OBSOLETE

HEWLETT hp PACKARD

OPERATING AND SERVICE MANUAL

2152A
FLOATING POINT
PROCESSOR

02152-90004

HEWLETT-PACKARD COMPANY
11000 WOLFE ROAD, CUPERTINO, CALIFORNIA, U.S.A.

Printed: SEPT 1970

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

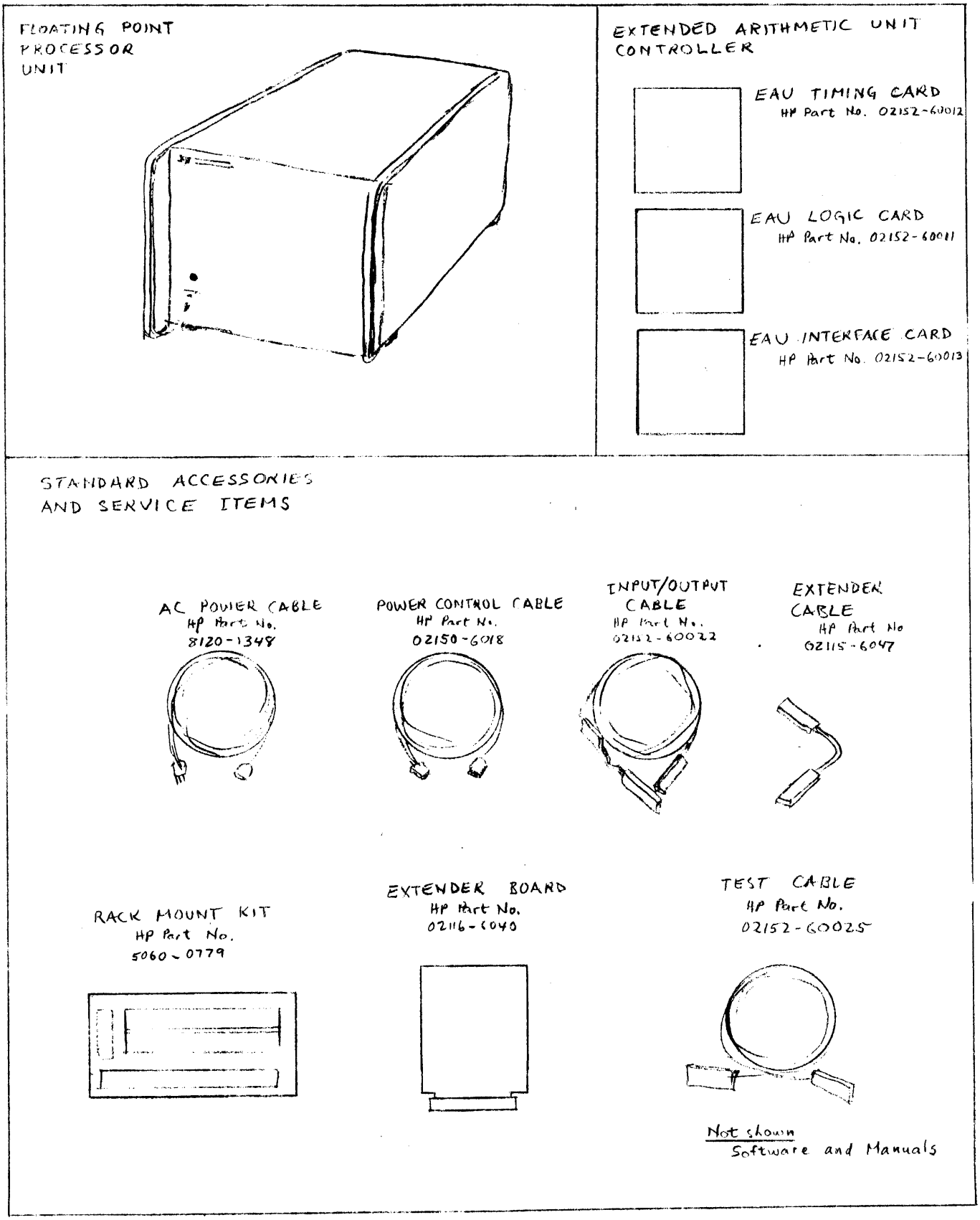


Figure 1-1. HP 2152A Floating Point Processor and Accessories

SECTION I

GENERAL INFORMATION

1-1. INTRODUCTION

1-2. This operating and service manual covers general information, installation, operating and programming information, theory of operation, maintenance procedures, and replaceable parts lists for the Hewlett-Packard 2152A Floating Point Processor. (See figure 1-1.) Two theory of operation sections are included, one for the processor unit itself (section IV) and one for the controller (section V). The remaining sections treat the processor and controller as a single entity.

1-3. For brevity, this manual will frequently use the abbreviation FPP to designate the Floating Point Processor, and the term FPP unit to designate the processor exclusive of the controller (i. e., only the circuitry physically contained within the cabinet module).

1-4. GENERAL DESCRIPTION.

1-5. The HP 2152A Floating Point Processor extends the hardware computing capability of the 2115/2116-series computers to include high-speed, extended-precision mathematical and trigonometric functions. Precision is increased to 40 bits of mantissa with 8 bits of exponent, which is equivalent to an accuracy of 12 decimal digits. Processing time is reduced to the range of 50 to 100 microseconds for floating-point add/subtract/multiply/divide and a range of 20 to 200 microseconds for the trigonometric functions.

1-6. The extended precision is accomplished through the use of a 48-bit accumulator. The instruction set includes triple-store and triple-load operations to transfer the triple-length quantities between this accumulator and three consecutive memory locations.

1-7. The greatly increased speed is accomplished through implementation of 13-field microprogramming of the algorithms, operating from a read-only memory. The algorithms for this application employ unique efficiency techniques of firmware programming, reducing the amount and complexity of the hardware required while affording the increase in speed. Additionally, the FPP is interfaced through the EAU controller directly to the computer CPU, rather than through the I/O system. By increasing the speed of the data transfers in this manner, overall system speed is also further increased.

1-8. To provide compatibility with double-precision data which may be used in conjunction with the triple-precision capability of the FPP, the instruction repertoire includes instructions which will automatically perform the necessary format conversions. Both integer and floating-point double precision data may be used.

1-9. In addition to the 30 floating-point instructions provided by the FPP unit, the extended arithmetic unit (EAU) controller adds 10 double-length integer operations, including long shifts and rotates. For these 10 instructions, the operand data and results are given in the computer A- and B-registers. The FPP unit is not involved. However, all 40 instructions comprise the repertoire of instructions added to the system by the HP 2152A Floating Point Processor.

1-10. Functionally, the FPP can be regarded as a calculator under the control of a computer. In the same way that a human operator manually uses a free-standing calculator, the computer enters a number into the unit, then issues a command (equivalent to pressing a button) which says, for example, "calculate the sine of this number." Analogous to the human operator example, the answer appears in far less time than the computer would have taken to calculate it. Thus the benefits to the computer are the same as the benefits derived from a calculator by a human operator: speed and efficiency.

1-11. As shown in figure 1-2, the computer controls the FPP unit by instructions issued to the EAU controller installed in the computer mainframe. The commands from this controller initiate specific firmware routines (programs permanently stored in the FPP read-only-memory). These routines are responsible for interpreting the commands, executing the commanded function, and transferring data in and out of the unit.

1-12. One typical mode of operation was mentioned above; i. e., load data into the unit, then command the unit to execute a function (such as sine) on that data. The input data goes into the X-register of the FPP

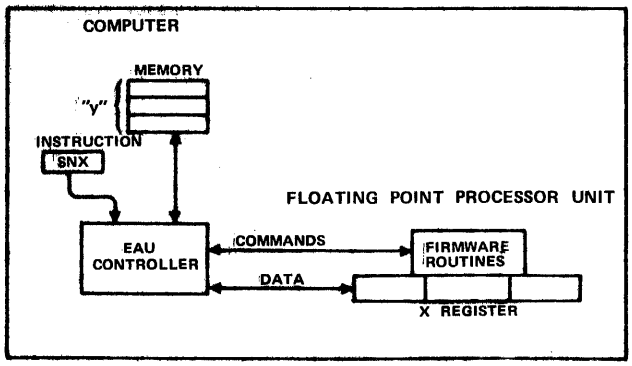


Figure 1-2. Floating Point Processor as a Calculator

unit; then, when the unit signals the controller that the function has been executed, the answer resides in the X-register, replacing the input data. The X-register is a triple-length register (48 bits).

1-13. A second mode of operation uses two operands. The first operand is initially loaded into the X-register. Then the instruction is issued to, for example, add the quantity y to the contents of the X-register. Since y is in memory, the controller must fetch this second operand (two or three memory words), then send it to the FPP. This is followed by an add command, which tells the unit to add the two numbers. Again, the answer appears in the X-register of the FPP.

1-14. Since, in both cases, the answer does not automatically return to the computer, a final store instruction is necessary in order to transfer the result to computer memory. Thus a complete instruction sequence for an add operation would typically consist of:

```
LDX  x
ADX  y
STX  z
```

1-15. COMPATIBILITY.

1-16. In general, the HP 2152A Floating Point Processor may be used with any Hewlett-Packard computer that is wired to accept the Extended Arithmetic Unit option. This excludes the 2114-series computers, and HP 2116A Computers with serial prefixes of 747- or lower. The FPP power supply includes circuitry necessary to operate with the computer power fail circuits, including the restart option.

1-17. IDENTIFICATION.

1-18. Hewlett-Packard uses four digits and a letter (0000A) for standard instrument model designations. Options installed as factory modifications to a standard unit are identified by a three digit suffix following the model designation (0000A-000). If the model number and option suffix on your unit do not agree with those on the title page of this manual, there are differences between your unit and the unit described in this manual. These differences are described in change sheets and manual supplements available at the nearest HP Sales and Service Office.

1-19. A two-section eight-digit serial number (000-00000) located on the rear panel identifies each unit. The first three digits are a prefix number used to identify a particular unit configuration. This prefix does not change unless unit changes are made. The last five digits identify each specific unit. If the serial number prefix on your unit does not agree with that shown on the title page of this manual, there are differences between your unit and the unit described in this manual. These differences are described in change sheets and manual supplements available at the nearest HP Sales and Service Office.

1-20. Printed-circuit card revisions are identified by a letter, a date code, and a division code stamped on the card (e.g., A-1055-22). The letter code identifies the version of the etched trace pattern on the unloaded card. The date code (four middle digits) refers to the electrical characteristics of the loaded card. The division code (last two digits) identifies the Hewlett-Packard division that manufactured the card. If the date code stamped on the printed-circuit card does not agree with the date code shown on the schematics in this manual, there are differences between your card and the card described in this manual. These differences are described in manual supplements available at the nearest HP Sales and Service Office.

1-21. SPECIFICATIONS.

1-22. Specifications for the HP 2152A Floating Point Processor are listed in table 1-1.

Table 1-1. Floating Point Processor Specifications

OPERATIONAL SPECIFICATIONS

Number of Instructions:	6	Transfer instructions (Load/Store)
	8	Triple-length floating point arithmetic
	16	Function instructions (trigonometric, etc.)
	10	Double-length integer arithmetic and shifts
	40	Total

Data Types: Extended Floating Point (39 bits mantissa, plus sign; 7 bits exponent, plus sign)

Floating Point (23 bits mantissa, plus sign; 7 bits exponent, plus sign)

Double-word Integer (32 bits)

**Execution Times (microseconds, maximum):

LDD	Load Double (integer)	40	ATX	Arctangent	100
LDF	Load Floating Point	30	HCX	Hyperbolic Cosine	125
LDX	Load Extended Floating Point	30	HSX	Hyperbolic Sine	125
STD	Store Double (integer)	27	HTX	Hyperbolic Tangent	200
STF	Store Floating Point	16	AHT	Archyperbolic Tangent	125
STX	Store Extended Floating Point	16	EXX	Exponential	125
ADF	Add Floating Point	43	LNX	Natural Logarithm	125
ADX	Add Extended Floating Point	43	SRX	Square Root	100
SBF	Subtract Floating Point	50	FIX	Round to nearest Integer	11
SBX	Subtract Extended Floating Point	50	RNX	Round to 24 Bits	14
MPF	Multiply Floating Point	100	MPY	Multiply Integer	19*
MPX	Multiply Extended Floating Point	100	DIV	Divide Integer	21*
DVF	Divide Floating Point	100	DLD	Double Load (A-/B-reg)	6*
DVX	Divide Extended Floating Point	100	DST	Double Store (A-/B-reg)	6*
ABX	Absolute Value	20	ASR	Arithmetic Shift Right	8*
ENX	Entier	17	ASL	Arithmetic Shift Left	8*
CMX	Complement	20	LSR	Logical Shift Right	8*
CSX	Cosine	160	LSL	Logical Shift Left	8*
SNX	Sine	160	RRR	Rotate Right	8*
TNX	Tangent	225	RRL	Rotate Left	8*

*Computer cycle time of 1.6 microseconds assumed (e.g., 2116-series)

**Execution times shown assume the following:

1. All operands normalized (except for LDD); if otherwise, add 12 microseconds per operand.
2. No indirect addressing; if otherwise, add 1.6 microseconds per level of indirect addressing.
3. No direct memory access (DMA) transfers; if otherwise, add 3.2 microseconds per transfer

Table 1-1. Floating Point Processor Specifications (Continued)

ELECTRICAL SPECIFICATIONS

Interface Logic Levels (at FPP unit data connector):

Input Data and Encode

"1" state +1.25V or more positive (+2.5V nominal)
"0" state +0.5V or more negative (-0.5V nominal)

Output Data and Flag

"1" state +2.25V or more positive (+2.5V nominal)
"0" state -0.36V or more negative (-0.5V nominal)

Test Logic Levels (at test board A7):

Input Data and Encode

"1" state +2.0V or more positive (+2.5V nominal)
"0" state +0.9V or more negative (-0.5V nominal)

Output Data and Flag

"1" state +2.5V or more positive (+2.5V nominal)
"0" state +0.4V or more negative (-0.5V nominal)

Command Interlocks:

1. OPC (Opcode) command, or ENC (Encode) command, may be given to FPP unit ("1") only if FLG (Flag) from FPP is "1". Loads input lines into FPP unit.

2. ENC may not go to "0" until FLG from FPP goes to "0". Acknowledges load.

3. ENC or OPC may not go to "1" again until FLG goes to "1". Signifies FPP operation complete; X Register contents are present on FPP output lines. (Computer requires that the time interval from FLG "0" to FLG "1" be no less than 0.5 microsecond.)

Power Requirements:

FPP Unit

115 volts ac ±10% (approximately 4 amperes), or
230 volts ac ±10% (approximately 2 amperes).
Line frequency, 48 to 66 Hz
Power, 500 watts maximum.

EAU Controller (3 cards)

7.7 amperes from +4.5V supply
5.2 amperes from -2V supply

PHYSICAL SPECIFICATIONS (FPP Unit)

Environmental Limits:

Temperature: 0° to +55°C (+32° to +131°F)
Relative Humidity: To 95% at +40°C (+104°F)

Heat Dissipation: 1700 BTU/hr maximum

Table 1-1. Floating Point Processor Specifications (Continued)

Ventilation: Forced air intake at rear, exhausted through side panels near front. 400 cubic feet per minute.

Weight: 86 lb (39 kg), net

Dimensions:

Width: 16-3/4 inches (425 millimeters); rack mounting kit increases width to 19 inches (482 millimeters)

Panel Height: 12-1/4 inches (311 millimeters)

Depth Behind Panel: 22-3/8 inches (568 millimeters)

Depth Overall (including handles): 24-3/8 inches (619 millimeters)

Recommended Air Exhaust Clearance at Sides: 2 inches (50 millimeters) minimum

Recommended Cable Clearance at Rear: 5 inches (127 millimeters) minimum

Cable Lengths:

Input/Output Cable: 15 feet (457 centimeters)

Power Control Cable: 15 feet (457 centimeters)

AC Power Cable: 88 inches (223 centimeters)

SECTION II

INSTALLATION AND OPERATION

2-1. INTRODUCTION.

2-2. This section contains information on unpacking mounting, connecting, and operating the HP 2152A Floating Point Processor.

2-3. UNPACKING AND INSPECTION.

2-4. If the shipping container is damaged upon receipt, request that the carrier's agent be present when the unit is unpacked. Inspect the unit for damage (scratches, dents, broken parts, etc.). If the unit is damaged and fails to meet specifications, notify the carrier and the nearest Hewlett-Packard Sales and Service Office immediately. (Sales and Service Offices are listed at the back of this manual.) Retain the shipping container and the packing material for the carrier's inspection. The Hewlett-Packard Sales and Service Office will arrange for the repair or replacement of the damaged unit without waiting for any claims against the carrier to be settled.

2-5. MOUNTING.

2-6. The FPP unit is designed for bench installation or mounting in a standard 19-inch rack. To mount the FPP unit in a rack, follow the instructions contained in the rack mounting kit (part no. 5060-0779) furnished with the unit. All necessary hardware is furnished as part of the rack mounting kit.

2-7. INSTALLATION AND CABLING.

2-8. Connecting the FPP unit to the computer consists of two steps: installing the EAU controller cards in the appropriate slots in the computer, and then connecting the interconnection cables. The following paragraphs describe the installation of the EAU controller cards in 2116-series computers (paragraph 2-9) and 2115-series computers (paragraph 2-11), followed by the interconnecting cabling information and a performance check.

2-9. INSTALLING EAU CONTROLLER IN 2116 COMPUTER.

2-10. To install the EAU controller cards in a 2116 series computer, proceed as follows:

a. Press the POWER switch, located on the front panel of the computer, to switch power off.

b. Open the computer front panel to gain access to the card cage.

c. Install the 02152-60011 EAU Logic Card into slot number 110 in the middle rack of cards.

d. Install the 02152-60012 EAU Timing Card into slot number 109 in the middle rack of cards.

e. Install the 02152-60013 EAU Interface Card into any desired I/O slot (203 through 218).

NOTE

The EAU interface card has no interrupt capability and therefore has no priority significance; this card need not be immediately adjacent to the other interface cards (as is normally required for priority continuity). The card simply occupies an I/O slot to have access to the I/O buses. It is normally not addressed, except during diagnostic testing. Also note that this card may not be installed in an I/O extender, as cable limitations require close proximity to the EAU timing card.

f. Proceed to paragraph 2-13.

2-11. INSTALLING EAU CONTROLLER IN 2115 COMPUTER.

2-12. To install the EAU controller cards in a 2115-series computer, proceed as follows:

a. Set the POWER switch, located on the front panel of the computer, to the off position.

b. Remove the computer top panel to gain access to the card cage.

c. Install the 02152-60011 Logic Card into slot number 17 in the front rack of cards.

d. Install the 02152-60012 Timing Card into slot number 16 in the front rack of cards.

e. Install the 02152-60013 EAU Interface Card into any desired I/O slot (114 through 121 in rear rack of cards). Refer to the note regarding priority, addressing, and use of extenders in paragraph 2-10.

f. Proceed to paragraph 2-13.

2-13. CONNECTING CABLES.

2-14. There are three cables to be connected for normal operation. (See figure 1-1.) The fourth

cable supplied (a short extender cable) is used for test purposes only.

2-15. Figure 2-1 shows the cabling interconnections for both the 2116- and 2115-series computers. Note that the input/output cable (02152-60022) has two connectors on one end and a single connector on the other end. Connect the single-connector end, as shown, to the top edge connector of the FPP interface card in the FPP unit. Then connect the other end to the edge connectors of both the EAU timing and interface cards as shown.

2-16. Connect the power control cable from J2 on the rear panel of the FPP unit to the power control connector of the computer (J2 on a 2116-series computer or J4 on the power supply of a 2115-series computer). If there are other extenders in the system, such as I/O extenders, connect the power control cable from any such extenders to J3 on the rear panel of the FPP unit. (Internally, J2 and J3 are directly jumpered, pin to pin.)

2-17. Before connecting the ac power cable, ensure that the 115-/230-volt switch and the fuse value are correct for the power source to be used. The access procedure is as follows:

a. Slide the clear plastic cover of the ac input power module to the left. (The cover cannot move unless the power cable is removed.)

b. Remove the fuse by pulling on the tab marked FUSE FULL.

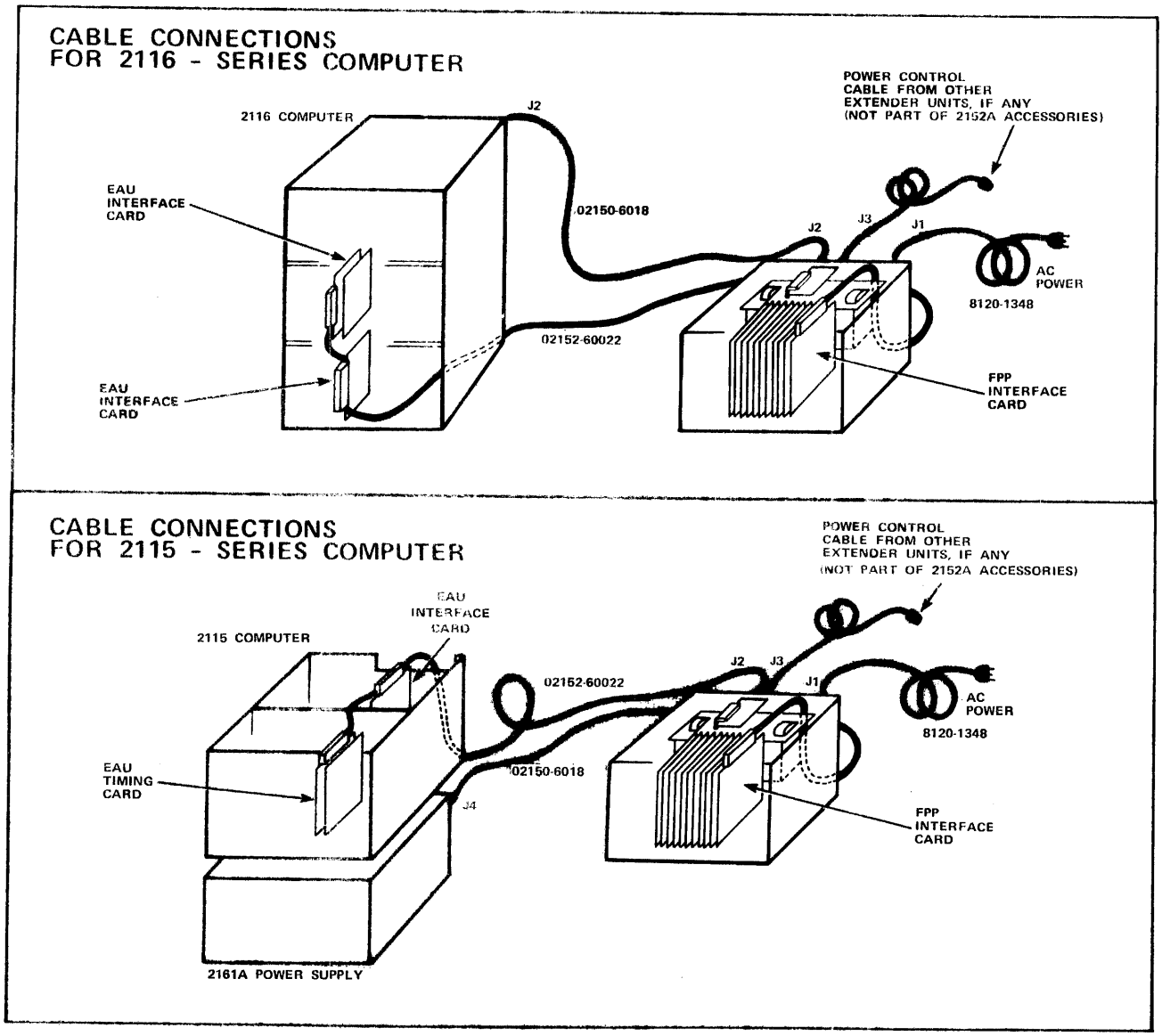


Figure 2-1. Interconnecting Cabling

c. If necessary, slide the arrow indicator to the correct voltage setting for the power source (115 or 230 volts ac). (The indicator cannot move unless the fuse is removed.)

d. Insert the correctly rated fuse for the line voltage (6 amperes for 115V, or 3 amperes for 230V).

e. Slide the clear plastic cover back to the right.

