7.2-3). The first byte of the two-byte I/O command is a control byte. The 0 data byte contains the device address and the three-bit function code. If the encoded address matches the strapped address of the controller (DAXX=1), the function code is latched and the connector for service flip-flop is set on the trailing edge of the function/device control signal (COXX).

The second byte of an input data command consists of only the data-in control code. Upon receipt of the data-out control code (DOXX), the latched function bits are decoded. The resulting control signal determines the next action. The data-input (DAIN) function code causes the receiver data byte to be gated onto the input lines. The status-input function code (STAT) causes receiver error flags to be input. Disarm interrupt function code DSM sets a latch upon receipt. The latch is reset only when the CPU sends another command addressed to this controller (DAXX active). While the latch is set, no interrupts can be produced.

The second byte of an output data command contains the information bits for a transmit character. Upon receipt of the data-out control code (DOXX active), the latched function bits are decoded to obtain the designation of the transmitter for which the character is intended (THRL1 through THRL8). The transmitter data byte is transferred to the designed transmitter.

S7.2.3 LOGIC DESCRIPTIONS

The following paragraphs describe the functional operation of the major circuits contained on the 8-Channel Communications Controller module. The circuit descriptions provided are for the following major circuits:

CPU I/O Control Circuits
Interrupt Control Circuits
Timing Circuits
Receiver/Transmitter Circuits
CPU I/O Data Multiplexer Circuits

The four-digit numbers (1210 through 1224 for the 4-channel controller and 1110 through 1131 for the 8-channel controller) referenced in the following paragraphs refer to the logic diagrams contained in LD 1010. Referencing will be to the applicable diagram for each controller; i.e., 1260/1160 indicates figure 1260 for the 4-channel controller and 1160 for the 8-channel controller.

S7.2.4 CPU I/O CONTROL CIRCUITS

The following subparagraphs describe the operation of the CPU I/O Control Circuits. These circuits are divided into three functional areas and each is described in detail. The CPU I/O Control Circuits consist of:

CPU I/O Data Processing
CPU I/O Data Controls
CPU Functional Controls

S7.2.4.1 CPU I/O Data Processing. Output bus signals OD00/ through OD07/ are supplied through line receiver/inverters (1210/1110) to the following points: R/T circuits (bits 7 through 0 transmit character information bits); Address Decoder (bits 4 through 0, device address); and Functional Bit Latch (bits 7 through 5, I/O function code).

The address bits are received and decoded during a control byte (COXA/ active) (1211/1111). When the encoded address is 18_{hex}, or greater (OD04 and OD03 = 1), the MSB input of the 1-of-10 Address Decoder is set false. The MSB false enables the three LSB's (OD02 through OD00) to set one of the lower order (0 through 7) output lines false. Each of these lines corresponds to an address between 18_{hex} and $1F_{hex}$. If the false line corresponds to the strapped address, device address detected (DAXX) is active. When DAXX is active, function code bits OD07 through OD05 are loaded into the Function Bit Latches. The latches are set on the trailing edge of DAXX.

S7.2.4.2 CPU I/O Data Controls. The I/O Control lines are terminated and supplied through line receiver/inverters to the 1-of-10 Control Decoder (1211/1111). Only the lower order (0 through 7) outputs of the decoder are used. The control bits can only be recognized when the decoder MSB input latch is set false. The latch is set false by the first phase 2 clock pulse after a control code is placed on the lines. Since the I/O Control lines are set by the phase 1 clock in the CPU, 20-nanosecond delay is introduced, allowing the lines to settle before the code is read. The decoder outputs are distributed to other circuits through a series of inverter and unloading gates. The Concurrent Acknowledge (CAKA/) and Interrupt Acknowledge (IAKA/) signals are ORed to produce composite acknowledge signal ACK/. The Master Reset (MRST) line is terminated, buffered, and inverted for distribution to controller circuits as MRSTA and its complement.