2-18. Now connect the ac power cable (8020-1348) to the ac power receptable of the ac input power module and plug the cable into the power source.

2-19. PERFORMANCE CHECK.

2-20. Following completion of the installation procedure, performance should be checked by running the diagnostic tests as outlined in the Manual of Diagnostics. If the test portion of the diagnostic program is completed without error, the installation is complete.

2-21. OPERATION.

2-22. POWER ON AND POWER FAIL.

2-23. To place the HP 2152A Floating Point Processor in operation, switch on power at both the FPP unit and the computer. The power fail interlocks require that both units be on before either can become operational.

2-24. In the event of a power failure in either the FPP unit or the computer, the power fail logic in the computer is activated. In most cases, depending on computer model and type of power fail option installed, this causes an interrupt to a location where the entry of a power fail subroutine is stored. The subroutine must be written so as to include saving the contents of the FPP X-register in addition to saving the contents of the computer registers. Use the STX (Store X) instruction to save the contents; in a subsequent restart (if the computer has this optional feature), restore the X-register contents by using a LDX (Load X) instruction, referencing the same location used for STX.

2-25. OPERATING PROCEDURE.

2-26. There are no further operating procedures or precautions. Once placed in operation, the HP 2152A Floating Point Processor becomes an extension of the computer CPU. The net effect is to add 40 instructions to the basic repertoire of the computer. Section III defines these instructions and discusses some considerations in their usage.

Table 3-1. Instruction Coding

		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIPLE LOAD AND STORE	LDD	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	0
	LDF	1	0	0	0	1	0	1	1	0	0	1	0	1	0	0	0
	LDX	1	0	0	0	1	0	1	1	0	0	1	0	1	0	0	1
	STD	1	0	0	0	0	0	1	1	0	0	1	0	0	1	1	0
	STF	1	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0
	STX	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1
TRIPLE PRECISION FLOATING POINT ARITHMETIC	ADF	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
	ADX	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1
	SBF	1	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0
	SBX	1	0	0	0	1	0	1	1	0	0	0	0	0	0	1	1
	MPF	1	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0
	MPX	1	0	0	0	1	0	1	1	0	0	0	0	0	1	0	1
	DVF	1	0	0	0	1	0	1	1	0	0	0	0	0	1	1	0
	DVX	1	0	0	0	1	0	1	1	0	0	0	0	0	1	1	1
FUNCTIONS	ABX	1	0	0	0	1	0	1	0	0	0	1	0	0	0	1	1
	ENX	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	1
	CMX	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1
	CSX	1	0	0	0	1	0	1	0	0	0	0	1	0	0	1	1
	SNX	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	1
	TNX	1	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1
	ATX	1	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1
	HCX	1	0	0	0	1	0	1	0	0	0	0	1	1	0	1	1
	HSX	1	0	0	0	1	0	1	0	0	0	0	1	1	0	0	1
	HTX	1	0	0	0	1	0	1	0	0	0	0	1	1	0	1	1
	AHT	1	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1
	EXX	1	0	0	0	1	0	1	0	0	0	0	1	1	1	1	1
	LNX	1	0	0	0	1	0	1	0	0	0	0	0	1	1	1	1
	SRX	1	0	0	0	1	0	1	0	0	0	0	1	0	1	1	1
	FIX	1	0	0	0	1	0	1	0	0	0	0	1	0	0	1	1
	RNX	1	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1
DOUBLE PRECISION INTEGER ARITHMETIC (A-/B-Registers)	MPY	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	DIV	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
DOUBLE WORD LOAD/STORE/SHIFT (A-/B-Registers)	DLD	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
	DST	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
	ASR	1	0	0	0	0	0	1	0	0	0	0	1		*n		
	ASL	1	0	0	0	0	0	0	0	0	0	0	1		*n		
	LSR	1	0	0	0	0	0	1	0	0	0	1	0		*n		
	LSL	1	0	0	0	0	0	0	0	0	0	1	0		*n		
	RRR	1	0	0	0	0	0	1	0	0	1	0	0		*n		
	RRL	1	0	0	0	0	0	0	0	0	1	0	0		*n		

*n = number of shifts in binary coded decimal (0 = 16 shifts)

SECTION III

PROGRAMMING INFORMATION

3-1. INTRODUCTION.

3-2. This section provides machine language programming information for the floating point processor. Specifically, this includes instruction coding and instruction definitions, plus an explanation of data formats and error codes. For software documentation refer to separate programming manuals furnished with the FPP software.

3-3. INSTRUCTION CODING.

3-4. Table 3-1 lists the machine codes for each of the 40 FPP instructions. The arithmetic and load/store instructions require an operand and therefore use two words of memory. The first word is the 16-bit instruction code, and the second word is the 15-bit address reference (plus an indirect address bit). The remaining instructions, which are shift and function instructions, require no operand and therefore use only one word in memory.

3-5. DATA FORMATS.

3-6. Various instructions in the FPP repertoire use one of three possible data formats, and in some cases perform conversions from one format to another. These characteristics are given in the instruction definitions, beginning at paragraph 3-12. The following paragraphs compare the three data formats, and describe the conversions between double and triple precision floating point formats. See figure 3-1.

3-7. DOUBLE WORD INTEGER. As shown in the upper box of figure 3-1, the double word integer format represents only whole numbers (binary point assumed at the right of the value), using 31 bits for the value and one bit for the sign. This format uses two computer words, and permits an equivalent decimal range of numbers from -2,147,483,648 through +2,147,483,647. When stored in memory, the most significant data word is stored in the location addressed by the operand word of an FPP double-word instruction (either directly or indirectly), and the least significant data word is stored in the next higher memory location. When represented in the B- and A-registers of the computer, the B-register contains the most significant data word, and the A-register contains the least significant data word.

3-8. FLOATING POINT. Like double word integer, floating point numbers require two computer words, and when stored in memory the most significant word occupies the first addressed location (or the B-register when loaded into the registers). However, the mantissa is assumed to be a fractional value (binary point at the left), and has only 23 bits of significance. This is equivalent to a decimal range of -8,388,608 through +8,388,607. The 7-bit exponent extends the range by effectively moving the binary point right or left; the range of exponent powers, in decimal, is -128 through +127. Note that the second word of the floating point format is split, such that bits 8 through 15 are the eight least significant bits of the mantissa, and the remaining eight bits are the exponent and exponent sign.

3-9. EXTENDED FLOATING POINT. The only difference between extended and double-word floating point formats is the addition of one computer word (16 bits) to the length of the mantissa. Since the computer registers cannot contain this triple-length word, it is representable only in memory (3 locations used) or in the FPP unit (which has 48-bit registers).

3-10. FORMAT CONVERSIONS. As shown in the lower box of figure 3-1, the conversion from double- to triple-length format consists of splitting the second word of the double-word format between bits 8 and 7, and inserting 16 zeros as the least significant bits of the mantissa in the triple-word format. Note that 8 of these zeros are present in the second word of the triple-word format, and the remaining 8 are in the third word. The reverse conversion, from triple- to double-length format, consists simply of truncating the 16 least significant bits of the mantissa. This means removing bits 7 through 0 of the second word, and bits 15 through 8 of the third word.

3-11. The conversions between integer and extended floating point formats (not illustrated) are more complex. Briefly, the process is as follows. For conversion to integer, the mantissa is arithmetically shifted right while the exponent value is correspondingly increased (one increment per shift) until the exponent equals +31. If bit 15 is a "1" (implying 1/2), the quantity in bits 16 through 47 is incremented by 1; this rounds the integer to the nearest whole number. Bits 8 through 15 are set to "0", and bits 16 through 47 comprise the integer value. The reverse conversion, integer to extended floating point, consists of filling in zeros for bits 8 through 15, setting the exponent to +31, and then normalizing the result. (Normalization is discussed later, in paragraph 3-99.)

3-12. INSTRUCTION DEFINITIONS.

3-13. From a programming standpoint, the addition of the HP 2152A Floating point Processor to the computer effectively adds a third accumulator, designated the X-register. This register can be loaded, manipulated, and its contents stored in memory in the same way as the A- and B-registers in the computer. The following instruction definitions make this assumption.

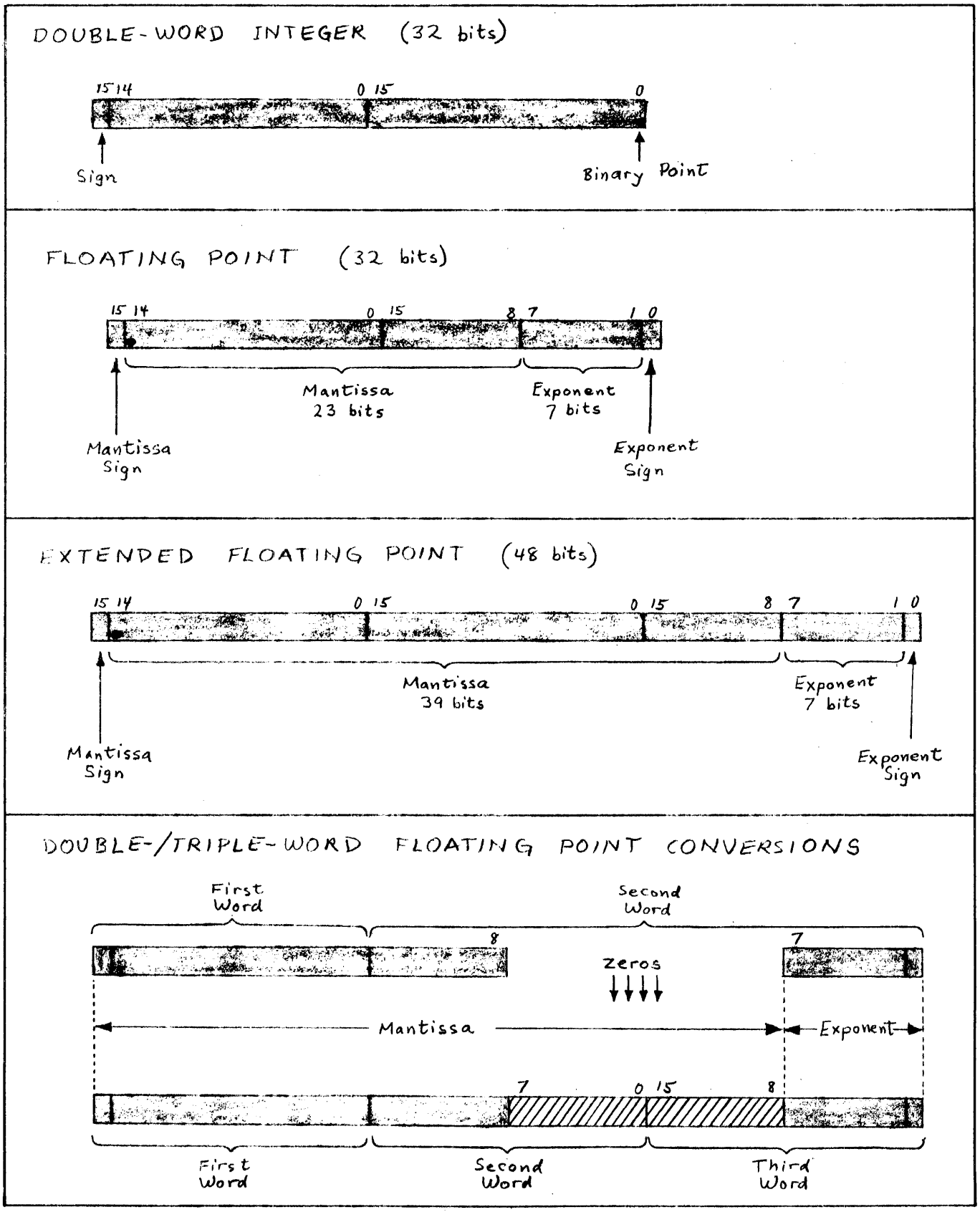


Figure 3-1. FPP Data Formats

3-14. LDD (LOAD DOUBLE).

3-15. Load the double-word integer from addressed memory location m (and m + 1) into the X-register and convert it into an extended floating point number. The extended floating point result occupies the X-register on completion of the instruction.

3-16. LDF (LOAD FLOATING).

3-17. Load the floating point number from addressed memory location m (and M + 1) into the X-register and convert it into an extended floating point number.

3-18. LDX (LOAD EXTENDED).

3-19. Load the extended floating point number from addressed memory location m (and m + 1 and m + 2) into the X-register.

3-20. STD (STORE DOUBLE).

3-21. DEFINITION. Convert the extended floating point number in the X-register into a double-word integer and store in addressed memory location m (and m + 1). The integer value occupies the 32 most significant bit positions of the X-register on completion of the instruction, and bits 1 through 5 of the exponent field are set to "1" (decimal 31).

3-22. COMMENTS. The conversions can cause an overflow condition (error code 2, overflow set); refer to paragraph 3-112. Setting the exponent field of the X-register to 31 permits any succeeding instruction to use the X-register contents as a floating point quantity. (If the result is not a normalized quantity, it will automatically be normalized as part of the execution sequence of the succeeding instruction.)

3-23. STF (STORE FLOATING).

3-24. Convert the extended floating point number in the X-register into a floating point number and store in addressed memory location m (and m + 1). The conversion is accomplished by truncating the mantissa to 24 bits. The original extended floating point value in the X-register is not changed.

3-25. STX (STORE EXTENDED).

3-26. Store the extended floating point number in the X-register in addressed memory location in (and m + 1 and m + 2).

3-27. ADF (ADD FLOATING).

3-28. Load the floating point number from addressed memory location m (and m + 1) and convert it to an extended floating point number; then add to the current value in the X-register. The extended floating point result of the addition occupies the X-register on completion of the instruction. Overflow or underflow error conditions can result; refer to paragraphs 3-110 and 3-113.

3-29. ADX (ADD EXTENDED).

3-30. Add the extended floating point number from addressed memory location m (and m + 1 and m + 2) to the current value in the X-register. The extended floating point result of the addition occupies the X-register on completion of the instruction. Overflow or underflow error conditions can result; refer to paragraphs 3-110 and 3-113.

3-31. SBF (SUBTRACT FLOATING).

3-32. DEFINITION. Load the floating point number from addressed memory location m (and m + 1) and convert it to an extended floating point number; then subtract from the current value in the X-register. The extended floating point result of the subtraction occupies the X-register on completion of the instruction.

3-33. COMMENTS. Overflow or underflow error conditions can result either from the subtraction or from initial values of the X-register. Refer to paragraphs 3-110, 3-111, and 3-113.

3-34. SBX (SUBTRACT EXTENDED).

3-35. Subtract the extended floating point number in addressed memory location m (and m + 1 and m + 2) from the current value in the X-register. The extended floating point result of the subtraction occupies the X-register on completion of the instruction. Overflow or underflow error conditions can result either from the subtraction or from initial values of the X-register; refer to paragraphs 3-110, 3-111, and 3-113.

3-36. MPF (MULTIPLY FLOATING).

3-37. Multiply the extended floating point number in the X-register by the floating point number in addressed memory location m (and m + 1). The extended floating point result of the multiplication occupies the X-register on completion of the instruction. Overflow or underflow error conditions can result; refer to paragraphs 3-110 and 3-113.

3-38. MPX (MULTIPLY EXTENDED).

3-39. Multiply the extended floating point number in the X-register by the extended floating point number in addressed memory location m (and m + 1 and m + 2). The extended floating point result of the multiplication occupies the X-register on completion of the instruction. Overflow or underflow error conditions can result; refer to paragraphs 3-110 and 3-113.

3-40. DVF (DIVIDE FLOATING).

3-41. Divide the extended floating point number in the X-register by the floating point number in addressed memory location m (and m + 1). The extended floating point result of the division occupies the X-register on completion of the instruction. Error conditions due to overflow, underflow, or divide-by-zero attempts can result; refer to paragraphs 3-110, 3-113, and 3-120.

3-42. DVX (DIVIDE EXTENDED).

3-43. Divide the extended floating point number in the X-register by the extended floating point number in addressed memory location m (and m + 1 and m + 2). The extended floating point result of the division occupies the X-register on completion of the instruction. Error conditions due to overflow, underflow, or divide-by-zero attempts can result; refer to paragraphs 3-110, 3-113, and 3-120.

3-44. ABX (ABSOLUTE).

3-45. Calculate the absolute value of the extended floating point value in the X-register; i.e., if the content of the X-register is negative, convert to positive. Overflow can result if the negative number is the maximum negative value; refer to paragraph 3-114.

3-46. ENX (ENTIER).

3-47. Calculate the entier of the extended floating point value in the X-register. The calculated result replaces the original contents of the X-register. Refer to paragraph 3-125 for comparison of entier, RNX and FIX instructions.

3-48. CMX (COMPLEMENT).

3-49. Convert the extended floating point value in the X-register to its two's complement. Overflow or underflow error conditions can result; refer to paragraphs 3-111 and 3-114.

3-50. CSX (COSINE).

3-51. Calculate the cosine of the value in the X-register, where the value is expressed in radians as an extended floating point number. The result of the cosine calculation replaces the original value in the X-register. Excessive complete rotations of the expressed angle can result in a no-resolution error condition; refer to paragraph 3-118.

3-52. SNX (SINE).

3-53. Calculate the sine of the value in the X-register, where the value is expressed in radians as an extended floating point number. The result of the sine calculation replaces the original value in the X-register. Excessive complete rotations of the expressed angle can result in a no-resolution error condition; refer to paragraph 3-118.

3-54. TNX (TANGENT).

3-55. Calculate the tangent of the value in the X-register, where the value is expressed in radians as an extended floating point number. The result of the tangent calculation replaces the original value in the X-register. Error conditions can result from excessive complete rotations of the expressed angle (no resolution) or from attempts to calculate the tangent of angles equal to odd values of π; refer to paragraphs 3-118 and 3-120.

3-56. ATX (ARCTANGENT).

3-57. Calculate the arctangent of the value in the X-register, where the result is expressed in radians as an extended floating point number in the X-register.

3-58. HCX (HYPERBOLIC COSINE).

3-59. Calculate the hyperbolic cosine of the extended floating point number in the X-register. The result of the hyperbolic cosine calculation replaces the original value in the X-register. Overflow or underflow error conditions can result; refer to paragraphs 3-111 and 3-116.

3-60. HSX (HYPERBOLIC SINE).

3-61. Calculate the hyperbolic sine of the extended floating point number in the X-register. The result of the hyperbolic sine calculation replaces the original value in the X-register. Overflow or underflow error conditions can result; refer to paragraphs 3-111 and 3-116.

3-62. HTX (HYPERBOLIC TANGENT).

3-63. Calculate the hyperbolic tangent of the extended floating point number in the X-register. The result of the hyperbolic tangent calculation replaces the original value in the X-register.

3-64. AHT (ARCHYPERBOLIC TANGENT).

3-65. Calculate the hyperbolic arctangent of the extended floating point number in the X-register. The result of the hyperbolic arctangent calculation replaces the original value in the X-register. An improper-variable error condition results if the argument value is equal to or greater than 1; refer to paragraph 3-122.

3-66. EXX (EXPONENTIAL).

3-67. Calculate the exponential of the extended floating point number in the X-register. The result

of the exponential calculation replaces the original value in the X-register. Overflow or underflow error conditions can result; refer to paragraphs 3-111 and 3-116.

3-68. LNX (NATURAL LOGARITHM).

3-69. Calculate the natural logarithm of the extended floating point number in the X-register. The result of the logarithm calculation replaces the original value in the X-register. An improper-variable error condition results if the argument value is zero or a negative number; refer to paragraph 3-122.

3-70. SRX (SQUARE ROOT).

3-71. Calculate the square root of the extended floating point number in the X-register. The result of the square root calculation replaces the original value in the X-register. An improper-variable error condition results if the argument value is a negative number; refer to paragraph 3-122.

3-72. FIX (ROUND TO NEAREST INTEGER).

3-73. Round-off the extended floating point number in the X-register to the nearest integer value. The result remains an extended floating point number and replaces the original value in the X-register. The number of bits affected depends on the exponent value. A comparison of FIX, RNX, and ENX instructions is given in paragraph 3-125.

3-74. RNX (ROUND TO 24 BITS).

3-75. Round-off the extended floating point number in the X-register to 24 bits of precision. An overflow error condition can result if the maximum positive number is rounded-off in the positive direction; refer to paragraph 3-115. A comparison of RNX, FIX, and ENX instructions is given in paragraph 3-125.

3-76. MPY (MULTIPLY INTEGER).

3-77. Multiply the 16-bit integer value in the computer A-register by the 16-bit integer value in addressed memory location m. The result is a double-word integer occupying the computer B- and A-registers, with the B-register containing the sign bit and most significant 15 bits of the quantity.

3-78. DIV (DIVIDE INTEGER).

3-79. Divide the double-word integer in the combined B- and A-registers of the computer by the 16-bit integer value in addressed memory location m. The result is a 16-bit integer quotient in the A-register and a 16-bit integer remainder in the B-register. An overflow condition can result from an attempt to divide by zero or from a dividend which is too large for the divisor. Refer to paragraph 3-106.

3-80. DLD (DOUBLE LOAD).

3-81. Load the contents of addressed memory location m (and m + 1) into the computer A- and B-registers, respectively.

3-82. DST (DOUBLE STORE).

3-83. Store the double-word quantity in the computer A- and B-registers into addressed memory location m (and m + 1).

3-84. ASR (ARITHMETIC SHIFT RIGHT).

3-85. Arithmetically shift the combined contents of the computer B- and A-registers right, n places. The value of n may be any number from 1 through 16. The sign bit is unchanged, and is extended into bit positions vacated by the right shift. Data bits shifted out of the least significant end of the A-register are lost. See ASR example in figure 3-2.

3-86. ASL (ARITHMETIC SHIFT LEFT).

3-87. Arithmetically shift the combined contents of the computer B- and A-registers left, n places. The value of n may be any number from 1 through 16. Zeros are filled into vacated low order positions of the A-register. The sign bit is unchanged, and data bits are lost out of bit 14 of the B-register. If one of the bits lost is a significant data bit ("1" for positive numbers, "0" for negative numbers), overflow will be set; otherwise, overflow will be cleared during execution. See ASL example in figure 3-2. Note that two additional shifts in this example would cause an error by losing a significant "1".

3-88. LSR (LOGICAL SHIFT RIGHT).

3-89. Logically shift the combined contents of the B- and A-registers right, n places. The value of n may be any number from 1 through 16. Zeros are filled into vacated high order bit positions of the B-register, and data bits are lost out of the low order bit positions of the A-register. See LSR example in figure 3-2.

3-90. LSL (LOGICAL SHIFT LEFT).

3-91. Logically shift the combined contents of the B- and A-registers left, n places. The value of n may be any number from 1 through 16. Zeros are filled into vacated low order bit positions of the A-register, and data bits are lost out of the high order bit positions of the B-register. See LSL example in figure 3-2.

3-92. RRR (ROTATE RIGHT).

3-93. Rotate the combined contents of the B- and A-registers right, n places. The value of n may be

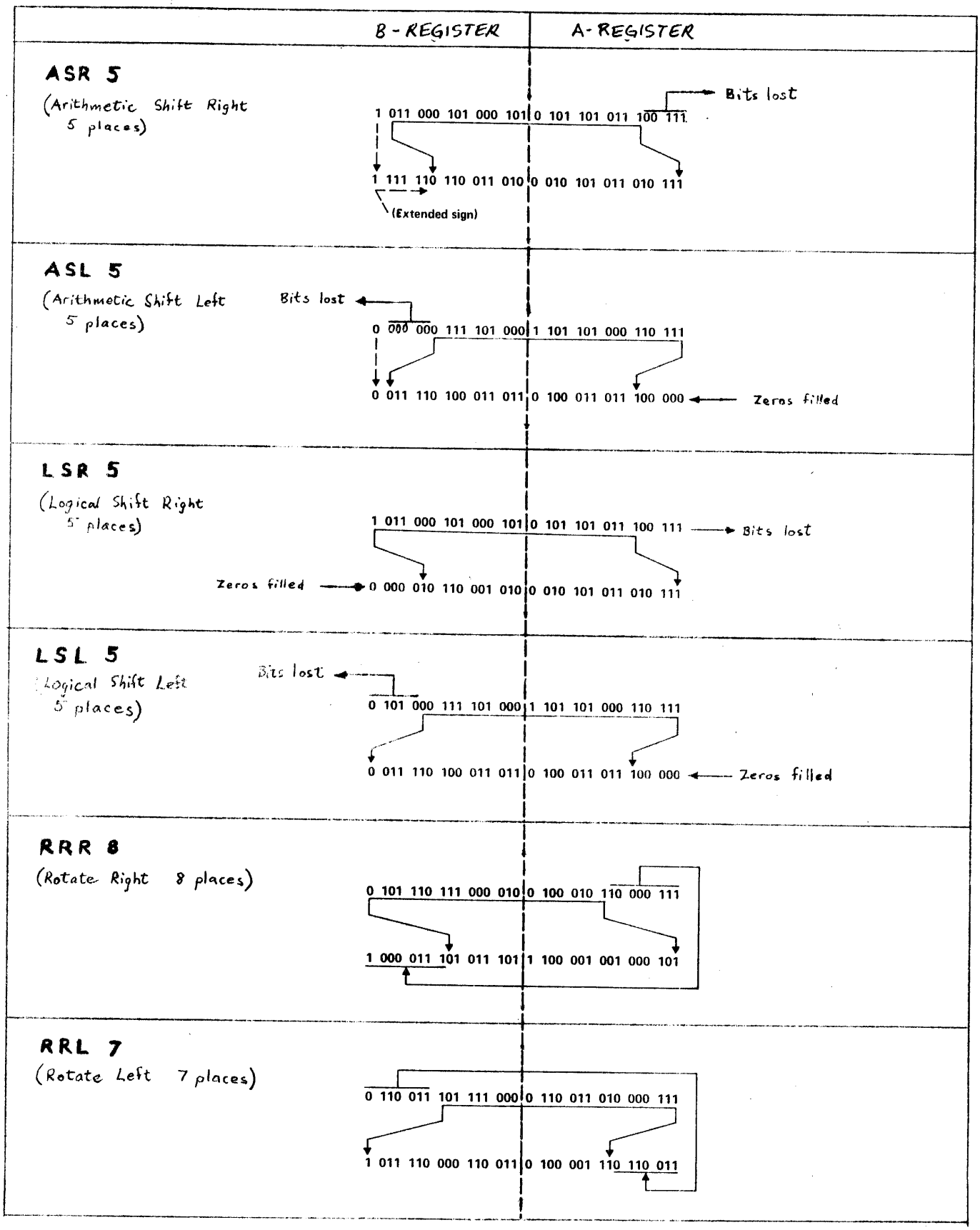


Figure 3-2. Examples of Shifts and Rotates

any number from 1 through 16. No bits are lost or filled in. Data bits shifted out of the low order end of the A-register are rotated around to enter the high order end of the B-register. See RRR example in figure 3-2.

3-94. RRL (ROTATE LEFT).

3-95. Rotate the combined contents of the B- and A-registers left, n places. The value of n may be any number from 1 through 16. No bits are lost or filled in. Data bits shifted out of the high order end of the B-register are rotated around to enter the low order end of the A-register.

3-96. EXTENDED FLOATING POINT NUMBERS.

3-97. Internally, the FPP processes all data as normalized extended floating point numbers. The FPP instructions will convert the input data if the data is not already in this form.

3-98. This requirement of using only normalized numbers means that a number can be too small for processing, as well as too large. Figure 3-3 illustrates this case. The example uses positive decimal numbers, and is equivalent to the binary magnitudes representable in the FPP unit.

3-99. As shown in figure 3-3, normalization consists of shifting the mantissa left to eliminate any zeros between the decimal point and the first non-zero digit, while the exponent is correspondingly reduced by subtracting 1 for each shift. In the upper example in figure 3-3, there are two zeros between the decimal point and the digit 8. Therefore two left shifts are necessary, and the exponent is reduced from -31 to -33. In the lower example, the number is too small to be normalized, resulting in an underflow condition. The mantissa is shown shifted left seven positions, which reduces the exponent to its smallest possible value, -38. (This is equivalent to the smallest binary exponent of -128.) No further left shifts can therefore be made, and there is still one zero left between the decimal point and the digit 8.

NORMALIZATION

Mantissa	Exponent
+ .008000000000	-31 (10^{-31})
+ .800000000000	-33 (10^{-33})

UNDERFLOW
(Cannot be normalized)

Mantissa	Exponent
+ .000000008000	-31 (10^{-31})
+ .080000000000	-38 (10^{-38})

Figure 3-3. Normalization, Decimal Example

3-100. Figure 3-4 defines the ranges of valid binary numbers that can be processed by the FPP unit. The shaded areas define overflow and underflow ranges. In the VALUE column, the mantissa is enclosed in parentheses, followed by the exponent outside of the parentheses.

3-101. Note that 0 (middle line of the figure) is represented by all zeros in both the mantissa and the exponent. Positive numbers are shown above this line, and negative numbers are shown below this line. In the far right column, the exponent values are shown increasing in both directions from the zero line, from the smallest representable value (-128) to the largest (+127). Nonrepresentable exponent values, smaller than -128, are also underflow conditions, but this situation is not considered in the figure.

3-102. For each finite value of exponent, the mantissa is assumed to go through its complete cycle of valid values. In approximate terms, the positive mantissa cycles from +1/2 to +1, and the negative mantissa cycles from -1/2 to -1. More precisely, the positive range is from +1/2 to the largest possible fractional number under the value of 1. The negative range is from the largest possible fractional number below (more negative than) -1/2 to -1.

3-103. The significance of +1/2 and -1/2 in determining mantissa ranges is a result of the normalization requirement, which dictates that there will be a significant digit immediately to the right of the binary point. This automatically eliminates all numbers between +1/2 and -1/2, including the exact value of -1/2, but excluding 0 and +1/2.

3-104. Figure 3-4 can also be applied to double-length floating point numbers, which may be used externally to the FPP unit. The only difference is that, since there are 16 fewer bits of mantissa, the three appearances of -39 as a power of 2 when expressing the VALUE of the mantissa would be changed to -23.

3-105. ERROR CONDITIONS.

3-106. After execution of most FPP instructions, overflow should be checked for a possible execution error. Those instructions which can result in an error will set the overflow bit, and most of these will load an error code into bits 10, 9, and 8 of the computer A-register. These three bits are read as an octal digit to identify the code, as listed in table 3-2. Table 3-2 lists the possible types of error that can occur for each of the 40 instructions. (A table of error codes is also given in section V.) The following paragraphs discuss the various types of errors.

3-107. NO RESPONSE.

3-108. An error code of 0 in the A-register following an affirmative overflow error check indicates that the FPP unit is not returning Flag signals and is

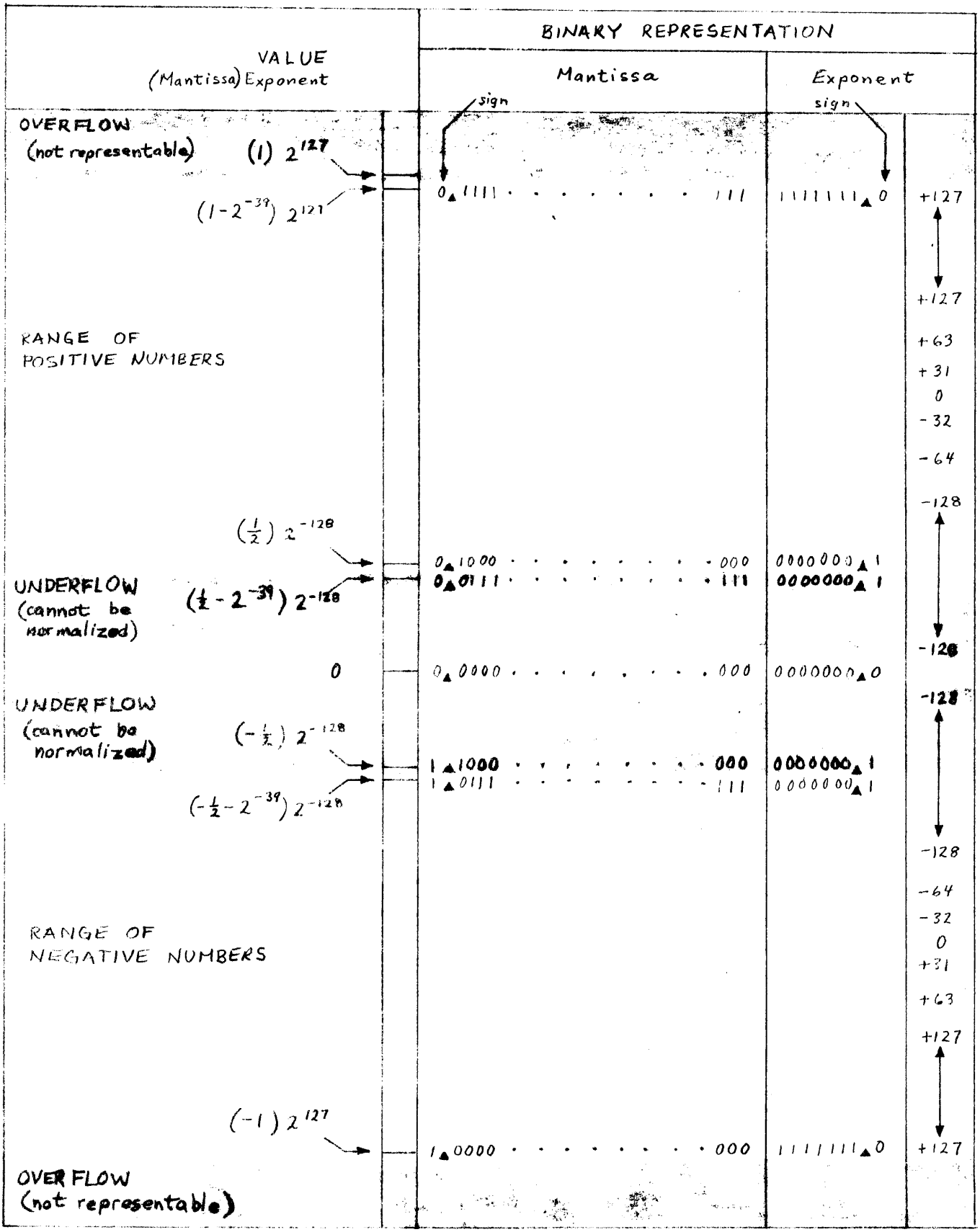


Figure 3-4. Ranges of Extended Floating Point Numbers

therefore disabled or not connected. Since the FPP unit is not used in the execution of the last ten instructions listed in table 3-2, response checks cannot be programmed for these instructions.

3-109. UNDERFLOW.

3-110. The floating point add, subtract, multiply, and divide instructions will produce an underflow error indication if the result can not be normalized; i.e., the result falls in one of the shaded areas in figure 3-4 immediately adjacent to zero. If the input data is not normalized, the instruction will normalize the operands before beginning the computation; if the numbers are too small to be successfully normalized, the instruction will attempt normalization as far as possible and then proceed with the computation. The answer will be correct except that the sign bit of the exponent will be incorrect (complemented).

3-111. The CMX, HCX, HSX, and EXX instructions check the pre-execution contents of the X-register for a value which, when altered by the instruction, would produce a result which cannot be normalized. Similarly, the subtract instructions (SBF and SBX) check if subtrahends from memory are at the minimum positive value; this would produce an underflow when converted to negative form during execution. However, as in all underflow conditions, the computation will proceed, and only the sign bit of the exponent will be incorrect.

3-112. OVERFLOW.

3-113. The floating point add, subtract, multiply, and divide instructions will produce an overflow error indication if the result exceeds the largest positive or negative number which can be represented; i.e., the result falls in either the top or bottom shaded areas in figure 3-4. Answers will be correct except for the sign bit of the exponent (will be complemented). However, the divide instructions, DVF and DVX, can produce one exception to this general rule: if the dividend is +126 or +127 and the divisor is -127 or -128, the resultant exponent will be incorrect. In this case, the exponent value will equal the original value of the dividend, plus two. This produces a rollover to an apparent negative exponent of -128 (if the original was +126) or -127 (if the original was +127).

3-114. The subtract instructions (SBF, SBX) also check the subtrahend from memory before execution begins. If the subtrahend is at the maximum negative value, an overflow will result when the number is converted to positive form during execution. Similarly, the CMX and ABX instructions check the pre-execution contents of the X-register for maximum negative value; overflow will result when the number is converted. However, in all four cases (SBF, SBX, CMX, ABX) execution will proceed, and only the sign bit of the exponent will be incorrect (complemented).

3-115. The RNX instruction will produce an overflow if the number is at the maximum positive value and then is rounded upward. This will cause rollover to the maximum negative number.

3-116. Overflows resulting from STD, FIX, HCX, HSX, and EXX produce results which are generally unpredictable, making it difficult or impossible to reconstruct correct answers.

3-117. NO RESOLUTION.

3-118. The CSX, SNX, and TNX instructions allow expressed arguments to include complete rotations of the angle. These whole rotations use up part of the available floating point bits, leaving fewer bits to express the fractional part of the rotation for the computation. When the number of whole rotations reaches about $2^{38}/2\pi$, there is not sufficient resolution left to express angles in increments smaller than 90 degrees. The no-resolution error code indicates this condition.

3-119. DIVIDE BY ZERO.

3-120. If the divisor for DVF or DVX instructions is zero, the division will not be attempted, and the error code of 4 will indicate this condition. Also, in the TNX computation of tangent, odd numbers of quarter rotations (1/4, 3/4, etc.) would result in a divide-by-zero condition (sin = 1, cos = 0). This is also indicated by an error code of 4. The sine value (1) will remain in the X-register on exit from the instruction.

3-121. IMPROPER VARIABLE.

3-122. Attempts to calculate the square root (SRX) of negative numbers, or the natural logarithm (LNX) of zero or negative numbers, will not be executed, and will leave the X-register unchanged. Attempts to calculate the archyperbolic tangent (AHT) of numbers which are ±1 or greater will also not be executed, and will usually leave the X-register unchanged; the exception is that the attempt to calculate archyperbolic tangent of -1 will result in clearing the X-register to zero.

3-123. IMPROPER OPCODE.

3-124. Undefined opcodes given to the FPP unit will cause the instruction to be ignored, and will indicate rejection by an error code of 6.

3-125. COMPARISON OF ENX, RNX, FIX.

3-126. The ENX, RNX, and FIX instructions all alter the representation form of an extended floating point number, without changing its value. The differences in effects are illustrated in figure 3-5 and are explained in the following paragraphs.

Table 3-2. Error Conditions

INSTR	POSSIBLE ERRORS CODE	POSSIBLE ERRORS ERROR	CONDITION	ERROR EFFECT ON X-REGISTER CONTENTS
LDD		None		
LDF		None		
LDX		None		
STD	2	Overflow	$x < -2^{31}$, or $x \geq 2^{31}$	Not predictable Not predictable
STF		None		
STX		None		
ADF	1	Underflow	See note 1	Correct, except exponent sign bit is complemented
	2	Overflow	See note 2	Same as underflow effect
ADX	1	Underflow	See note 1	Same as above
	2	Overflow	See note 2	Same as above
SBF	1	Underflow	See note 1, or $x = (1/2)\ 2^{-128}$	Same as above
	2	Overflow	See note 2, or $x = -2^{-128}$	Same as above
SBX	1	Underflow	See note 1, or $x = (1/2)\ 2^{-128}$	Same as above
	2	Overflow	See note 2, or $x = -2^{128}$	Same as above Same as above
MPF	1	Underflow	See note 1	Same as above
	2	Overflow	See note 2	Same as above
MPX	1	Underflow	See note 1	Same as above
	2	Overflow	See note 2	Same as above
DVF	1	Underflow	See note 1	Same as above
	2	Overflow	See note 2	See paragraph 3-113
	4	Divide by Zero	Divisor = 0	Unchaged
DVX	1	Underflow	See note 1	Same as ADF underflow
	2	Overflow	See note 2	See paragraph 3-113
	4	Divide by Zero	Divisor = 0	Unchanged
ABX	2	Overflow	$x = -2^{127}$	Same as ADF underflow
ENX		None		
CMX	1	Underflow	$x = (1/2)\ 2^{-128}$	Same as ADF underflow
	2	Overflow	$x = -2^{127}$	Same as ADF underflow
CSX	3	No Resolution	$\|x\| = \geq 2^{38}$	Unchanged
SNX	3	No Resolution	$\|x\| = \geq 2^{38}$	Unchanged
TNX	3	No Resolution	$\|x\| = \geq 2^{38}$	Unchanged
	4	Divide by Zero	$x = (2k + 1)\ \pi/2$ for k=0, ±1, ±2...	Sine remains in X (= +1 or -1)
ATX		None		
HCX	1	Underflow	$x < -88$	Same as ADF underflow
	2	Overflow	$x > 88$	Not predictable

Table 3-2. Error Conditions (Continued)

INSTR	POSSIBLE ERRORS CODE	POSSIBLE ERRORS ERROR	CONDITION	ERROR EFFECT ON X-REGISTER CONTENTS
HSX	1	Underflow	$x < -88$	Same as ADF underflow
	2	Overflow	$x > 88$	Not predictable
HTX		None		
AHT	5	Improper Variable	$\|x\| \geq 1$	Unchanged, unless x = -1; then contents will = 0.
EXX	1	Underflow	$x < -87.3$	Same as ADF underflow
	2	Overflow	$x > 87.3$	Not predictable
LNX	5	Improper Variable	$x \leq 0$	Unchanged
SRX	5	Improper Variable	$x < 0$	Unchanged
FIX	2	Overflow	$x < -2^{31}$, or	Not predictable
			$x \geq 2^{31}$	Not predictable
RNX	2	Overflow	$z = 2^{127}$	Same as ADF underflow

INSTR	POSSIBLE ERRORS	CONDITION	EFFECT ON A-/B-REGISTERS
MPY	None		
DIV	Overflow	Dividend/Divisor $\geq 2^{31}$,	Not predictable
		or $< 2^{31}$,	Not predictable
	Divide by Zero	Divisor = 0	Unchanged
DLD	None		
DST	None		
ASR	None		
ASL	Arithmetic Overflow		Significant bits lost
LSR	None		
LSL	None		
RRR	None		
RRL	None		

Notes:

1. Underflow result is in the range $0 < z < (1/2)2^{-128}$ or $0 > z \geq (-1/2)2^{-128}$

2. Overflow result is $\geq 2^{127}$ or $< -2^{127}$

3. All instructions except last 10 (MPY through RRL) can also produce a no-response error condition, implying that the FPP unit does not respond to the instruction. Error code = 0.

4. x = Contents of X-register before execution.
   z = Contents of X-register after execution.

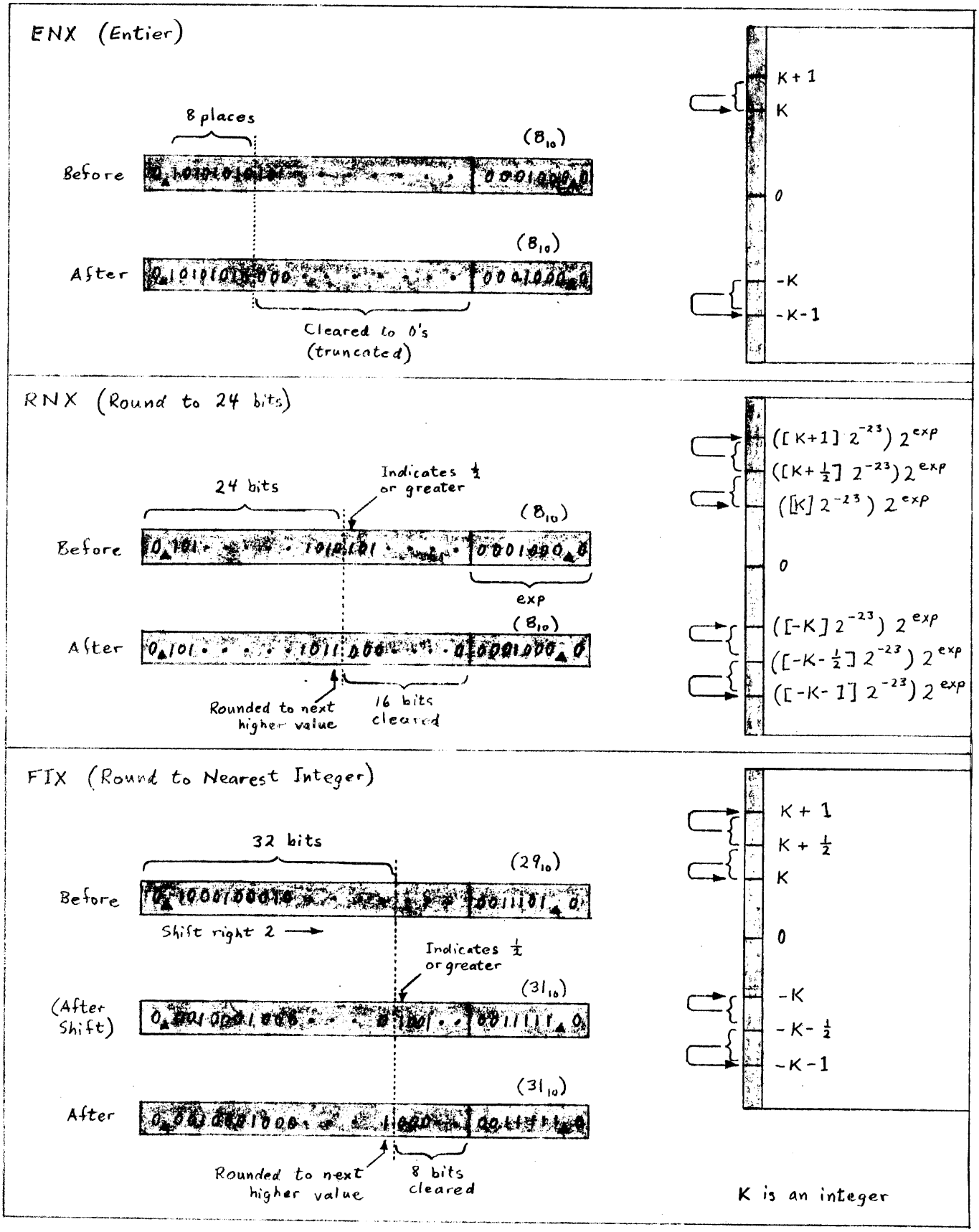


Figure 3-5. Examples of ENX, RNX, and FIX

3-127. ENX. Entier is simple truncation of the fractional part of a number. As shown in the upper box of figure 3-5, this is accomplished by noting the value of the exponent (8 in the example) and saving that number of places in the most significant part of the mantissa. The remaining bits of the mantissa are cleared (to zeros). As shown by the numeric graph on the right side of the box, the effect for positive numbers is to reduce the X-register contents (if there is a fraction) to the integer-only value. For negative numbers, since the representation is in two's complement form, the value becomes the next more negative integer (even if there is no fraction).