S7.2.4.3 CPU I/O Function Controls. The connect for service (CONN) flip-flop (1212/1112) is set true on the trailing edge of COXX when the control byte being processed is addressed to this controller (DAXX active). CONN is reset when the data output is completed (DOXX/ active) or data input is completed (DIXX/ active). If data output is being processed (DOXX active), CONN enables the output channel decoder (MSB input false). The decoer output line (THRL1 through THRL8) corresponding to the selected transmit channel is set false. Load Transmitter Holding Register strobes the CPU output data byte into the transmitter. If data input is being processed (DIXX active), CONN enables the Input Mode Decoder (MSB input false). The decoder output line corresponding to the input mode code is set false. The Disarm Interrupt (DSM) line is latched. The latch (1213/1113) is set when DSM occurs and is reset (re-enabling interrupts) when another command is addressed to the controller (DAXX active).

S7.2.5 INTERRUPT CONTROL CIRCUITS

A receiver interrupt is generated during receiver scan time (SCAN1 active) if one of the receivers (1213/1113) indicates that data is ready for input to the CPU (DR=1). The external interrupt flip-flop (EIFF) is set with the first delayed clock after the set external interrupt gate (SEIN) becomes active. EINT is gated out to the CPU if no higher priority controller has control (PRIN/ inactive). The priority out signal to lower priority devices is disabled (PROT/ active) while an interrupt is in progress. The CPU responds to the interrupt with interrupt acknowledge (ACK/ active). If the Select line from higher priority devices is enabled (SELI/ inactive), interrupt response (Answer) is enabled (ANXX active) and the Select line output to lower priority devices remains disabled (SELO/ inactive). The interrupt flip-flop remains true (delayed clock is disabled during interrupt) until reset at the end of response time by the trailing edge of ANXX/. A transmitter interrupt is produced during transmitter scan time (SCAN1 active) under the following conditions:

1. Next character has not been received (THRE active).
2. Next character has not yet been requested (TR active) by this transmitter (SCAN2 through SCAN4).

Disarm interrupt (DSM active) holds the interrupt flip-flop false until the disarm condition is removed. Master Reset (MRSTA) strobes the flip-flop false when power is applied to prevent extraneous interrupts.

Receipt of Concurrent Acknowledge from the CPU (ACK/ or CACK/ active) causes Select (SELI/) to be passed through (SELO/) the controller. CACK has no other function, since the controller does not utilize the block transfer I/O mode. Select is also passed through for any Interrupt Acknowledge (IACK or ACK/) which occurs in response to an interrupt by a lower priority device (EIFF/ or SELI/).

S7.2.6 TIMING CIRCUITS

The following three paragraphs describe the operation of the Timing Circuits. These circuits are divided into three functional areas and each is described in detail. The Timing Circuits consist of:

Timing Generator
Receiver Transmitter Sequencer
Baud Rate Clock Generator

S7.2.6.1 Timing Generator. The Timing Generator (1214/1114) consists of a clock pulse generator and a 16-line scanner. The output of oscillator Y1 feeds the clock pulse counter which is mechanized in a divide-by-six configuration. One of the 0.614-MHz outputs is the rate clock for a series of baud rate counters, the other drives the shift register. The shift register produces the scan clock and a delayed clock. The delayed clock is used to provide settling delays for interrupt and scan lock flip-flops.

The scan counter is a four-bit binary counter. The three MSB's (SCAN4 through SCAN2) represent channel designations 1 through 8. The LSB divides each channel into transmitter (SCAN1 = 0) and receiver (SCAN1 = 1).

Both the scan counter and shift register are disabled by the Scan Lock flip-flop during interrupts. The flip-flop is set by the first delayed clock after an interrupt is initiated (SEIN active). The output of the flip-flop (SCLK/ active) locks the shift register at all ones (via the parallel load input) and locks the clock input to the scan counter false. The flip-flop is reset by the trailing edge of DAIN, for a receiver interrupt (SCAN1 = 1), or by the trailing edge of ANXX for a transmitter interrupt. Clock pulse output is resumed and the scan counter advances to the next state.