3-128. RNX. Rounding an extended floating point number to 24 bits of significance implies that a specific number of bits (16) will be cleared. Before these bits are cleared, however, the most significant bit of those to be cleared is examined. If this bit is a "1", indicating for positive numbers that the part of the number to be cleared represents a value of 1/2 (or more) of the least significant bit of the saved part, the saved part is incremented by +1. As shown by the numeric graph, if the cleared part of a positive number is 1/2 or greater, the value is rounded upward (not necessarily to an integer); otherwise, the value is rounded downward. If the cleared part of a negative number is -1/2 or more negative, the value is rounded in the negative direction; otherwise, the value is rounded in the positive direction. Note that if all 23 significant data bits were "1's" before execution, rounding upward (incrementing by +1) would cause rollover; unless an overflow condition exists, the instruction will automatically renormalize the number.

3-129. FIX. Rounding an extended floating point number to the nearest integer involves two steps: conversion to integer, and rounding. The lower box in figure 3-5 shows both steps. First, the mantissa is shifted right while the exponent value is incremented, once per shift, until the exponent equals +31. Then the most significant bit of the eight bits to be cleared is examined. If this bit is a "1", the saved part is incremented by +1. For positive numbers, the "1" bit indicates a fraction of 1/2 or greater; for negative numbers, it indicates a fraction smaller than 1/2. After rounding, bits 8 through 15 are cleared. As shown by the numeric graph, if the cleared part of a positive number is 1/2 or greater, the saved part is rounded to the next higher integer; otherwise the number is reduced to the integer-only value. If the cleared part of a negative number is -1/2 or more negative, the value is rounded to the next more negative integer; otherwise the number becomes the integer-only value.

3-130. POWER FAIL.

3-131. When the HP 2152A Floating Point Processor is installed in a computer system, the power fail subroutine must be revised to save the contents of the X-register (STX). If the computer has the optional restart feature, restore the X-register contents with LDX.

SECTION IV
THEORY OF OPERATION
FLOATING POINT PROCESSOR UNIT

4-1. SCOPE OF SECTION.

4-2. As shown in figure 4-1, a complete HP 2152A installation consists of a floating point processor unit, external to the computer and a controller installed internally in the computer. This section of the manual contains theory of operation of the floating point processor unit. Section V of this manual contains the EAU controller theory.

4-3. Included in the floating point processor unit are 10 logic cards (including an interface card), a test card, and a power supply. The test card is used only in diagnostic procedures, and is not described in this section; the description and use of the test card are included in the Manual of Diagnostics.

4-4. Paragraphs 4-5 through 4-90 describe the floating point processor logic, including the interface card. Paragraphs 4-91 through 4-111 describe the power supply.

4-5. INTRODUCTION.

4-6. The floating point processor has two basic modes of operation, one for two-operand routines, and one for function routines. (The load and store routines are considered to be interface control routines, and are discussed with the EAU controller in section V.) Figure 4-2 illustrates these two basic modes of operation.

4-7. In the figure, note that the three registers, A, B, C, together comprise what was earlier called, for simplicity, the X-register. Each of the three registers has 48 bits for data (mantissa), and also has 8 bits for an exponent byte (E) and 8 bits for a shift control byte (S). Since the C-register has no facilities for shifting, the C shift control byte is used to receive the opcode from the EAU controller. Shaded areas in the figure indicate the location of operands at the start of the operations. The following paragraphs outline the two basic modes.

4-8. FUNCTION ROUTINES. The function routines, defined earlier in this manual, calculate trigonometric and other mathematical functions on a quantity previously loaded into the floating point processor unit. Box A of figure 4-2 illustrates the operations involved.

4-9. The operand quantity x is assumed to exist in the FPP B-register. The operation begins when the EAU controller decodes the fact that the current instruction is of the function type. The controller then puts the opcode (least significant 8 bits of the instruction) on the interface data lines and issues an OPC (Opcode) command to the FPP unit.

4-10. The FPP unit, which operates under control of firmware programs in the ROM (read-only memory), cycles in a wait mode as long as it is in the ready state, looking for a command such as OPC. (The other command is ENC, discussed under two-

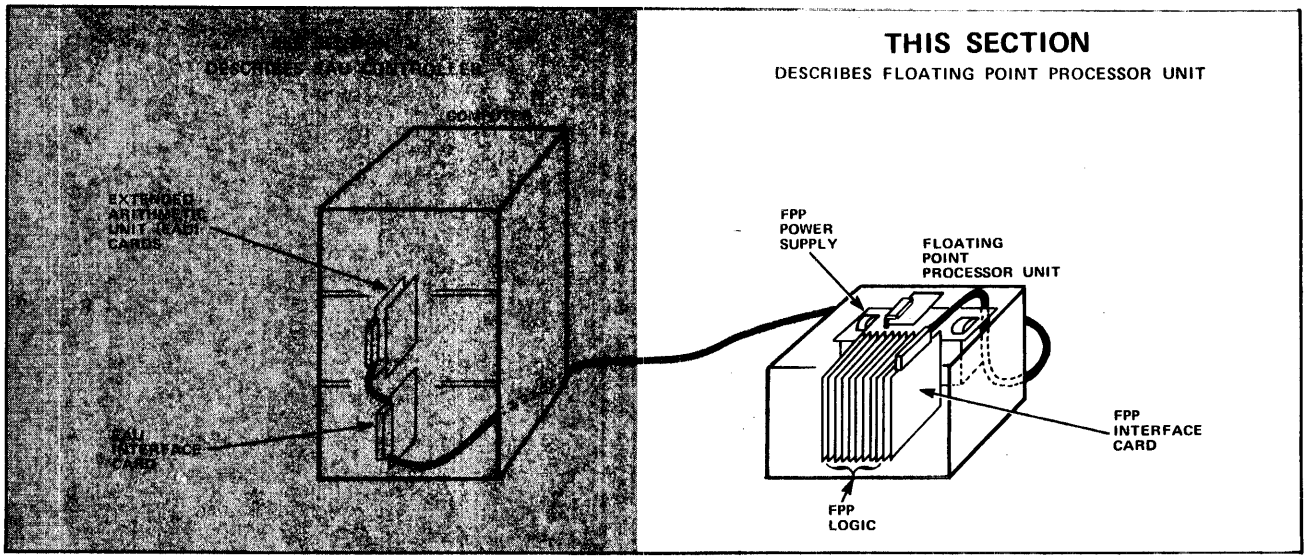


Figure 4-1. Processor Section

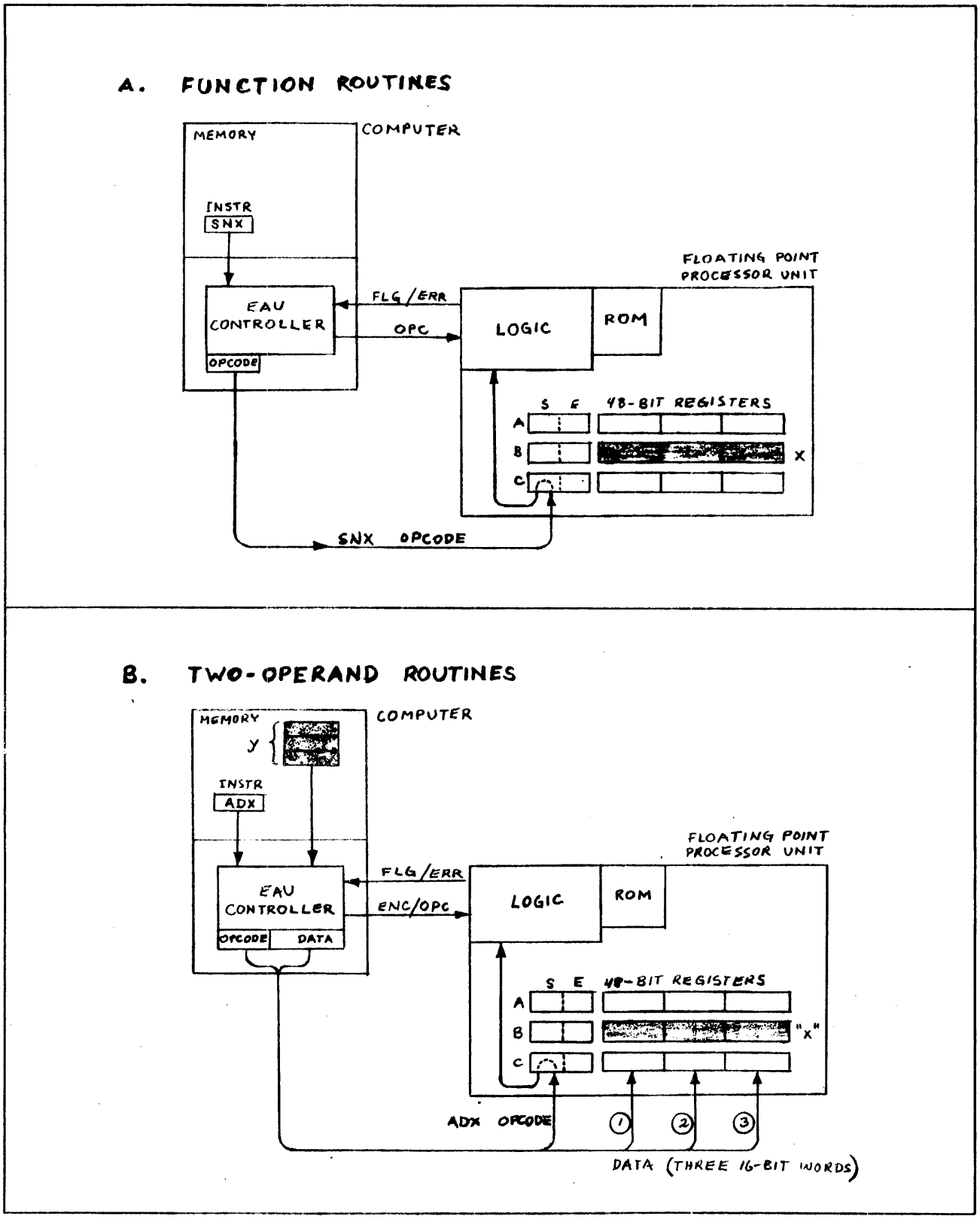


Figure 4-2. Basic FPP Operations

operand routines.) When the ROM program detects the presence of OPC, it loads the opcode data into the S byte of the FPP C-register. Then the ROM identifies the type of operation by decoding the opcode bits, and branches to the appropriate function routine. The opcode is now no longer needed.

4-11. As the program proceeds, the value in the FPP B-register is manipulated according to the algorithm for the particular function, using all three registers. At the end of the routine, the final answer resides in the FPP B-register, and a FLG (Flag) signal is sent back to the EAU controller. The FLG signal indicates that the unit is ready for further commands, having completed the last issued command.

4-12. The FLG signal, incidentally, has no association with the computer flag and interrupt system. It is simply an interface signal between the FPP unit and the EAU controller. The FLG signal indicates to the EAU controller that it should allow the computer to proceed to the next instruction.

4-13. If an error occurs during the calculation, or if the opcode is improper, an ERR (Error) signal is simultaneously sent back to the EAU controller with the FLG signal. This will set the computer Overflow flip-flop, which may then be tested by an SOS or SOC instruction. Also, the error code will appear in the computer A-register, where it may be checked by an error routine. It is the user's option to decide what to do about an error condition. In general, it may be said that the FPP unit will attempt the calculation, rather than abort, even if input values will result in an error. The FPP unit will provide the best answer it can, along with the error indication. This allows the programmer some flexibility to reconstruct correct answers from results which normally could not be represented. The exception to this generalization is that divisions by zero will not be attempted.

4-14. TWO-OPERAND ROUTINES. The two-operand routines take two values, one that was previously loaded into the FPP unit, and one that exists in memory, and operate on these numbers in some specified way. The operations include: adding the two numbers together, multiplying one by the other, subtracting, and dividing. The value in memory may be either double-length or triple-length; the value in the FPP unit (and the answer) will always be triple-length. Box B of figure 4-2 illustrates the operations involved.

4-15. Initially the quantity x is assumed to exist in the FPP B-register. (It may have been left there as the result of a previous instruction, or it may have been loaded by a load instruction. The load instructions operate similarly to the following procedure, except that the opcode simply causes the loaded FPP C-register contents to move up into the FPP B-register.) The operation begins when the EAU controller decodes the fact that the current instruction is of the two-operand type. The controller first fetches one word of the three-word operand (y) from memory. It then puts this data word on the interface data lines and issues an ENC (Encode) command to the FPP unit.

4-16. As mentioned previously, the FPP unit, under control of the ROM programs, continuously searches for ENC or OPC commands as long as it is in the ready state. When the ROM program detects the presence of ENC, it loads the data word (in two 8-bit bytes) into the high order third of the FPP C-register.

4-17. After the second byte has been loaded, the FPP unit sends a FLG signal back to the EAU controller, indicating readiness for the next word of y. The EAU fetches this next word from memory and repeats the process: the word is placed on the interface data lines, ENC commands the FPP unit to load these two bytes, and another FLG signal is issued to again repeat the process for the third and final time.

4-18. At the end of the three-word transfer, the quantity x is in the FPP B-register and the quantity y is in the FPP C-register. The FPP unit now needs to be told what to do with these numbers. Thus the entire process described above under function routines must now be added on. In brief, the procedure is:

a. The FLG signal allows the EAU controller to issue OPC.

b. The FPP unit loads the opcode into the FPP C-register.

c. The ROM program interprets the opcode and branches to the appropriate function routine (add, subtract, etc.).

d. The function routine calculates the answer, and leaves it in the FPP B-register.

e. A final FLG back to the EAU controller tells it that the FPP unit is ready for further commands.

f. In case of error, an ERR signal causes setting of the Overflow flip-flop and loading of the computer A-register with the error code.

4-19. FPP DETAILED THEORY.

4-20. The remainder of this section describes in detail the procedures described above (up to paragraph 4-90), plus a description of the FPP unit power supply (paragraphs 4-91 through 4-111). All logic is positive-true. The high (or true) state ranges from +1.25 to +2.5 volts; the low (or false) state ranges from -0.5 to +0.5 volts.

4-21. BLOCK DIAGRAM. The block diagram (figure 4-11) at the end of this section can remain folded out for convenient reference throughout this part of the section. Facing the block diagram are two tables and a figure which provide supporting information: the detailed coding of the instruction register, definitions

of the ROM instructions, and a list of tests used for branching decisions. These tables and the figure should be referred to frequently, since definitions will not be given within the descriptive text.

4-22. LOGIC DIAGRAMS. The logic diagrams for the FPP unit are given in section VII. Since the logic itself is comparatively simple (at the gate level), the logic diagrams will generally not be referred to in these discussions. Specific signal names have been given on the block diagram, facilitating direct reference from a function on the block diagram to the comparable function on the logic diagrams. Wiring lists and signal indexes and signal tracing from board to board.

4-23. FPP CLOCK AND TIMING.

4-24. The clock generator for the floating point processor unit operates at a rate of 5 MHz (200-nanosecond period) supplying a 100-nanosecond clock signal, and its complement, to the FPP logic. The clock generator is located on the FPP D-register card.

4-25. The complemented clock signal ($\overline{\text{Clock}}$) allows the clock cycle to be split, so that an active bit may be loaded into a register in the first 50 nanoseconds and perform its function in the second 50 nanoseconds. This high-speed feature employs a master/slave pair of flip-flops for each bit. (See figure 4-3.)

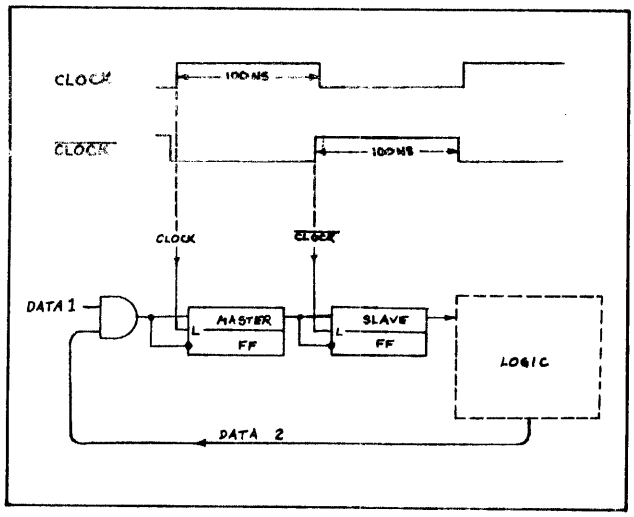


Figure 4-3. Clock Timing

4-26. Note that the Clock signal latches the master flip-flop, and Clock latches the slave flip-flop. The master flip-flop loads the data bit (Data 1) at the start of the Clock signal, and about 45 nanoseconds later Clock transfers the bit to the slave flip-flop. (There is a slight offset between clocks.) The output of the slave flip-flop can then be used in the logic without the possibility of resultant actions (Data 2) affecting any conditional inputs that determined the setting of the master flip-flop.

4-27. READ-ONLY MEMORY.

4-28. The read-only memory (ROM) is essentially a diode encoder. A certain combination of signals on the input lines (address) is applied to a diode matrix, activating a certain combination of output lines (contents). The FPP ROM has nine input lines (RA0 through RA8), thus allowing 512 addresses (2^9), and 48 output lines (R0 through R47), giving a word length of 48 bits.

4-29. The ROM constantly reads out whatever contents are enabled by the nine address lines. ROM itself is not strobed. Instead, the ROM output lines are clocked into either the instruction register (normally) or the D-register (if the preceding instruction contained a constant call). All words in ROM are either instructions or constants. For start-up purposes (power-on), ROM is always forced to start at address 0.

4-30. The FPP ROM contents are listed in two separate tables in section VII of this manual. This first table gives the octal printout of the logic 1/0 bit, pattern for each address. The second table lists the contents in the form of mnemonics and constants.

4-31. Physically, the matrix is contained in 24 microcircuit packages on a single printed circuit card. Each package accepts 8 of the 9 address lines and has 4 of the 48 output lines. (See logic diagram, figure 7-10.) The ninth address bit (RA8) selects either the high half or low half of ROM by enabling one of the two ranks of packages. The lower rank of 12 packages is enabled when RA8 is "0" and is therefore active for address 0 through 255 (decimal). The upper rank of 12 packages is enabled when RA8 is "1" and is therefore active for address 256 through 511. The output of the two ranks are "or"-tied together.

4-32. INSTRUCTION REGISTER.

4-33. The instruction register is clocked to load the ROM output every 200 nanoseconds. As explained previously (figure 4-3), the data word is loaded into the master flip-flop (of the instruction register) by the block signed and into the slave flip-flop by $\overline{\text{Clock}}$. Therefore, at $\overline{\text{Clock}}$ time, the contents of the instruction register are applied as command lines to various points throughout the logic (see block diagram). Loading the instruction register has occupied one clock cycle. See figure 4-4 (time intervals 1 and 2).

4-34. As soon as the instruction is in the slave latch of the instruction register (2), execution begins. A typical execution would read a pair of bytes and add them during $\overline{\text{Clock}}$ time (2) and store the result during the next Clock time (3) into the master rank of the specified register. The next $\overline{\text{Clock}}$ (4) would transfer the stored result from master to slave, where it may be used (read) by the next instruction. Notice that there is a time overlap, and the second instruction has already been loaded from ROM (3) and execution has begun (4).

4-35. Since addresses are not automatically incremented after instruction execution, each instruction

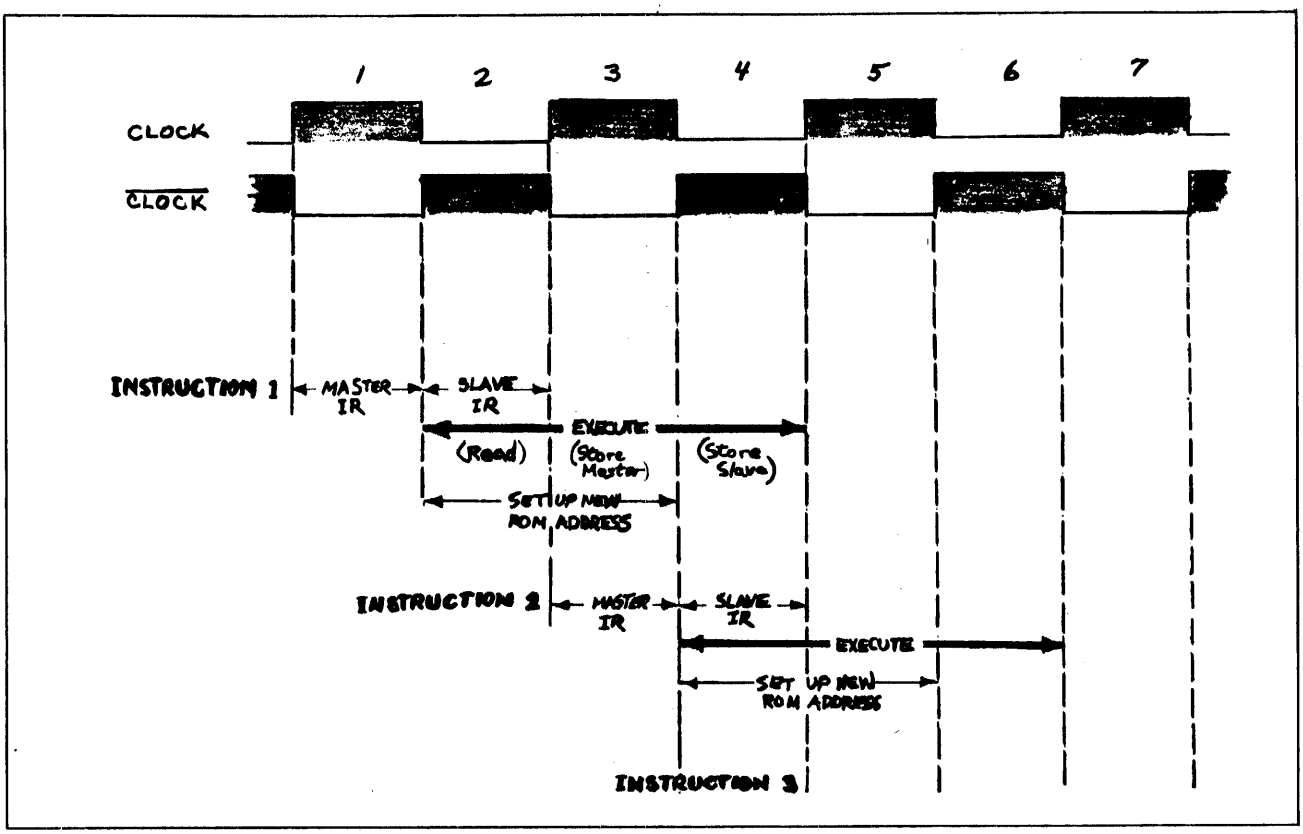


Figure 4-4. Instruction Execution Timing

must specify where the next instruction is to be obtained. This is done during time interval 2, either directly by the contents of the BP and BL fields, or indirectly by the branching or JSB logic (discussed in succeeding paragraphs).

4-36. Detailed coding of the instruction register, plus definitions of the instruction fields, are presented facing the block diagram. Physically, the register is split up and located on three separate cards: ROM address card, D-register card, and FPP interface card.

4-37. ROM ADDRESSING.

4-38. The floating point algorithms require considerable flexibility for branching from one area of ROM to another. The following paragraphs describe the various modes employed to specify the next address within a current instruction. All addressing modes are tabularized in table 4-1 for convenient reference; this table may be referred to throughout the following discussions.

4-39. CONDITIONAL BRANCHING.

4-40. By instruction, the ROM address may be caused to branch to one or the other of two specified addresses, dependent on a certain condition being either true or false. One of 16 conditions may be selected for the true/false test by the TS field of the instruction. These conditions are numbered T0 through T15. (See table 4-4.)

4-41. The block diagram shows the sources of several, but not all, of the test inputs; for example, OPC and ENC from the computer, selected outputs of the A, B, C adders, and certain count values of the byte counter. Note that these signals are applied as one input to a three-input gate in the conditional branching block. (This gate represents a series of gates performing this function.) The second input to the gate is the decoded test number, and the third is the BRN signal (also decoded from the instruction register) which must be true for all branching instructions.

4-42. The output of the three-input gate in the conditioned branching block will be either true or false, depending on the condition of the test input. If true, the upper of the two other gates in the block will be enabled; if false, the lower gate will be enabled. Therefore, if the test is true, the four least significant bits of the ROM address (line) will take the value of the BL field; if the test is false, line will take the value of the BP field. (The page value remains unchanged, since the current page value is stored by the MP register, and read out by the three-input gate that is enabled by BRN and TRU.) It follows, therefore,

Table 4-1. ROM Addressing Modes

MODE	NEXT ADDRESS	
	PAGE	LINE
Unconditional Branching (TS = 0)	BP	BL
Conditional Branching		
TS True	MP (Current Page)	BL
TS False	MP (Current Page)	BP
Indirect and Constants		
IND	CS (4-7) Bit 8 = 0	CS (0-3)
CON	CS (4-7) Bit 8 = AD9	CS (0-3)
JSB and Return		
JSB	BP	BL
(JSB and RTN complement JAR after execution)	(If JAR = 0, save MP in FP) (If JAR = 1, save MP in GP)	(If JAR = 0, save TS in FL) (If JAR = 1, save TS in GL)
RTN	If JAR = 0: GP If JAR = 1: FP	If JAR = 0: GL If JAR = 1: FL

that conditional branches may be made only to one of any two lines on the current page.

4-43. UNCONDITIONAL BRANCHING.

4-44. Most of the instructions in ROM call for conditional branches. However, it is also possible to specify an unconditional branch. This simply gives ROM the next address, regardless of test conditions. One example of usage would be to cause execution of two or more instruction words in a sequential series.

4-45. The page and line values are given by the BP and BL fields, respectively. The transfer is accomplished by coding BRN and TRU in the instruction. This enables the first of the three gates in the unconditional branching block, which in turn enables BP and BL onto the page and line address lines.

4-46. JSB CONTROL.

4-47. The JSB instruction is an unconditional branch with two special provisions: a specific address value is stored, allowing return from subroutine completion to a desired address, and register switching is provided to allow one level of JSB nesting.

4-48. The decoded JSB signal enables BP and BL onto the page and line address lines (in the unconditional branching block), the same as described earlier when BRN and TRU were enabled for the unconditional branch. The same JSB signal also loads the current page value into either the GP or FP register (depending on the current state of the JAR flip-flop), and loads a return line value from the TS field into either the GL or FL register (also depending on the JAR flip-flop state). The TS field is used for line specification since its normal usage (for conditional branches) does not apply for JSB. The net result is that a return address is stored in either GP and GL or FP and FL. Since the stored page value is always the current page value, it follows that subroutine returns must be to the same page that contained the JSB call. Furthermore, since only four of the five page bits are stored, both the call and the return must be in the lower half of ROM (addresses 0 through 255). The subroutine itself, however, may be located on any page, specified in the BP field.

4-49. After the return address has been stored (on block), the JAR flip-flop toggles to its complementary state (on block). The initial state of JAR is random, and it is immaterial whether the G or F pair of registers is the first selected; the important fact is that JAR toggles after each occurrence of JSB or RTN. Assume for this discussion that the JAR flip-flop was in the set state and therefore has loaded the return address into GP and GL. It then toggled to the reset state. In this state (note the four output gates of JSB Control), an RTN signal would read the GP/GL contents onto the address lines and then toggle the JAR flip-flop back to the initial set state. If, however, another JSB were issued before an RTN (note the four input gates), a second return address would be loaded into FP and FL. The JAR flip-flop then toggles to the set state, so the first RTN will read back this second

return address, thus ensuring a return from the nested subroutine before going into position for returning from the "outer" subroutine. Any number of nested subroutines may be called out and completed before returning to the outer routine, provided they occur in succession; since storage is provided for only two return addresses, only one level of nesting can exist at a given time.

4-50. CONSTANT CALL.

4-51. As mentioned previously, all words in ROM are either instructions or constants. Instructions are loaded into the instruction register, and constants are loaded into the FPP D-register (48 bits). To obtain a constant, the ROM program must first load the address of the desired constant into the CS byte of the C-register, and then issue an instruction containing the CON code in the BC field. Since CS contains only 8 bits (and the ninth bit is forced to a "1"), constants must be in the upper half of ROM (addresses 256 through 511).

4-52. The CON signal enables the CS byte onto the ROM address lines, stops Clock for one cycle only, and enables a special clock that loads the addressed ROM contents into the D-register. At this point, reference should be made to the D-register logic diagram. When the CON bit (R21) is detected, it is loaded by TS (equivalent to Clock) into the master CON latch. At $\overline{\text{Clock}}$ time, the bit is transferred to the slave latch and inverted by U92B to disable Clock at U72B. (Note that $\overline{\text{Clock}}$ is not affected.) The low input to pin 6 (input control code = 01) causes the master latch to clear, so that at the next $\overline{\text{Clock}}$ CON will go false. This, inverted to true by U92B, re-enables Clock. The net result is that one Clock pulse has been inhibited, temporarily halting program execution while the constant is being read.

4-53. Note, however, that the TS clock has continued to run, and this clock, enabled during the interval that CON is high, loads ROM bits R0 through R47 (the constant) into the D-register.

4-54. If a series of constants is called, the CIC bit (R38) may be used to increment the constant address in CS. The purpose of the CKC flip-flop (which, like the CON flip-flop, is clocked by TS) is to assume the function of the CIC flip-flop (disabled in the absence of Clock) during the CON cycle. Note that this feature applies only for CS addresses below 24 (decimal). These 24 locations provide a table that results in a numerical convergence after 25 steps. The S24 signal inhibits CIC for addresses 24 or higher in order to conserve ROM space. The program may continue to call for constants and keep issuing the CIC command, but in effect the contents of location 24 will continue to be read on each further call.

4-55. The CON signal, which goes off the board at pin 82, goes to the ROM address card for addressing the constant. On the ROM address card, note that CON is "or"-tied with the Indirect command line (IND); refer to next paragraph.

4-56. INDIRECT ADDRESSING.

4-57. IND enables the 8-bit CS byte onto the ROM address lines. For IND alone, the low half of ROM is addressable (0 to 255) since the ninth address bit (RA8) cannot be controlled by the 8-bit CS register. For CON, however, U61A forces the ninth bit to a "1" (since AD9 is normally "0"), so that constants will always be read from the high half of ROM (256 to 511). For certain purposes (such as the ROM dump routine), AD9 can be made true, so that CON can also read the lower half of ROM.

4-58. A - REGISTER/SHIFTER/ADDER.

4-59. The floating point processor contains three nearly identical arithmetic sections, each typically consisting of a register, a shifter, and an adder. The A-register/shifter/adder will be discussed in detail first; differences for B and C will be described later.

4-60. The FPP A-register accommodates 48 bits of data in six separately controllable bytes (A0 through A5). In addition, there is an exponent byte (AE) and a shift byte (AS).

4-61. The A-shifter provides a means of selecting any eight adjacent data bits from the 48 bits stored in the FPP A-register, irrespective of byte boundaries.

4-62. The A-adder adds an 8-bit byte, selected from the A-register (unshifted), to another 8-bit byte, selected from the A-, B-, C-, or D-register (shifted or unshifted).

4-63. With reference to the block diagram, the logic will be discussed left to right across the A-register/shifter/adder block. On the left side of the block is a series of four transfer mode gates (each representing eight separate gates for the complete byte). If transfer mode 1 is selected in the ROM instruction, one of the C-register bytes (on the C50 through C57 lines) will be transferred as an input to the A-adder. (The C byte number selected will be the same as the A byte number selected; refer to paragraph 4-69.) Similarly, if transfer mode 2, 3, or 4 is selected, the gates will transfer a shifted B byte (on the B70 through B77 lines), a shifted D byte (on the D70 through D77 lines) or a shifted A byte (on the A70 through A77 lines).

4-64. The output of the Transfer Mode gates is applied to a true/complement network consisting of an "and"/"nor" pair of gates for each bit. If the CPA instruction bit is true, each bit is complemented before being routed to the A-adder on the A90 through A97 lines.

4-65. The other input to the A-adder (lines A60 through A67) is enabled if the RRA bit of the instruction register is true. The input consists of one of the FPP A-register bytes on the A50 through A57 lines (byte selection described later in paragraph 4-69).

4-66. The result of the addition appears on the A00 through A07 lines, with a possible carry saved in the

CY bit register. The carry may be used (propagated) by a PCY instruction in the next cycle. It is also possible to inject a carry (actually an increment by one) by means of a CIA signal. PCY and CIA are functions of the Special field of the instruction word, as is BI8, which can force a one on the eighth bit (A97) of the transferred input to the adder.

4-67. Tests which can be made on the A00 through A07 output (see conditional branching, discussed earlier) are: eighth bit true or false, eighth bit of A does or does not equal eighth bit of B, and adder output is zero or non-zero.

4-68. The adder output is applied to all eight byte positions of the A-register. (the data is stored in a master register at Clock time). At Clock time, the data will be transferred into a byte position (slave register) which is selected by one of eight enabling signals: AY0 through AY5, AYE, or AYS. The enabling signal is derived from the SR, SY, and YC fields of the instruction register. The SR field specifies the A-register (SRA), and SY either specifies byte AS, AE, A5 or else enables the YC field (byte counter) to select one of the six data bytes, A0 through A5. The byte counter produces an octal output on the ROM address card, consisting of signals Y0, Y1, Y2, which is decoded on the FPP interface card to produce the SY0 through SY5 signals. The decoder is not enabled if the SY field specifies SY5 (store in A5 byte), SYS (store in AS byte) or SYE (store in AE byte).

4-69. The bytes stored in the FPP A-register can be selectively read out by read signals derived from the RY and YC fields of the instruction register. The RY field either specifies byte AS, AE, or A5 (by RYS, RYE, or RY5 signals), or else enables the decoded byte count from the YC field to select one of the six data bytes (by the RY0 through RY5 signals). The selected byte is routed via the A50 through A57 lines to the A-adder.

4-70. In addition to the selective byte reading described in the preceding paragraph, provision is also made to read any adjacent eight bits in the data portion of the A-register. Byte boundaries are ignored, and the register is looked at as a 48-bit data register. Selection is accomplished by the A-shifter, under control of the shift byte (AS) in the A-register.

4-71. The shifter may be viewed as an addressable reader. (See figure 4-5.) The numerical value of the shift byte (decimal) points to the least significant bit of the desired 8-bit series. This bit and the next higher seven bits are read out to the transfer mode gates.

4-72. As shown in figure 4-5, special cases occur when the shift byte points to bit positions higher than 40. (Seven of the eight bits of AS are used for addressable reading, so AS can point to values as high as 127.) When the AS value is between 41 and 47 (inclusive), one or more bits selected at the high end of the series of eight will be nonexistent. These nonexistent bits are referred to as phantom bits (P); provision is made to fill these bit positions on the output lines (A70 through A77) with either zeros or copies of the sign

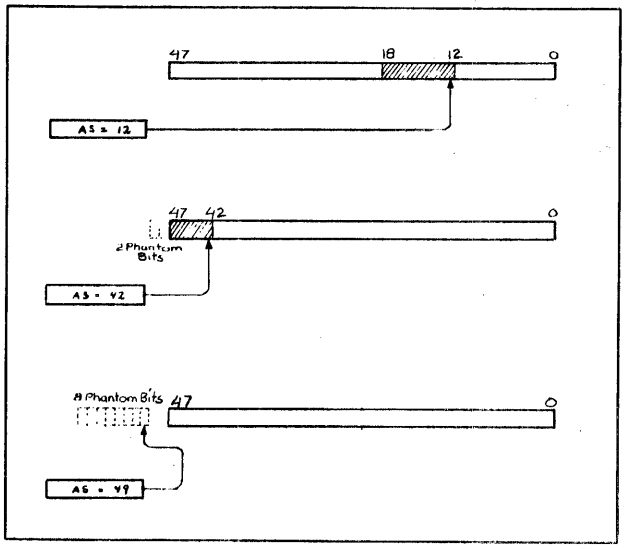


Figure 4-5. Addressable Reading by Shifter

bit (bit 47). The desired choice is made by controlling bit 7 of the shift byte (AS7): if "0", signs will be copied (arithmetic shifting); if "1", zeros will be filled in (logical shifting). Note that when AS is 48 or higher, all of the selected bits will be phantom bits.

4-73. Details of the selection process are shown in figure 4-6. The six least significant bits of AS are decoded octally into two sets of selection signals, designated SW0 through SW7 and SV0 through SV7. (AS6, if true, would result in the all-phantom condition, so it is not decoded but is simply "or"-tied with SW6 and SW7; see next paragraph.) The SW0 through SW5 signals accomplish a preselection of 15 out of the 48 register bits, and the SV0 through SV5 signals select 8 out of the 15 preselected bits. These final eight bits are routed out on the A70 through A77 lines.

4-74. Refer to the A-shifter logic diagram for details on the generation of phantom bits. Note that U50C enables sign bit 47 to the higher order SW5 positions if AS7 is "0". If AS7 is a "1", the output of U50C is "0". (Final selection of one or more of these bits is made by the SV0 through SV7 signals.) For the all-phantom condition, the shifter network is ignored completely (all zeros on the A70-77 lines); instead, a true or false TSA signal is sent to the complementing networks. Gate U50D is enabled by SW6, SW7 or AS6, and will provide a true output if phantom signs are desired (AS7 = "0") and the sign bit happens to be a "1". Otherwise, if the sign bit is "0" or if phantom zeros are desired (AS7 = "1"), TSA will be false. Depending on the transfer mode selected, TSA will affect one of the three complementers (A, B, or C) by inverting the existing all-zero output to all ones (TSA true) or will leave the data as all zeros (TSA false). The result is eight copies of the sign ("1" or "0"), or eight zeros.

4-75. For microprogramming purposes, it is advantageous to have the pointer in AS keep in step with the

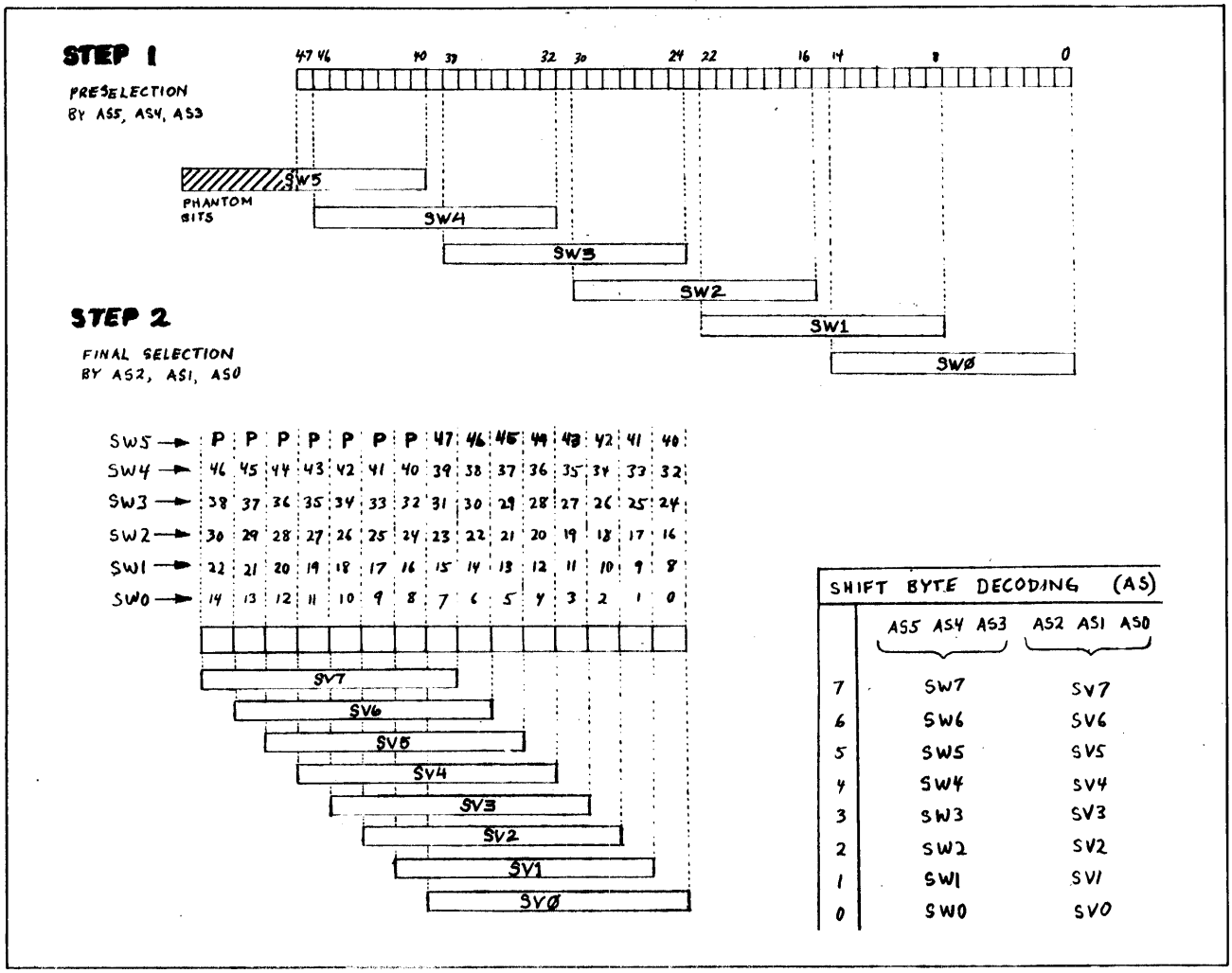


Figure 4-6. A-Shifter Selection Process

byte counter. This means that when the byte counter is incremented or decremented to enable the next higher or lower byte position, the shift pointer should also change value to enable the next higher or lower series of eight bits. The AY8 adder performs this function.

4-76. In order for the AS value to point to a new series of eight bits, its value must increase by 8 when YP1 increments the byte counter and must decrease by 8 when YM1 decrements the byte counter. Furthermore, when the byte counter rolls over from 5 to 0 (incrementing, modulo 6) or from 0 to 5 (decrementing), the AS value must change correspondingly: return to its original value or go to the original value plus 40 (i.e., 5 x 8), respectively.

4-77. Referring to the ROM address card logic diagram, figure 7-8, note that when YP1 increments the byte counter (via U35E), it also increments the AY8 adder. Since the AY8 adder operates on bits 3 through 6 of the AS register (rather than 0 through 3), each increment adds 8 to the contents of AS, via the AP3 through AP6 lines. Similarly, when YM1 decrements the byte counter (by adding all "1's" via U35D/C, U33B, and U32A), it also decrements the AY8 adder decrementing AS by 8 via the AP3-6 lines.

4-78. Gates U35E, U33A, and U32D cause the byte counter (and AY8 and BY8 adders) to act as modulo 6 counters when incrementing. When the count of 5 is detected by U24A and U15D, the next YP1 will inject a quantity which, when added to 5, will produce zero. For the 3-bit byte counter this quantity is 3 (via U35 U35E and U33A). For the 4-bit AY8 and BY8 adders this quantity is 11 (all three gates).

4-79. To achieve modulo 6 when decrementing, gates U33B and U32A are disabled at the count of zero, and allow U35D and U35C to inject a quantity of 5. This reverts the byte counter to the count of 5 and adds 40 to the AS register via the AP3 through AP6 lines. (Incidentally, the AP3-6 lines are disabled when AS is originally loaded, by the $\overline{\text{SYSA}}$ signal.)

4-80. The method by which the byte counter is forced to zero (YF0) is to add the current value of Y to its complement (U35C, U33C, U16B) and inject a carry (U35A). For the 4-bit adders, U32B injects the necessary one-bit for the most significant bit position.

4-81. B/C/D ARITHMETIC SECTIONS.

4-82. The FPP B-register/shifter/adder is identical to the A section described in the preceding pages, with only signal nomenclature changes and a different assignment of inputs for the transfer modes.

4-83. The C-register/adder does not have an associated shifter. Instead, the third shifter is assigned to the D-register. The D-shifter is controlled in parallel with the A-shifter by the AS shift byte.

4-84. Since the CS byte is used for indirect addressing of ROM (see paragraph 4-51), the CS output is routed to the ROM address card. Also the S24 signal (discussed in paragraph 4-54) is made available to the conditional branching test logic.

4-85. On the C-adder card, the CSX line is open ("0"), whereas on the A- and B- adder cards ASX and BSX are enabled by tying to +4.75 volts. This disables the CP input lines to the shift byte (CS), since these lines are not used in the C arithmetic section.

4-86. IN/OUT TRANSFER SEQUENCE.

4-87. As mentioned in the introduction to this section, data is transferred into or out of the FPP unit in three successive 16-bit words. It was also stated that EAU sends 16 bits of data with every ENC, and the FPP unit returns 16 bits of data with every FLG, whether or not data is actually used at either end. Data to the FPP is loaded into the C-register (and transferred to the B-register if a load opcode follows), and is sent from the B-register. The process is as follows. (Refer to table 4-2.)

4-88. On the first ENC, the entry routine first loads the high order eight bits (IM0 through IM7) into C5 while, simultaneously, B5 is transferred to A4 and read out to the output lines (A50 through A57). Then the byte counter is decremented (pointing to byte 4). The low order input bits (IL0-7) are loaded into C4, and B4 is read out to the B50 through B57 lines. A FLG signal is issued to EAU, telling it that it can store the 16 bits from A4 and B4.

Table 4-2. X Register Transfer Sequence

ENC	INPUT	OUTPUT
#1	IM0-7 → C5	B5 → A4
	IL0-7 → C4	A4 (FLG)
#2	IM0-7 → C3	B3 → A2
	IL0-7 → C2	B2 (FLG)
#3	IM0-7 → C1	B1 → A0
	IL0-7 → Convert to FPP format → C0	B0 Convert to FPP format (FLG)

4-89. On the second ENC, the byte counter decrements to 3, and IM0-7 is loaded into C-3, while B3 is transferred to A2. Decrementing to count 2 allows C2 to be loaded, and A2 and B2 to be read out (with FLG).

4-90. On the third ENC, the byte counter decrements to 1, and IM0-7 is loaded into C1. Then, when the byte counter decrements to 0, a format conversion occurs which moves the exponent sign bit to the proper position. (Internally in the FPP unit, this bit is in the most significant bit position; externally in the computer, it is in the least significant bit position.) Byte B1 is now transferred to A0, and A0 and B0 are read out (with FLG).