The scan lock reset term is ORed with Master Reset to form a scan control (SCCL/) output to the R/T sequencer.

S7.2.6.2 Receiver/Transmitter Sequencer. Two decoders derive control terms from the scan count to multiplex receiver/transmitter outputs (1215/1115). One of the output lines of the R/T Select Decoder is false for each scan count, enabling the outputs of the corresponding R/T channel. The R/T Reset Decoder is enabled at the beginning of response time (ANXX active), for a transmitter interrupt, or the beginning of data input (DAIN active) for a receiver interrupt. The decoder output corresponding to the receiver or transmitter causing the interrupt is set false to reset the ready flag (DR or TR). The decoder is also enabled during master reset to prevent extraneous ready flags from occurring when power is applied.

S7.2.6.3 Baud Rate Clock Generator. The Rate Clock (RCLK) is counted down to produce ten individual clock pulses. Each clock pulse is sixteen times the required baud rate (1216/1116).

S7.2.7 RECEIVER/TRANSMITTER CIRCUITS

The following five paragraphs describe the operation of the Receiver/Transmitter Circuits. These circuits are divided into five functional areas and each is described in detail. The Receiver/Transmitter Circuits consist of:

External Device Format Select
External Device Inputs
Receiver/Transmitter
Transmitter Status Register
External Device Outputs

S7.2.7.1 External Device Format Select. A two-section switch and three sets of strapping eyelets establish the format of serial characters processed by each channel (1217/1117 and 1118). Table S7.2-6 and S7.2-7 list the switch and strap settings for the baud rate and format options.

S7.2.7.2 External Device Inputs Each external device is provided with nine lines for interface with the controller (table S7.2-1). Six of these lines are receiver inputs or operational terminations (1218/1129). Each receiver input (C) line is provided with a line receiver capable of handling 20 milliamps or RS-232 type inputs. A low frequency filter is provided for 20-milliamp operation. The filter is enabled by an external jumper from F line to E line (ground).

S7.2.7.3 Receiver/Transmitter (figures 1219 through 1222/1120 through 1127). Each Receiver/Transmitter (1219/1120) is a monolithic MOS/LSI device containing the following functions:

a. Receiver.

1. Start Bit Detect Circuit. Monitors serial input (RI) to detect a 1 to 0 transition (start bit) while in the idle state. The start bit is verified by locating the center of the bit and checking for 0. If the start bit is valid (still 0), decoding of the remainder of the character is enabled.

2. Receiver Register. Assembles character bits as the center of each bit is detected.

3. Receiver Holding Register. Stores character information bits upon completion of character assembly. Data Ready (DR) flag indicates holding register contains a character for output to the CPU. DR is reset when the character is extracted by the CPU. When DR is not reset before the next character is transferred to the holding register, a character is lost and the overrun error (OE) flag is set. OE remains set until DR is reset and the next character is transferred to the holding register.

4. Error Controls. Detect framing and parity errors in the input character. If the first stop bit is not 1, the Framing Error (FE) flag is set. When parity does not match that selected by the format select inputs, the Parity Error (PE) flag is set. The FE and PE flags remain set until the next character is transferred to the holding register.

b. Transmitter.

1. Transmitter Holding Register. Stores character information bits from the CPU, strobed in by Load Transmitter Holding Register (THRL). Character information bits are automatically transferred to the transmitter register when it becomes empty (TRE = 1). The Transmitter Holding Register Empty (THRE) flag is set and held until a new character is received from the CPU.

2. Transmitter Register. Accepts information bits from the holding register. Start, stop, and parity bits are added and the complete serial character is shifted out to the external device. Timing and character encoding (stop and parity bits) are controlled by the format select inputs.

c. Control Register. Accepts format selection inputs and controls character processing in the receiver and transmitter.

Table S7.2-8 summarizes R/T functions as related to specific signals.