NOTE

This completes the discussion of the logic portion of the floating point processor unit. The remainder of the section discusses the internal power supply of the unit.

4-91. POWER SUPPLY.

4-92. The power supply of the floating point processor generates two regulated dc supply voltages for all logic circuits in the unit: +4.75 volts and -2 volts. (A third dc voltage, +10 volts, is also generated, but this supply is used only within the power supply itself.)

4-93. Figure 4-7 illustrates the power supply circuits in simplified form. The 115- or 230- volt ac input is stepped down to a nominal 12 volts ac and rectified by a pair of silicon-controlled rectifiers (SCR). The inductor/capacitor filtered output is 6.75 volts dc, referenced to ground such that the positive line is +4.75 volts and the negative line is -2 volts with respect to ground.

4-94. The full 35-ampere current capacity is available to the +4.75-volt load, and up to 35 amperes is available to the -2-volt load. Since the -2-volt load is less than the +4.75-volt load, the difference current is diverted through the -2-volt shunt regulator. This regulator acts in the same way as would a a Zener diode. A variable amount of current is drawn through the shunt in order to maintain a constant -2-volt level.

4-95. The level of the +4.75 voltage is maintained constant by controlling the conduction time of the SCR SCR's. The +4.75-volt level is detected by a differential amplifier, which compares the voltage with a Zener diode reference. The difference output is used to control the slope of a ramp voltage, which is synchronized to the 120 Hz rectified line frequency. When the ramp reaches the trigger level of a unijunction transistor in the ramp generator, the ramp terminates, generating a positive pulse of about 10 volts

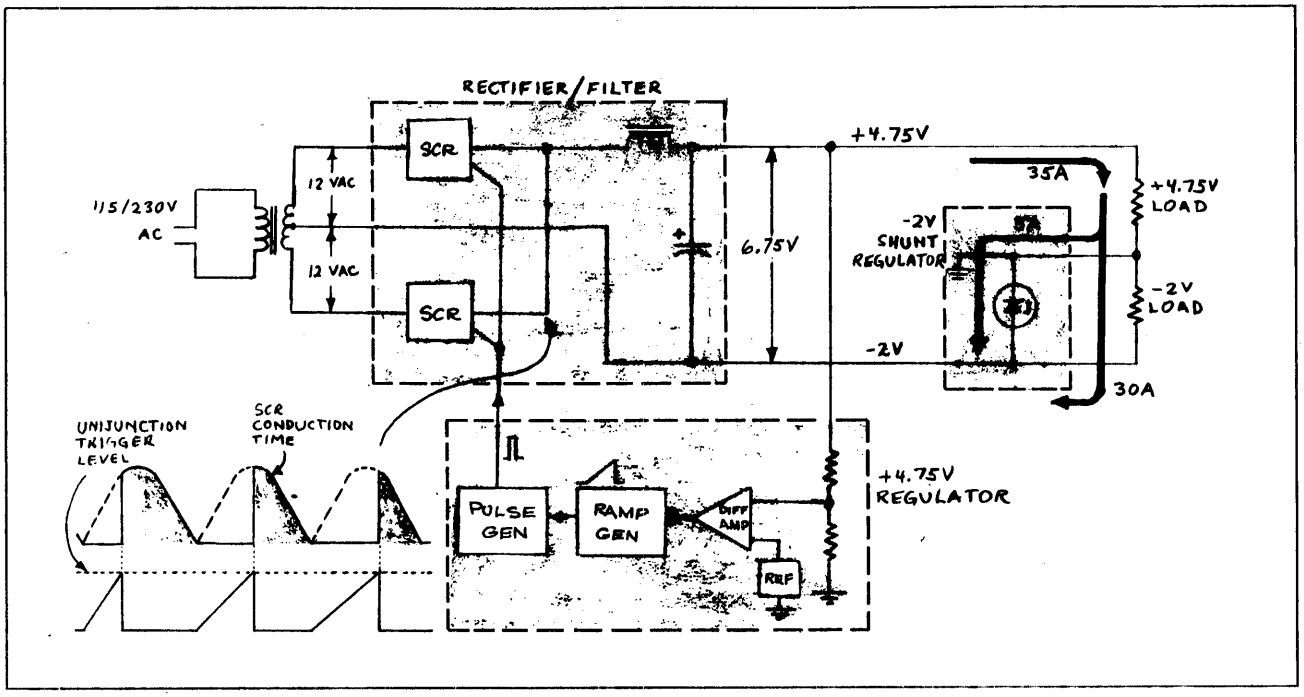


Figure 4-7. Power Supply Block Diagram

amplitude and 20 microseconds duration. This pulse triggers the SCR's, which will then continue to conduct for the remainder of the half cycle. As shown in figure 4-7 (note examples of ramp slope and rectified sine wave), a variance of ramp slope has the net result of altering the conduction time (shaded area). Consequently, the energy delivered to the LC filter will be increased or reduced proportionately, thus providing the means of controlling the output dc level.

4-96. Refer to the power supply schematic, figure 7-30, for the following detailed circuit descriptions.

4-97. AC INPUT CIRCUIT. Input ac power is applied to power line assembly A1. This snap-in module contains the ac line connector, line fuse, rf interference filter, 115/230V line voltage switch, and terminals for connection of the front panel POWER switch, power-on indicator lamp (DS1), and power transformer. Relay K1 is inserted in series with the transformer primary, so that power will be turned off if either the computer loses power (-23.8V drops) or the ambient temperature in the FPP unit rises too high.

4-98. +4.75V REGULATOR. Sensing of the +4.75-volt level is made from a point on the backplane bus. Due to the high currents involved, bus resistance itself will drop the dc level slightly; power is therefore applied to the bus at two points, and the sense line is connected to a point that represents an average value.

4-99. The sensed +4.75 voltage is applied to a differential amplifier at Q1/Q2, which compares a divided sample (R20/R21) to a pre-settable reference level from resistor R25 (+4.75V ADJ). Any voltage difference between the bases of Q1 and Q2 is amplified and applied to Q3, altering the charging rate of ramp capacitor C30. When Q3 has charged C30 to the triggering level of unijunction transistor Q4, Q4 discharges C30 to the -2-volt clamping level. The sharp negative transition at the base of Q5 turns on Q5 for about 20 microseconds, dependent on circuit constants, and the resultant positive pulse is applied through emitter follower Q6 to the SCR trigger inputs (CR5, CR6). Diode CR22 limits the pulse amplitude to +10 volts and protects Q6; CR8 protects the SCR's (which are non-conducting before the pulse arrives).

4-100. The positive pulse turns on CR5 or CR6 (depending on the ac cycle polarity), charging filter capacitors C11, C12, and C13 through inductor L4. At the end of the half cycle, ac polarity reverses and the SCR ceases conduction. Since the other SCR will not begin its conduction until triggered, neither SCR is conducting at this time. The inductive field of L4 begins to collapse, building up a reverse voltage which could be destructive if protection were not provided. Diode CR7 provides this protection by coming into conduction when the reverse voltage exceeds the -2-volt level, and provides a current path back to the filter capacitors. Thus, even when both SCR's are off, the inductor still delivers current to the load. When the next SCR is triggered, it abruptly puts out a positive voltage to the inductor, and thus reverse biases CR7. In summary: CR7 conducts when the SCR's are not conducting.

4-101. As explained earlier (paragraph 4-95), the timing of the SCR trigger accomplishes the voltage regulating function.

4-102. SYNC AMPLIFIER. The primary purpose of Q7 is to synchronize the unijunction oscillator to twice the line frequency. A secondary function is to inhibit the triggering of unijunction transistor Q4 when the crowbar is on, thus reducing current delivered to the Crowbar, CR80. When the input voltage (pulsating dc from the input to L4) is in excess of +9 volts, Q7 is saturated (on), providing a low impedance path for the Q3 collector current, diverting it from C30. Thus the unijunction oscillator is held in the off state. (Note that a positive input from the crowbar, via Q29, could permanently hold the oscillator in this off state.) Then, when the pulsating voltage drops below +8 volts, Q7 is cut off, and the current from the Q3 collector is shunted to ramp capacitor C30. This results in a voltage ramp on the emitter of Q4, the slope of which (as discussed earlier) is determined by the collector current of Q3. The start of the ramp is therefore determined by the on-to-off transition of Q7, which occurs twice for each cycle of the line.

4-103. -2V SHUNT REGULATOR. The -2-volt sense voltage (refer to comment on sensing in paragraph 4-98) is applied through a pre-settable divider to the base of Q18. The bottom end of the divider is held constant by a Zener diode reference. The -2-volt adjustment resistor is set so that the Q18 base is at zero volts when the -2-volt output is at its nominal value. This zero-volt-level is compared with the zero-volt ground at the emitter of Q19. Any difference is amplified by Q19, Q20, and Q21, altering the flow of shunt current through Q22. The direction of change (more or less current) is such as to maintain a fixed voltage value on the -2-volt sense line. As mentioned earlier (paragraph 4-94), the circuit acts like a Zener diode in maintaining a fixed voltage out put. About 5 amperes is passed through Q22.

4-104. CURRENT LIMIT CONTROL. Transistors Q8 and Q9 are normally conducting. When an unusual current drain increases the dc voltage drop across inductor L4 to a specific level (determined by the selected values of R40 through R44), Q8 and Q9 will be biased off. Under this condition, CR35 clamps the unijunction input to a level that is below the trigger point. No pulses are therefore applied to the SCR's, and no further conduction occurs. Both +4.75-volt and -2-volt outputs are thus cut off.

4-105. VOLTAGE LIMITS. There are three separate circuits involved in detecting and acting on out-of-limit dc voltage conditions. These three circuits (-2V limit sense, +4.75V limit sense, and crowbar) are discussed together under the current heading.

4-106. Figure 4-8 illustrates the actions that occur when either the +4.75 or -2 voltages go out of limits. When the +4.75 voltage (applied to Q23/Q24 bases) rises too high, to a level set by R91, Q24 will conduct and activate the power fail circuit (discussed later under paragraph 4-110). Similarly, if the +4.75 voltage drops below a negative limit set by R92, Q23 will conduct and activate the power fail circuit. In the -2V limit sense circuit, if the -2 voltage (applied to the top of the divider), becomes too positive, to a level set by R61, Q14 will conduct and activate the

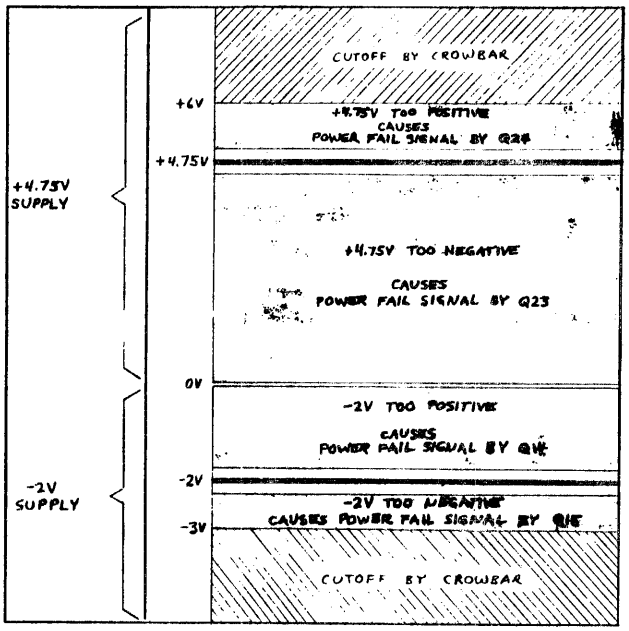


Figure 4-8. Voltage Limit Ranges

power fail circuit. The negative limit sensing circuit uses a normally conducting emitter follower (Q13). When the -2 voltage becomes too negative, Q15 will conduct and activate the power fail circuit.

4-107. If the +4.75 voltage becomes excessively positive (above about +6 volts), or if the -2 voltage becomes excessively negative (more than about -3 volts), the crowbar circuit triggers and cuts off both supplies.

4-108. The crowbar circuit uses an SCR diode (CR80). When the -2-volt level goes more negative than the breakdown level of CR82 (normally an effective open circuit), CR82 causes Q30 (and Q31) to conduct. Or, if the +4.75-volt level goes more positive than the breakdown level of CR81, Q31 will again be caused to conduct. This is because both emitter and base voltages increase together as the +4.75 voltage rises; then CR81 breaks down and holds the base low. When, from either cause, Q31 conducts, SCR diode CR80 is triggered, effectively short-circuiting the +4.75V and -2V supplies together. This protects logic circuits from overvoltage damage. To prevent the rectifiers from delivering any more current to this short circuit, Q29 (which goes into conduction when the SCR triggers) inhibits the sync amplifier. Transistor Q7 is driven into saturation, thus preventing further trigger pulses to SCR rectifiers CR5 and CR6, as discussed in paragraph 4-102.

4-109. LINE FAIL SENSE. Diodes CR60 and CR61 rectify a sample of the transformer secondary output, and the resulting pulsating direct voltage is applied to two RC filters. One filter (R70, C50) has a short time constant, and the other (R71, R72, C51) has a long time constant. The filters are isolated from each otherby CR63. As a result (see figure 4-9), a

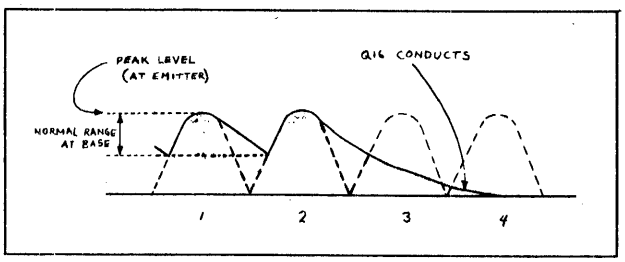


Figure 4-9. Line-Fail Sensing

dc voltage representing the peak value of the rectified ac is present at the emitter of Q16, and a partially filtered waveform is present at the base. Normally (see half-cycle number 1), the exponential decay is not sufficient to cause conduction of Q16 before the next half-cycle restores the C50 charge. If, however, at least two half-cycles are missed (assume ac power is lost at the end of half-cycle number 2), the base voltage will drop to the point where conduction of Q16 will occur. With Q16 conducting, Q17 will also be turned on, thus activating the power fail circuit.

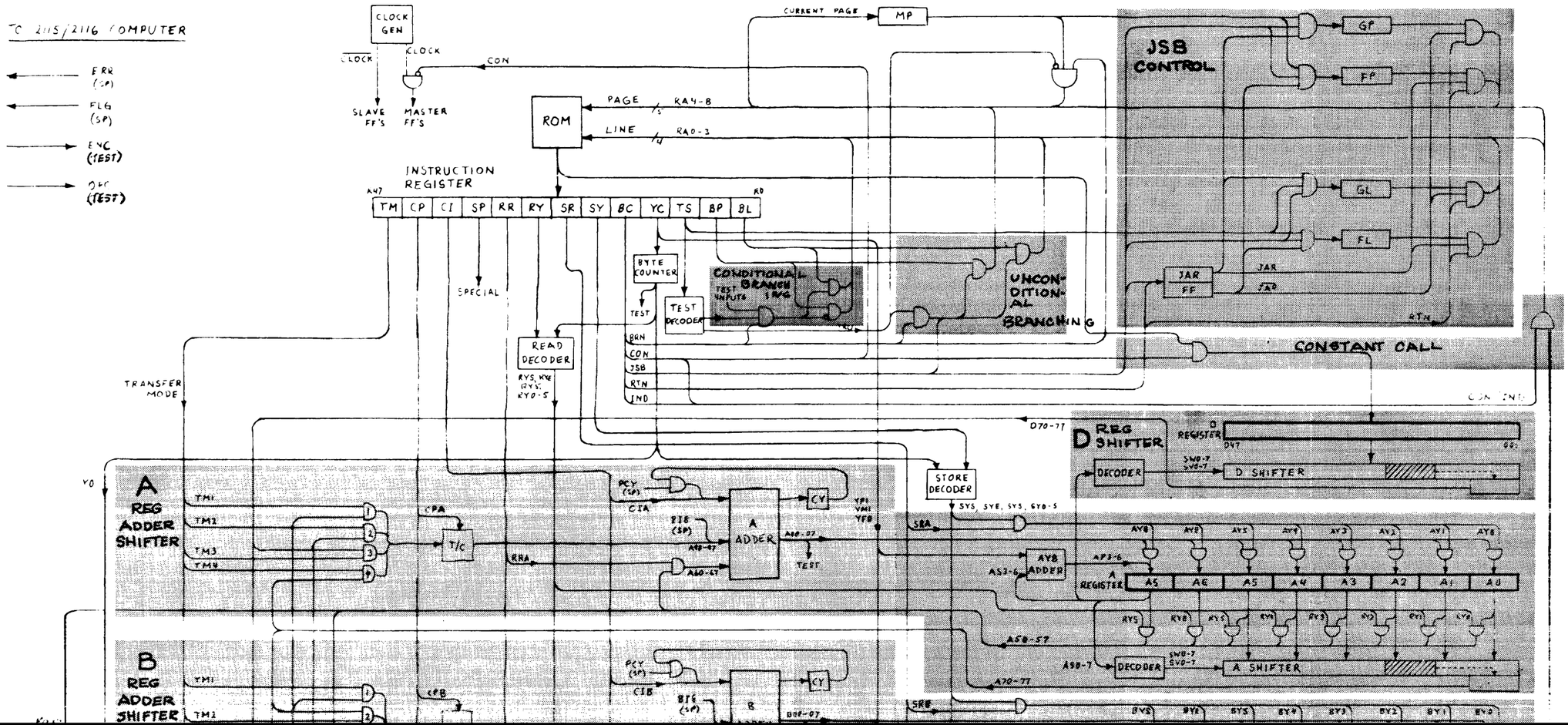
4-110. POWER FAIL. When any of the previously discussed voltage sensing circuits indicate a failure, Q26 is caused to conduct. (Note that four of the sources, Q14, Q15, Q17, and Q23, require inversion by Q25, whereas the Q24 source does not.) The conduction of Q26 in turn causes the other four transistors in the power fail circuit to conduct. The EPF signal (normally low) goes high to initiate a power fail interrupt in the computer. A few milliseconds later, EPO (normally high) goes low; when power is restored, EPO will go high again, initiating a restart sequence in computers which have the restart option installed and enabled.

4-111. +10V SUPPLY. Several circuits in the power supply require a +10-volt operating voltage. To supply this, the transformer secondary is rectified by CR40 and CR41, filtered by C45, and regulated by Q10. The control for Q10 is the differential amplfier consisting of Q11 and Q12. The reference voltage provided by CR42 is compared with a divided sample of the +10V output, and any difference is applied as a correction signal to the base of Q10.

Table 4-3. ROM Instructions

INSTR	DEFINITION
TM1	Transfer Mode 1 C (shifted) to A-adder C (shifted) to B-adder A (shifted) to C-adder
TM2	Transfer Mode 2 B (shifted) to A-adder A (shifted) to B-adder D (shifted) to C-adder
TM3	Transfer Mode 3 D (shifted) to A-adder A to B-adder B to C-adder
TM4	Transfer Mode 4 A (shifted to A-adder B (shifted to B-adder IM* or IL** to C-adder * if Y0 = 0 ** if Y0 = 1
CP ( )	Complement TM input to A, B, or C-adder
CI ( )	Inject Carry into A-, B-, or C-adder
PCY	Propagate Carry into all adders
IRT	Inhibit Result of Test (for diagnostic use only)
BI8	Inject eighth fit (7) into all adders
ERR	Error line to computer I/O
FLG	Flag to computer I/O
RR ( )	Read Register A, B, or C into corresponding adder (add to TM input)
RYY	Read Byte Y of all registers (Y = 0 to 5, determined by byte counter)
RY5	Read Byte 5 of all registers
RYE	Read Exponent Byte of all registers
RYS	Read Shift Control Byte of all registers
SR ( )	Store into Register A, B, or C from output of corresponding adder

INSTR	DEFINITION
SYY	Store into Byte Y, designated by byte counter
SY5	Store into Byte 5
SYE	Store into Exponent Byte
SYS	Store into Shift Control Byte
CON	Load Constant into D-register from ROM location specified by S byte of C-register (plus AD9 bit to specify upper or lower half of ROM); then execute remainder of instruction word. (If CIC is used in the same word as CON, CIC will be inhibited if CS is 24 or higher.)
BRN	Branch conditionally to line (on current page) specified by BL field (test true) or by BP field (test false); or unconditionally to page and line specified by BP and BL fields (test specification = TRU).
IND	Indirect address. Use 8 bits of C-register S byte for next ROM address. (Lower half of ROM, since 9th bit = 0.)
JSB	Jump to Subroutine at address specified by BP and BL fields; TS field saved for return line address; return is always to same page on which JSB was given. One nested JSB level permitted.
RTN	Return to address saved by most recent JSB instruction; see JSB above.
YM1	Y Minus One (decrement byte counter)
YF0	Y Forced to Zero (clear byte counter)
YP1	Y Plus One (Increment Byte Counter)
T ( )	Test specification. Coded to specify 1 of 16 conditions for true/false test. (Refer to table 4-4). Used with branch instruction BRN.
AD9	Ninth Address bit; used only for ROM dump routine to read low half in conjunction with CON.
BP	Branch Page address field; 5 bits.
BL	Branch Line address field; 4 bits.


IC 2115/2116 COMPUTER
ERR (SP)
FLG (SP)
ENC (TEST)
OVF (TEST)
CLOCK GEN
CLOCK
CLOCK
SLAVE FF'S
MASTER FF'S
CON
CURRENT PAGE
MP
ROM
PAGE 5 RA4-8
LINE 4 RA0-3
INSTRUCTION REGISTER
A47
RD
TM CP CI SP RR RY SR SY BC YC TS BP BL
SPECIAL
BYTE COUNTER
TEST
TEST DECODER
CONDITIONAL BRANCH
TEST INPUTS
UNCONDITIONAL BRANCHING
BRN
CON
JSB
RTN
IND
READ DECODER
RYS, RYE RY3, RY0-5
JSB CONTROL
GP
FP
GL
FL
JAR FF
JAR
JAR
RTN
CONSTANT CALL
CON INH
TRANSFER MODE
D70-77
D REG SHIFTER
D REGISTER
D47
D0
SW0-7 SV0-7
DECODER
D SHIFTER
STORE DECODER
SYS, SYE, SY3, SY0-5
A REG ADDER SHIFTER
TM1
TM2
TM3
TM4
CPA
T/C
RRA
PCY (SP)
CIA
PIB (SP)
A0-07
A ADDER
CY
A00-07
TEST
YP1 YM1 YP0
SRA
AY8 ADDER
AS3-6
AP3-6
A REGISTER
A5 A6 A5 A4 A3 A2 A1 A0
AY8 AY6 AY5 AY4 AY3 AY2 AY1 AY0
A50-57
A70-77
A50-7
DECODER
SW0-7 SV0-7
A SHIFTER
B REG ADDER SHIFTER
TM1
TM2
CPB
PCY (SP)
CIB
BIB (SP)
B
B00-07
CY
SRB
BY8 BY6 BY5 BY4 BY3 BY2 BY1 BY0

SECTION V
THEORY OF OPERATION
EXTENDED ARITHMETIC UNIT CONTROLLER

5-1. SCOPE OF SECTION.

5-2. This section of the manual describes theory of operation of the extended arithmetic unit (EAU) controller. The controller portions of the HP 2152A installation are identified in figure 5-1. The EAU controller, as shown, is installed internally in the computer and consists physically of three printed circuit cards.