Table S7.2-8. Receiver Transmitter Functions

Signal	Mnemonic	Function
Transmitter Holding Register Empty	THRE	High-level output voltages on this line indicate the Transmitter Holding Register has transferred its contents to the Transmitter Register and can be loaded with a new character.
Word Length Select	WLS2-WLS1	These two lines select the character length (5, 6, 7 or 8 bits).
Parity Inhibit	PI	High-level input voltages on this line inhibit the parity generation and verification circuits. The stop bit(s) immediately follow the last data bit on transmission if parity is inhibited. Low-level input voltages enable the parity generation and verification circuits. PI will, when a high-level input voltage is applied, also clamp the PE line (pin 13) to a low-level output voltage.
Even Parity	EPE	This line selects either even or odd parity to be generated by the transmitter and checked by the receiver. High-level input voltages select even parity and low-level input voltages select odd parity.
Stop Bit(s) Select	SBS	This line selects the number of stop bits generated after the parity bit during transmission. High-level input voltages on this line select two stop bits, and low-level input voltages select a single stop bit.
Control Register Load	CRL	High-level input voltages on this line load the Control Register with the control bits (WLS1, WLS2, EPE, PI, SBS). This line is wired to a high-level input voltage.
Status Flags Disconnect	SFD	High-level input voltages applied to this pin disconnect the PE, FE, OE, DR and THRE circuit outputs.
Receiver Input	RI	Serial input data received on this line enters the Receiver Register at a point determined by the character length, parity, and the number of stop bits. High-level input voltages must be present when data is not being received.
Data Received	DR	High-level output voltages indicate an entire character has been received and transferred to the Receiver Holding Register.
Data Received Reset	DRR	Low-level output voltages applied to this line reset the DR line.

Table S7.2-8. Receiver Transmitter Functions (continued)

Signal	Mnemonic	Function
Receiver Holding Register Data	RR1-RR8	The contents of the Receiver Holding Register appear on these lines in parallel when a low-level input voltage is applied to RRD. Selection of a word length less than 8 bits causes the MSB of the character to be forced to a low-level output voltage. The character is right justified. RR1 (pin 12) is the LSB of the character.
Receiver Register Clock	RRC (CLK)	This clock is 16 times faster than the desired receiver shift rate.
Receiver Register Disconnect	RRD (SFD)	High-level input voltages applied to this line disconnect Receiver Holding Register outputs from the RR8 thru RR1 data outputs (pins 5-12).
Parity Error	PE	The status of the parity verification circuit appears on this line when a low-level input voltage is applied to the Status Flags Disconnect (pin 16) control line. Wired-OR capability is provided on this line allowing PE lines from other R/T's to be OR-tied. High-level output voltages on this line (under the conditions above) indicate a Parity Error in the received parity bit as programmed by Even Parity Enable control line (pin 39). The status is updated each time a character is transferred from Receiver Register to Receiver Holding Register.
Framing Error	FE	Status of the stop bit detection circuit appears on this line when a low-level input voltage is applied to Status Flag Disconnect (pin 16) control line. Wired-OR capability is provided on this line allowing FE lines from other R/T's to be OR-tied. High-level output voltage indicate received character has no valid stop bit (i.e., the bit following the parity bit is not a high-level input voltage). The status is updated each time a character is transferred from Receiver Register to Receiver Holding Reg.
Overrun Error	OE	Status of Data Received circuit appears on this line when a low-level input voltage is applied to Status Flag Disconnect (pin 16) control line. Wired-OR Capability is provided on this line allowing OE lines from other R/T's to be OR-tied. High-level output voltages indicate previously received character was not read (DR line not reset) before present character was transferred to the Receiver Holding Register.

Table S7.2-8. Receiver Transmitter Functions (continued)

Signal	Mnemonic	Function
Master Reset	MR (MRST)	This line is strobed to a high-level input voltage to clear the logic after power turn-on. It resets all registers and set the serial output line to a high-level output voltage.

S7.2.7.4 Transmitter Status Register. Each register flip-flop produces a flag (TR) to indicate that the corresponding transmitter is ready for a new character (1215/1115). The flip-flop is set on the trailing edge of Transmitter Register Empty (TRE, indicating that transmission of the last character received from the CPU has begun). The flip-flop is reset when interrupt is processed (TRR/ active), and is not set again (TRE active) until a new character is received from the CPU.

S7.2.7.5 External Device Outputs. Each transmitter output line is provided with two line drivers, one for RS-232 devices and one for 20-milliamp current loop devices (1218/1129).

S7.2.8 CPU DATA MULTIPLEXER

The multiplexer (1223/1130) gates data onto the Byte I/O input lines in response to three control signals, as follows:

```
STAT:  Bits 7 through 3 = No data (zeros)
       Bit 2            = Parity Error flag
       Bit 1            = Framing Error flag
       Bit 0            = Overrun Error flag

DAIN:  Bits 7 through 0 = RR8 through RR1

ANXX:  Bits 7 through 5 = Strapped device address
       Bit 4            = Receiver/transmitter bit (SCAN1)
       Bits 3 through 1 = Channel designation (SCAN4 to SCAN2)
       Bit 0            = 0
```

S7.3 PARTS LIST

All electronics parts for the SSBR 4-channel controller are illustrated on figure S7.1-1 and for the SSBR 8-channel controller on figure S7.1-2. The parts for each are listed on the following pages.

Parts List for SSBR Four-Channel Controller (figure S7.1-1)

Ref. Designation	Part Number	Description	Qty	Vendor
C1,C2,C17,C26	150D475X0010A2	Capacitor, 4.7uf, 10v	4	Sprague
C3 thru C16	TG-S10	Capacitor, 0.01uf, 100v	14	Sprague
C18 thru C21	150D474X9020A2	Capacitor, 0.47uf 20v 10%	4	Sprague
CR1	1N4742A	Diode	1	
CR2	1N5233	Diode	1	
J2	D20000307-49	Header, 50-pin	1	Microdata
R1,R4,R6	RC07GF221J	Resistor, 220 ¼w 5%	3	
R2,R3,R8,R9	RC07GF102J	Resistor, 1K ¼w 5%	4	
R5,R7	RC07GF301J	Resistor, 300 ¼w 5%	2	
R10,12,14,16,18,20,22, 24	RC20GF132J	Resistor, 1.3K ½w	8	
R11,13,15,17,19,21,23, 25	RC20GF151J	Resistor, 150 ½w	8	
S1 thru S4	C42315A61-A2	Switch	4	Seimons
S1 thru S4	C42315A61-C9	Stop, Switch	4	Seimons
Y1	SO-3537	Osc, 3.686MHz .450 high	1	Monitor Prod.
2A,2B,4B,4L,7H	SN7400N	Integrated Circuit	5	TI
5E	SN7402N	Integrated Circuit	1	TI
3A,2D,4C,5J,7D	SN7404N	Integrated Circuit	5	TI
3E,4K	SN7410N	Integrated Circuit	2	TI
2F,2H,2J,2K,2L,4H	SN7438N	Integrated Circuit	6	TI
3B,3C,4D,4E,4F	SN7442N	Integrated Circuit	5	TI
5K	SN7450N	Integrated Circuit	1	TI
3D	SN7475N	Integrated Circuit	1	TI
2E	SN74L04N	Integrated Circuit	1	TI
3H,3K,3L,4A,4J,5D,5H	SN74107N	Integrated Circuit	7	TI
5B,5C,5F,6C,6E,6F,6H,6J	SN74161N	Integrated Circuit	8	TI
5A1,5A2	SN75452N	Integrated Circuit	2	TI
6D	U7B930059X	Integrated Circuit	1	Fairchild
3J	U6N931159X	Integrated Circuit	1	Fairchild
3F	U7B931259X	Integrated Circuit	1	Fairchild
1A,1B,1D,1E	CS20001120	Integrated Circuit	4	Microdata
--	4-40NCx5/16LG	Screw, Pan Hd, Cross Rec	2	
--	#4	Washer, Split Lock	2	
--	D20001065	Cable Retainer	1	Microdata
--	47743	Contact, Female	40	Berg Elect.
--	D20002612	Printed Wiring Board	1	Microdata
6A	MC1488L	Integrated Circuit	1	Motorola
6B	MC1489L	Integrated Circuit	1	Motorola
2C	500007-14	Resistor Module, 10K	1	Microdata
Mates with J2	A20001114-49	Connector	1	Microdata