5-3. The EAU controller provides interfacing and initializing functions for the FPP unit; in addition, it executes 10 of the instructions implemented by the HP 2152A option. These 10 instructions, which provide arithmetic and long shifting functions in the computer registers, do not involve the external FPP unit.

5-4. Therefore, the 10 non-floating-point instructions are independently discussed beginning at paragraph 5-70. Essentially, this latter part of the section is a discussion of basic EAU. As such, it includes a detailed description of the operation cycle counter and many basic EAU signals. Thus it may be helpful to refer ahead to this part of the section whenever the need occurs.

5-5. INTRODUCTION.

5-6. The EAU controller performs two basic functions: to transfer data between memory and the FPP unit, and to execute integer arithmetic and long shifts in the computer CPU. Figure 5-2 illustrates these functions in simplified form.

5-7. In the figure, the shaded area identifies the three cards of the controller. The EAU timing and logic cards are installed in two dedicated slots of the CPU section, and the EAU interface card occupies any one of the I/O interface card slots in the I/O section.

5-8. The operation begins when a macro group instruction is read out of memory into the T-register. The actions that follow depend on the type of instruction that is read. Decoding of the instruction is therefore the first operation to occur. Full decoding is accomplished in two or three distinct steps; figure 5-3 will help to visualize the decoding process.

5-9. The first step in decoding the instruction occurs in the CPU instruction decoder. Here, the macro group of instructions is identified by decoding bits 15, 14, 13, 12, and 10 of the T-register (coded 1-0-0-0--0, respectively). The macro group includes all EAU and floating point instructions. Then, a MAC signal to the operation decoder in the EAU controller enables the second step in the decoding process.

5-10. Bits 11, 9, and 8 of the T-register are examined to determine which type, or sub-group, is coded in these three bits. These bits select one of eight types of operations, according to the octal value represented. The specific breakdown of

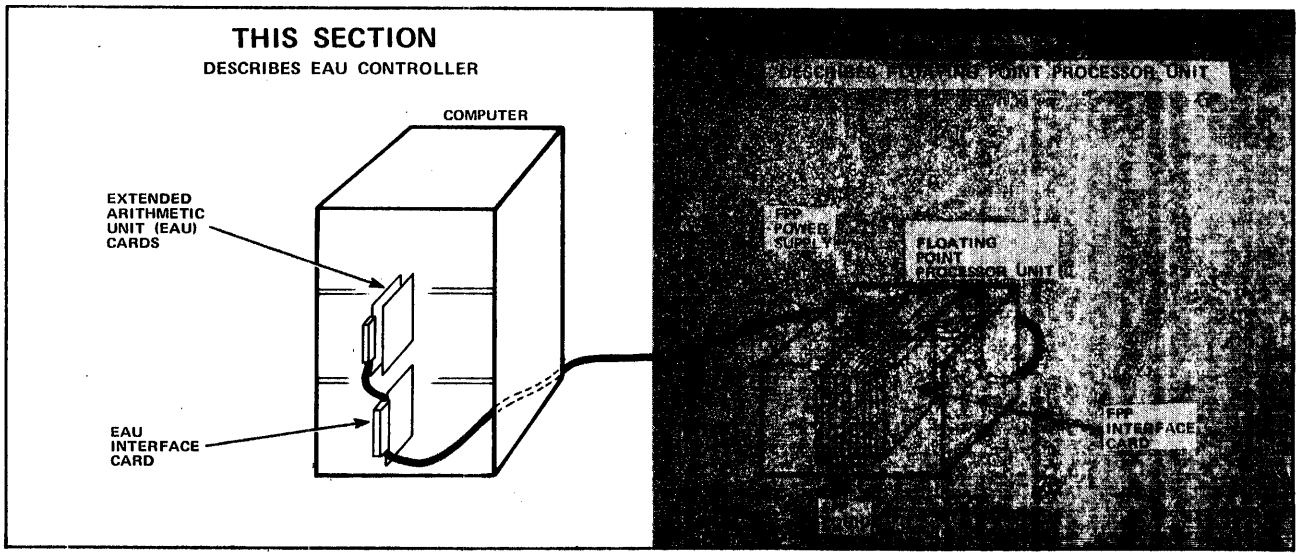


Figure 5-1. Controller Section

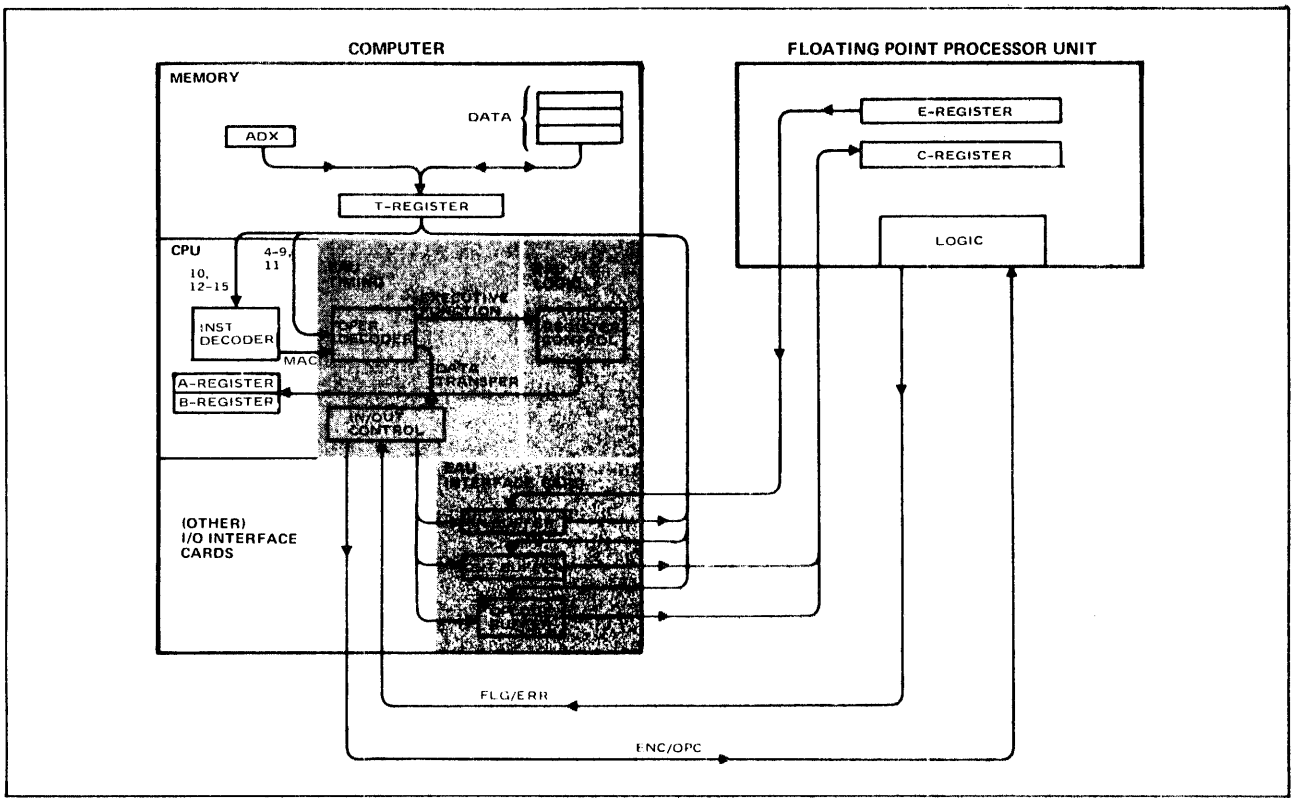


Figure 5-2. Basic EAU Controller Operations

operation groups is shown in table 5-1. In general, if the octal value is 7, 6, or 3, the in/out control logic causes bits 0 through 7 to be stored in the op-code buffer. This 8-bit opcode will later be sent to the FPP unit, where a third decoding step takes place. This third step is accomplished by the entry routine in the FPP ROM programs.

Table 5-1. Instruction Grouping

OCTAL VALUE OF BITS 11 - 9 - 8	INSTRUCTION OR GROUP
0	MPY (Multiply), or left shifts and rotates
1	DIV (Divide)
2	RT (Right) shifts and rotates
3	TST (Triple Store) group
4	DLD (Double Load)
5	DST (Double Store)
6	FPF (Floating Point Function) group
7	TLD (Triple Load) group

5-11. However, if the octal value decoded by the operation decoder is one of the numbers from 0 through 5, the instruction is not of the floating point type. Therefore, the instruction will be executed by the EAU controller, and the remaining instruction bits, 0 through 7, are then further decoded by the operation decoder.

5-12. After decoding of the instruction, the succeeding operations depend on whether the basic function of the instruction is to transfer data or to execute some function in the CPU. These two functions include the following instructions or groups.

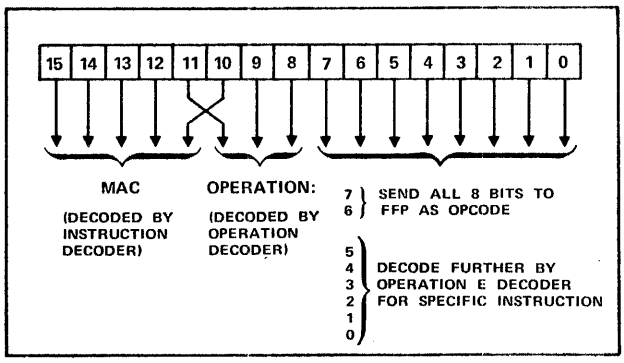


Figure 5-3. Instruction Decoding Process

Data Transfer:

3 TST
6 FPF
7 TLD

Execute Function:

0 MPY, left shifts
1 DIV
2 Right shifts
4 DLD
5 DST

5-13. The following paragraphs discuss both basic functions of the EAU controller.

5-14. DATA TRANSFER. There are three types of transfers to and from the floating point processor unit. These are: storing in memory a triple-length word from the FPP unit (TST group), sending an 8-bit opcode to the FPP unit (FPF group), and sending both a triple-length word and an opcode to the FPP unit (TLD group).

5-15. Referring to figure 5-2, note that the data transfer line enables the EAU in/out control logic. This logic controls the loading and reading of the three buffers shown on the EAU interface card and sends or receives the four interface control signals (FLG, ENC, OPC, ERR).

5-16. To store data from the FPP unit (TST group), the in/out control logic sends an ENC (Encode) signal, and the FPP unit responds by sending 16 bits of data from its B-register with a FLG (Flag) signal. The in/out control logic loads this data into the input buffer, then sends the data to memory while issuing another ENC. This process repeats for a second and third time, until the entire 48-bit contents of the FPP B-register are stored in memory.

5-17. To send an opcode to the FPP unit, the eight least significant bits of the instruction word (in the T-register) are loaded by in/out control into the opcode buffer. The opcode is then read onto the interface data lines, and an OPC (Opcode) signal is issued to the FPP unit. The FPP unit loads the opcode into its C-register, executes the function specified by the opcode, then returns a FLG signal to indicate completion.

5-18. To send data plus an opcode to the FPP unit, in/out control first loads the opcode from the T-register into the opcode buffer, then fetches a data word from memory and loads it into the output buffer. Then the data is read onto the interface data lines, and an ENC (Encode) signal is issued to the FPP unit. The FPP unit loads the 16 bits of data into its C-register and returns a FLG signal to ask for the next word. This process repeats for a second and third time until the entire 48-bit C-register is filled with the three words from memory. Then the opcode is read onto the interface data lines and an OPC signal causes the FPP unit to load the opcode into the shift byte of the C-register. After the FPP unit executes the function specified by the opcode, it returns a final FLG signal to indicate completion.

5-19. EXECUTE FUNCTION. As shown in figure 5-2, some of the instructions decoded by the operation decoder (such as MPY, DIV, etc.) can cause EAU timing to directly activate the register control logic. Primarily, as shown, the computer A- and B-registers are manipulated. However, all five CPU registers can be manipulated, and data can be fetched from memory, in order to execute the arithmetic functions. Note that this is entirely an internal operation; the floating point processor unit, the EAU interface card, and the in/out control are not involved.

NOTE

1. Pages 5-4 through 5-13 describe in detail the operation of the part of the EAU controller which is responsible for data transfers to and from the floating point processor unit. These descriptions are presented in time sequence beginning with the phase 1 operation, in which the instruction is initially fetched from memory.
2. The flowcharts include references to the logic diagrams. These references (e.g., U42C) refer to a specific gate responsible for the action in the adjacent flowchart symbol.
3. All logic is positive-true. The high (or true) state ranges from +1.25 to +2.5 volts, and the low (or false) state ranges from -0.5 to +0.5 volts.

5-20. PHASE 1 OPERATION

5-21. The EAU controller operation begins during phase 1, on the same cycle that reads the instruction out of memory. The memory-read cycle occurs from about the middle of T0 to the middle of T2. While this is occurring, at T1, the operation cycle counter is cleared. By T3, the instruction is in the T-register, so a TIN1 signal is generated to store opcode bits TR0 through TR7 on the interface card (see figure 7-6) in case the opcode is needed by the FPP. At the same time, T3, a RESET signal is generated to clear the shift, HI-LO, and IO1/IO2 flip-flops. The cycle flip-flop is also cleared since neither of the inputs clocked by T3 is true at this time.

5-22. If the instruction decoded by the CPU is an EAU class instruction (MAC signal true at pin 62 of the board), the instruction group is decoded and stored in the operation decoder. The eleven groups are identified in table 5-2, which shows the T-register bits required to identify the group. As mentioned on the preceding page, the EAU-only groups (first seven) are described later in the section (paragraph 5-58). This part of the section will discuss in detail the four FPP groups: TST, STD, TLD, and FPF.

5-23. When the instruction group is decoded at T3, a multiple branch occurs as shown in the flowchart. First it is determined whether or not the instruction is a shift or rotate. If it is, the number of shifts required is loaded into the operation cycle counter; T-register bits 0-3 give the number of shifts (see instruction coding table in section III.) Otherwise the counter is set to 5 (decimal) for all memory operand routines, or is simply not used (FPF group only). Also, for memory operand routines, the P123 signal is generated. For the rest of the cycle, this signal is redundant since phase 1 is still in progress, but it will enable reading the address word from memory on the next cycle. (Remember that all EAU instructions using a memory operand occupy two words in core: an instruction word, and an address word that points to the operand location.) From this point (T3) the groups are considered separately. (In all cases, the Overflow bit is cleared at T5.)

5-24. FPF GROUP. From the standpoint of inter-unit transfers, the only operation required for the FPF group is to send the opcode to the FPP unit. (The opcode will tell the FPP what function to perform on data that is already present in the FPP unit.) This is done at T4 by setting the IO1 flip-flop if the FPP Flag is high (ready). This gives an IO2/IO1 code of 01, which generates an OPC signal. This OPC signal reads out the stored opcode (see figure 7-6) and is sent to the FPP unit. When the FPP unit receives OPC, it loads the opcode and begins its programmed operation. Meanwhile, at T5, the Overflow flip-flop is cleared in anticipation of a possible error. At T5 the CPU is disabled by an IIR signal, which remains high until exit. For succeeding operations after phase 1, refer to page 5-10.

5-25. TST/STD GROUPS. At T4, if the FPP is ready (Flag high), the IO code is set to 01. This is done by setting the IO1 flip-flop. If STD is true, an OPC signal is generated, which remains true until the FPP Flag goes low; this tells the FPP unit to load the opcode. If TST is true, an ENC (Encode) signal is generated; this tells the FPP to send the first 16 bits of the FPP B-register. Since the IO code is no longer 00, TIN4 and TIN5 are true, thus enabling the input buffer. (See figure 7-6.) For succeeding operations after phase 1, refer to page 5-6.

5-26. TLD GROUP. In the triple-load operation the opcode is sent to the FPP last (after all three data words). Since no data is fetched from memory during the first two cycles, no OPC or ENC is sent during phase 1. The only operations are to disable the CPU at T7S and increment the counter to 6 at the end of T7S. For succeeding operations after phase 1, refer to page 5-8.

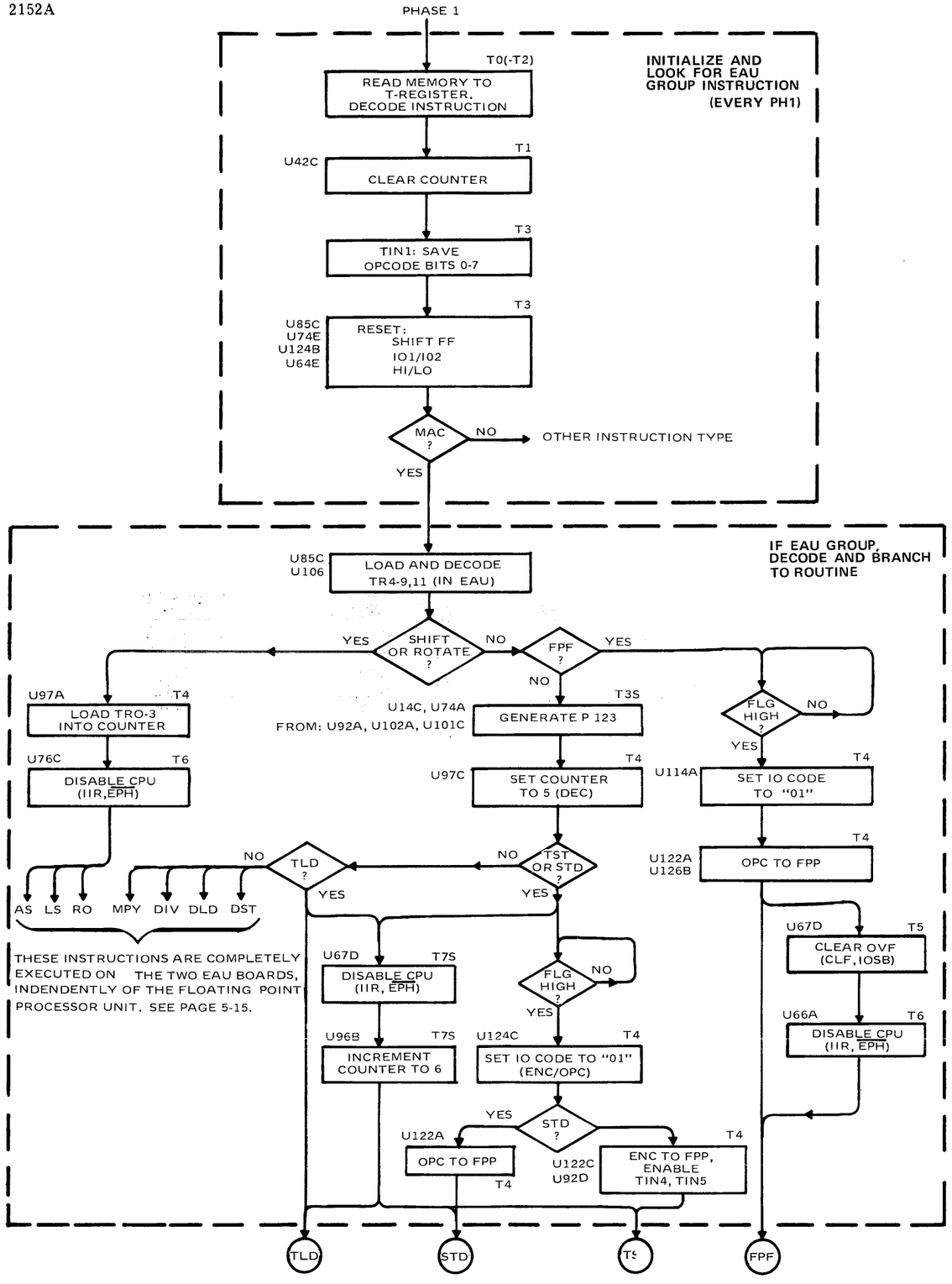


Figure 5-4. Phase 1 Flowchart

Table 5-2. Instruction Group Coding

INSTRUCTION GROUPS	IDENTIFYING BITS	
	TR11, 9, 8 (Octal)	ADDITIONAL
AS (ARITHMETIC SHIFT)	0 or 2	TR4 = 1
ASR Arithmetic Shift Right ASL Arithmetic Shift Left		
LS (LOGICAL SHIFT)	0 or 2	TR5 = 1
LSR Logical Shift Right LSL Logical Shift Left		
RO (ROTATE)	0 or 2	TR6 = 1
RRR Rotate Right RRL Rotate Left		
MPY (MULTIPLY)	0	TR7 = 1
DIV (DIVIDE)	1	
DLD (DOUBLE LOAD)	4	TR7 = 1
DST (DOUBLE STORE)	5	
STD (STORE DOUBLE)	3	TR1 = 1
STD Store Double word, converted from Extended Floating Point		
TST (TRIPLE STORE)	3	TR1 = 0
STF Store Floating point word STX Store Extended floating point		
TLD (TRIPLE LOAD)	7	
LDD Load Double LDF Load Floating point LDX Load Extended floating point ADF Add Floating point ADX Add Extended floating point SBF Subtract Floating point SBX Subtract Extended floating point MPF Multiply Floating point MPX Multiply Extended floating point DVF Divide Floating point DVX Divide Extended floating point		
FPF (FLOATING POINT FUNCTION)	6	
ABX Absolute value ENX Entier CMX Complement CSX Cosine SNX Sine TNX Tangent ATX Arctangent HCX Hyperbolic Cosine HSX Hyperbolic Sine HTX Hyperbolic Tangent AHT Archyperbolic Tangent EXX Exponential LNX Natural Logarithm SRX Square Root		

TIME PERIODS

MEMORY CYCLE

TINI

IIR

MEMORY-OPERAND ROUTINES ONLY

COUNT

P123

FPF, TST, STD ROUTINES ONLY

ENC/OPC

Table 5-2. Instruction Group Coding

INSTRUCTION GROUPS	IDENTIFYING BITS	
	TR11, 9, 8 (Octal)	ADDITIONAL
AS (ARITHMETIC SHIFT)	0 or 2	TR4 = 1
ASR Arithmetic Shift Right		
ASL Arithmetic Shift Left		
LS (LOGICAL SHIFT)	0 or 2	TR5 = 1
LSR Logical Shift Right		
LSL Logical Shift Left		
RO (ROTATE)	0 or 2	TR6 = 1
RRR Rotate Right		
RRL Rotate Left		
MPY (MULTIPLY)	0	TR7 = 1
DIV (DIVIDE)	1	
DLD (DOUBLE LOAD)	4	TR7 = 1
DST (DOUBLE STORE)	5	
STD (STORE DOUBLE)	3	TR1 = 1
STD Store Double word, converted from Extended Floating Point		
TST (TRIPLE STORE)	3	TR1 = 0
STF Store Floating point word		
STX Store Extended floating point		
TLD (TRIPLE LOAD)	7	
LDD Load Double		
LDF Load Floating point		
LDX Load Extended floating point		
ADF Add Floating point		
ADX Add Extended floating point		
SBF Subtract Floating point		
SBX Subtract Extended floating point		
MPF Multiply Floating point		
MPX Multiply Extended floating point		
DVF Divide Floating point		
DVX Divide Extended floating point		
FPF (FLOATING POINT FUNCTION)	6	
ABX Absolute value		
ENX Entier		
CMX Complement		
CSX Cosine		
SNX Sine		
TNX Tangent		
ATX Arctangent		
HCX Hyperbolic Cosine		
HSX Hyperbolic Sine		
HTX Hyperbolic Tangent		
AHT Archyperbolic Tangent		
EXX Exponential		
LNX Natural Logarithm		
SRX Square Root		

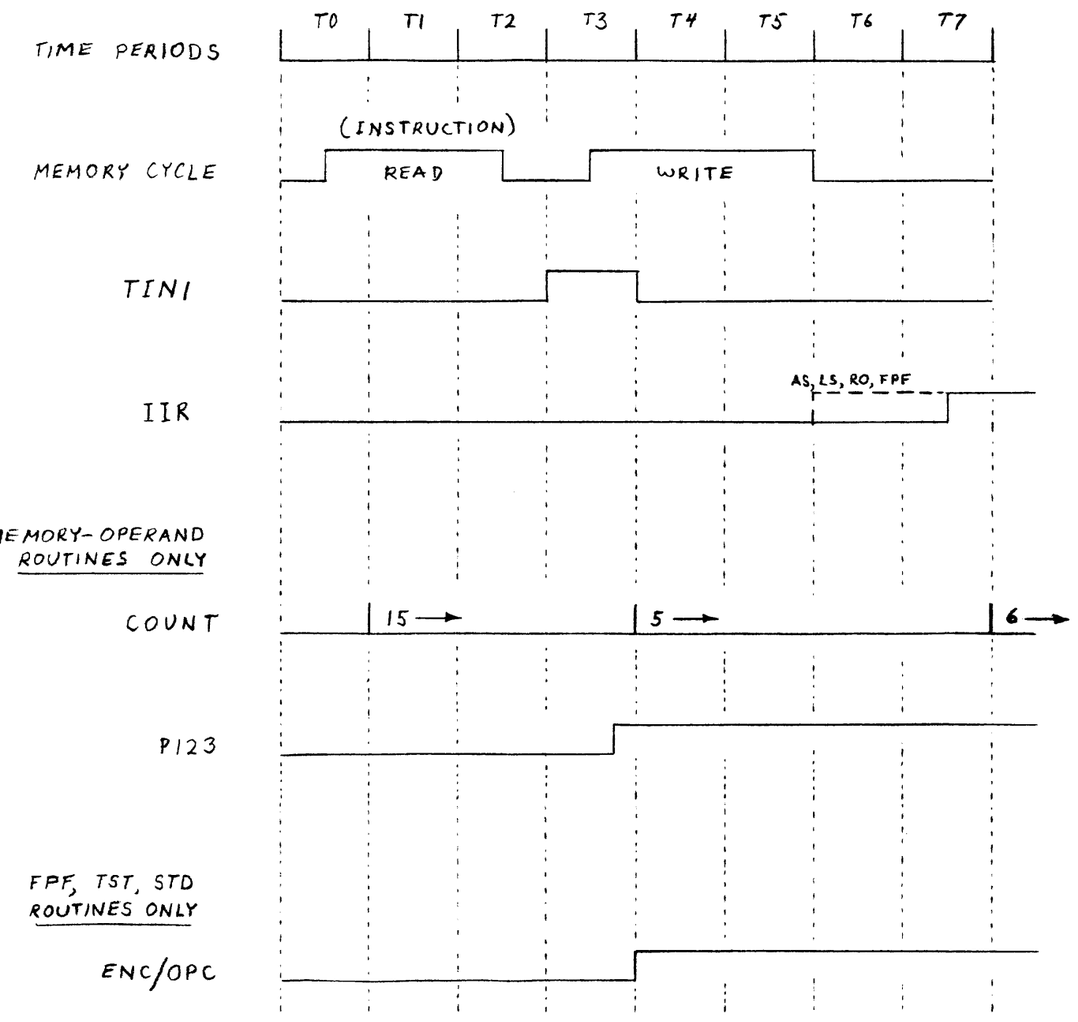


Figure 5-5. Phase 1 Timing

5-27. TST OPERATION.