Parts List for SSBR Eight-Channel Controller (figure S7.1-2)

Ref. Designation	Part Number	Description	Qty	Vendor
C1,C2,C17,C26	150D475X0010A2	Capacitor, 4.7uf, 10v	4	Sprague
C3 thru C16	TG-S10	Capacitor, 0.01uf 100v	14	Sprague
C18 thru C25	150D474X9020A2	Capacitor, 0.47uf 20v 10%	8	Sprague
CR1	1N4742A	Diode	1	
CR2	1N5233	Diode	1	
J2	D20000307-49	Header, 50 pin	1	Microdata
J3	D20000307-48	Header, 50 pin	1	Microdata
R1,R4,R6	RC07GF221J	Resistor, 200 ¼w 5%	3	
R2,R3,R8,R9	RC07GF102J	Resistor, 1K ¼w 5%	4	
R5,R7	RC07GF301J	Resistor, 300 ¼w 5%	2	
R10,12,14,16,18,20,22, 24	RC20GF132J	Resistor, 1.3K ½w	8	
R11,13,15,17,19,21,23, 25	RC20GF151J	Resistor, 150 ½w	8	
S1 thru S8	C42315A61-A2	Switch	8	Seimons
S1 thru S8	C42315A61-C9	Stop, Switch	8	Seimons
Y1	SO-3537	Osc, 3.686 MHz, 0.450 high	1	Monitor Prod.
6A,6L	MC1488L	Integrated Circuit	2	Motorola
6B,6K	MC1489L	Integrated Circuit	2	Motorola
2A,2B,4B,4L,7H	SN7400N	Integrated Circuit	5	TI
5E	SN7402N	Integrated Circuit	1	TI
3A,2D,4C,5J,7D	SN7404N	Integrated Circuit	5	TI
3E,4K	SN7410N	Integrated Circuit	2	TI
2F,2H,2J,2K,2L,4H	SN7438N	Integrated Circuit	6	TI
3B,3C,4D,4E,4F	SN7442N	Integrated Circuit	5	TI
5K	SN7450N	Integrated Circuit	1	TI
3D	SN7475N	Integrated Circuit	1	TI
2E	SN74L04N	Integrated Circuit	1	TI
3H,3K,3L,4A,4J,5D,5H	SN74107N	Integrated Circuit	7	TI
5B,5C,5F,6C,6E,6F,6H,6J	SN74161N	Integrated Circuit	8	TI
5A1,5A2,5L1,5L2	SN75452N	Integrated Circuit	4	TI
6D	U7B930059X	Integrated Circuit	1	Fairchild
3J	U6N931159X	Integrated Circuit	1	Fairchild
3F	U7B931259X	Integrated Circuit	1	Fairchild
1A,1B,1D,1E,1F,1H,1K,1L	CS20001120	Integrated Circuit	8	Microdata
--	D20002612	Printed Wiring Board	1	Microdata
2C	500007-14	Resistor Module, 10K	1	Microdata
--	4-40NCx5/16LG	Screw, Pan Hd, Cross Rc	2	
--	#4	Washer, Split Lock	2	
--	D20001065	Cable Retainer	1	Microdata
Mates with J2	A20001114-49	Connector	1	Microdata
Mates with J3	A20001114-48	Connector	1	Microdata
--	47743	Contact, Female	80	Berg Elect.