5-28. There are three store instructions in the 2152A repertoire. These instructions were defined earlier in this manual, but for the purposes of this discussion they may be defined as follows.

a. STX (Store Extended word): get triple-length word from FPP unit and store in three consecutive memory locations.

b. STF (Store Floating-point word): get triple-length word from FPP unit, remove bits 8 through 23, and store remaining double-length word in two consecutive memory locations.

c. STD (Store Double word): tell FPP unit to reformat its data word to double-length integer; then get bits 16 through 47 and store in two consecutive memory locations.

5-29. Except for differences due to formatting, the basic storing process is the same for all three instructions. The following paragraphs initially describe the operation for STX; then paragraphs 5-38 and 5-39 discuss the formatting differences for STF. Since STD requires additional time to send an opcode to the FPP unit and then wait for the FPP to perform its format conversion, the timing for STD is considerably different. (Refer to following page.)

5-30. For STX, then, the timing is as shown in figure 5-7. The first word is stored during cycles 2 and 3, the second word during cycles 4 and 5, and the third during cycles 6, 7, and 8.

5-31. During cycle 2 (the cycle that immediately follows phase 1), the EAU controller is waiting for the FPP Flag to go low, meaning that the FPP unit has received the ENC signal issued in phase 1, and is now busy getting the first 16 bits ready for transfer. When the flag does drop (asynchronously, during T0), the IO code is changed from 01 to 11.

5-32. Meanwhile, while waiting for the low Flag signal, EAU proceeds to obtain the address word of the double-word instruction. This operand address will tell EAU where to store the data when FPP sends it. During T0 through T2, the P123 signal causes memory to read the address word into the T-register. Then, at T7, the MD2 signal (which is high during the count of 6) causes the T-register contents to be transferred to the M-register. (Referring to the EAU logic card diagram, figure 7-4, note that in the read T block, RTSB is enabled at T6T7; in the add function block, ADF is enabled at T6T7; and in the store in M block, SWSM is enabled at T7S. These three actions accomplish the T-to-M transfer.) Unless bit 15 of the address indicates indirect addressing, the count now increments to 7.

5-33. By cycle 3, the FPP unit has returned a high Flag signal, indicating that the first 16 bits of data is now on the input lines. At T1, the IO code changes to 10, which causes an IOIO signal to load the data into the input buffer. (See Figure 7-6.) The TIN4 and TIN5 signals, which control the loading of the buffers, are both high at this time, allowing all 16 bits to be loaded. Now the word must be transferred from the buffer to memory. (Note that an ISG signal prevents reading memory to the T-register during store cycles 3, 5, and 8.) The storing process is as follows.

5-34. At T2 an IOI signal strobes the buffer contents (IOBI lines) onto the S-bus in the CPU. A DS34 signal, generated by IOIO, causes EOFB and STBT signals (see figure 7-4) to load the S-bus into the T-register. Then the CPU proceeds to write the T-register into memory while the count and operand address are incremented. The count increments to 8 on the first loop, and incrementing of the M-register is enabled by HI (set at T3) and the TIN4 signal, which generate TS34. (TS34 generates RMSB, RB0, ADF, and SWSM on the EAU logic card; see figure 7-4. These signals increment the M-register.)

5-35. Also, while the CPU is writing the current word into memory, EAU sends a new ENC at T4 to get the next word from the FPP unit. When the Flag goes low (meaning it has received ENC and is preparing the next word), the IO code is set to 11.

5-36. By the next T1, the FPP unit normally has its data ready and sends a high Flag signal. This initiates a repeat of the entire loading-storing process (to paragraph 5-33).

5-37. During the last cycle, the count increments to 10, ending ISG. Count 10 also sets the Exit flip-flop at T4. At T5 the computer A-register is cleared; and at T6T7 an Exit signal initiates the exit sequence: the P-register is incremented and stored in P and M (see figure 7-4), RESET clears the Operation Decoder flip-flops (which in turn ends P123), and the CPU is enabled again (EPH flip-flop set).

5-38. The STF operation inhibits those signals indicated by dashed lines in the timing diagram. As mentioned previously, the intent is to remove bits 8 through 23 of the 48-bit FPP word and to store the remaining 32 bits as two words. In order to inhibit bits 16 through 23 from being loaded into the input buffer, the TIN4 signal is low. As shown in the timing diagram this results in TIN4 being low when the second data word arrives from the FPP unit (when the Flag signal goes high during cycle 4).

5-39. Similarly, the TIN5 signal is low in cycle 5 in order to inhibit bits 8 through 15 of the third word, which arrives during cycle 7. The word in the input buffer therefore consists of bits 24 through 31 and 0 through 7 of the original 48-bit FPP word. To prevent the M-register from incrementing twice, before the word is ready, TS34 is inhibited during cycle 3. (Note SWSM does not occur.) Consequently, the location of the first word will continue to be addressed during cycles 4 and 5. The ISG signal must therefore be inhibited (see timing diagram) so that normal read/write memory cycles will occur; otherwise the contents would be destroyed by clear/write cycles. The combined word in the Input Buffer is stored during cycle 8.

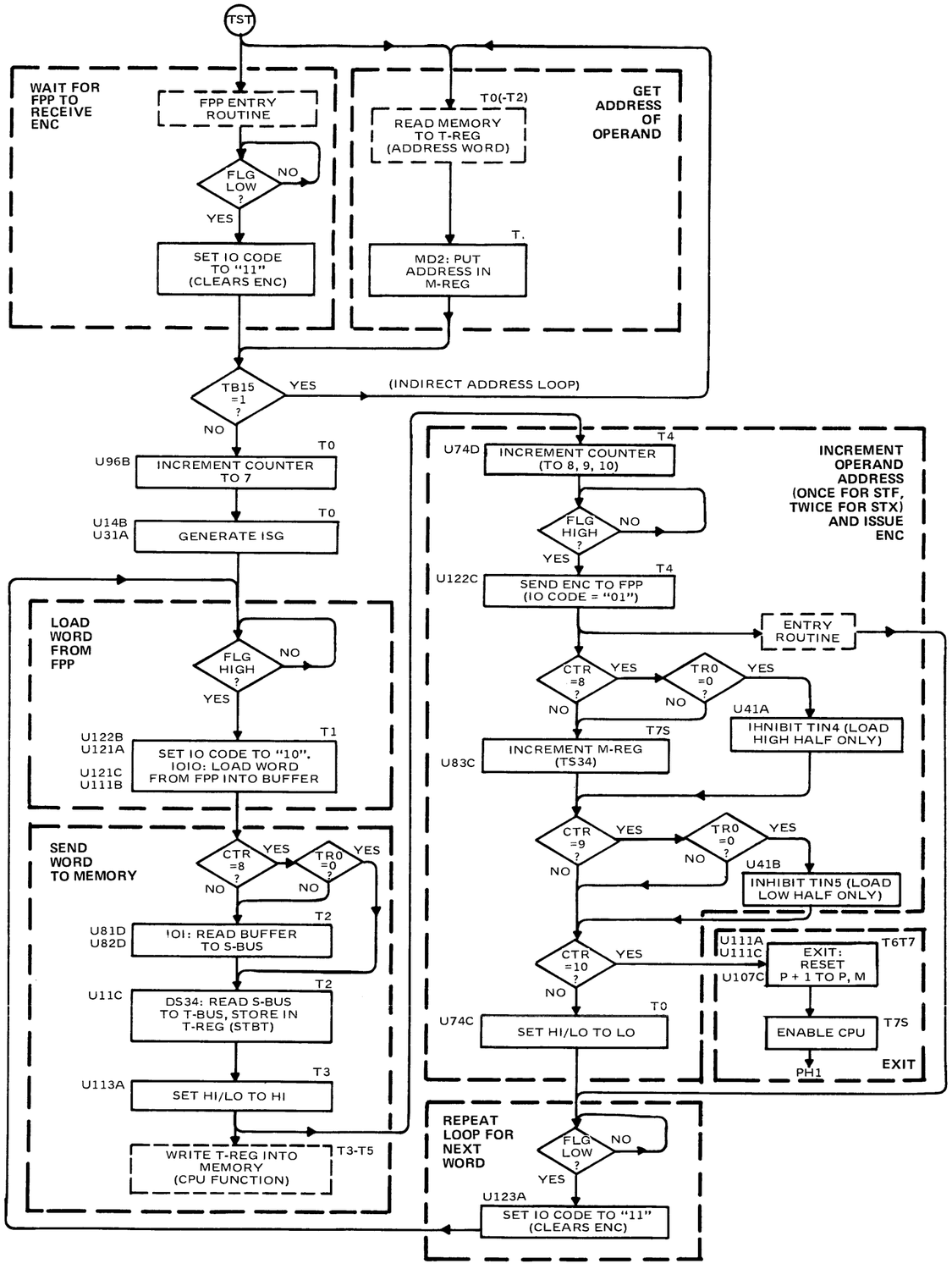
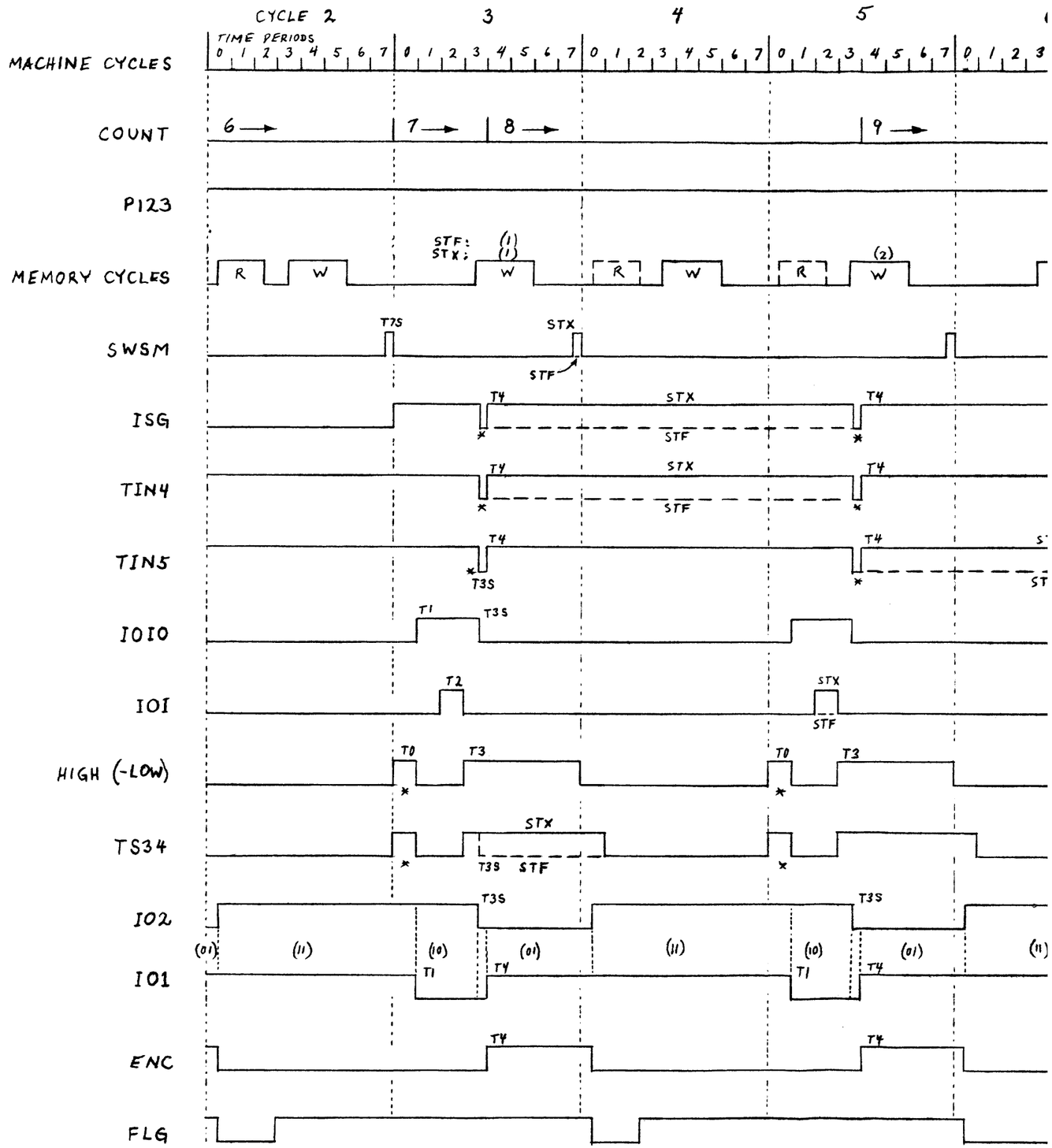


Figure 5-6. TST Flowchart



* Insignificant state changes

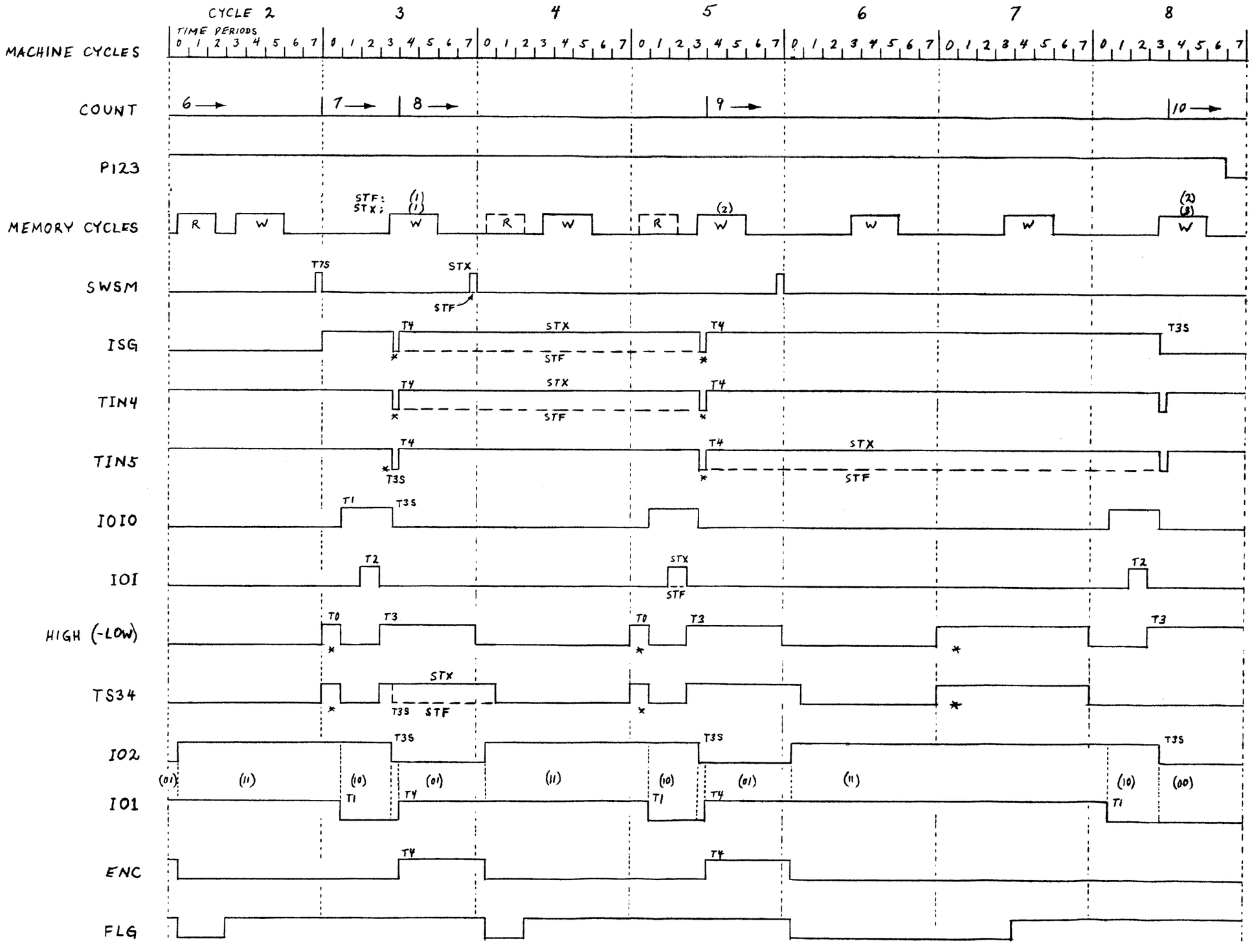


Figure 5-7. TST Timing

5-40. STD OPERATION.

5-41. Paragraphs 5-28 and 5-29 on the preceding page defined the differences between the three store instructions. Essentially, STD (store double word) differs from the other two in format and in the fact that an opcode is sent to the FPP unit. The format is double-word integer, converted from triple-word floating point. This conversion is too complex to be accomplished by the EAU controller, as in the case of STF. Therefore, the FPP is given the opcode for STD, and it performs the conversion before any data is transferred. The opcode has already been sent to the FPP unit. (Refer to page 5-4, phase 1 operation.)

5-42. When the STD routine begins, in cycle 2, the EAU controller is waiting for the FPP Flag to go low, meaning that the FPP unit has received the OPC command and is busy converting the data. When the Flag does drop (asynchronously, during T0), the IO code is changed from 01 to 00.

5-43. This change of IO code represents a significant difference from the TST operation, which changed the IO code to 11 at this point. The 00 state for STD means that the next action will be another command (ENC), rather than the loading of data. To explain the significance of the IO codes, refer to table 5-3. The 00 code is a ready state, which requires a high Flag signal for any change; the only change that can be made is to the 01 code. The 01 code issues either an OPC or ENC command, which terminates when the Flag goes low. The low Flag either resets the code to 00 (to repeat the process for another command), or advances the code to 11 (to prepare for loading data). If the code is now 11, the EAU controller is waiting for a high Flag, indicating readiness to load data. When the Flag signal goes high, the code changes to 10, which causes an IOIO signal to load data into either the input or the output buffer, depending on the routine. The IOIO signal lasts for about three time periods (approximately 550 nanoseconds), and then the IO code resets to 00. Exit occurs during the count of 10, and the IO code can be either 11 or 10.

Table 5-3. IO Codes

IO CODE	SIGNIFICANCE
00	Ready. Next state 01.
01	Issues OPC or ENC. EAU controller waiting for low FLG. Next state 11 or 00.
11	Waiting for high FLG (or count 10: Exit). Next state 10.
10	Load Buffer (IOIO), (or, count 10: Exit). Next state 00.

5-44. Returning to the STD sequence of operations, the EAU controller proceeds to obtain the address word of the instruction, while the FPP unit begins its conversion routine. The address word is read out of memory into the T-register during T0 through T2 of cycle 2, and transferred to the M-register during T7 by the MD2 signal. If bit 15 of the address word indicates indirect addressing, one or more additional memory cycles may be necessary to obtain a direct address. Then the count increments to 7. The count of 7 enables the TST line (see logic diagram, figure 7-2), since most of the operations after this point are common to the TST operation.

5-45. After about ten cycles (160 microseconds) the FPP Flag goes high, indicating that the format conversion has been completed. Now the EAU controller must tell the FPP unit to read out the first 16-bit word. The IO code is changed to 01 at the next T4, causing an ENC signal to be sent to the FPP unit. This occurs during cycle 3. Actually about 12 cycles have elapsed; the cycle numbers are for identification purposes. Also occurring at T4 is the enabling of ISG, TIN4 and TIN5. The ISG signal prevents reading of memory data to the T-register during the memory-storing cycles, and TIN4 and TIN5 enable loading of FPP data into the input buffer.

5-46. When the Flag goes low, acknowledging the ENC command, ENC terminates and the IO code changes to 11. The EAU controller is now waiting for a high Flag. When this signal arrives, it indicates that data is on the interface lines. At the next T1, the IO code changes to 10; causing the IOIO signal to load the data into the input buffer.

5-47. Then, at T2, the combination of IOI and DS34 transfers the word from the input buffer to the T-register. During T3 through T5 the T-register contents is written into memory, while (at T3) the HI/LO flip-flop toggles to the HI state. The combination of HI and TIN4 causes TS34 to enable incrementing of the M-register. This occurs once, at T7S of cycle 5.

5-48. Meanwhile, at T4 of cycle 5, the count increments to 8, and a new ENC is sent to the FPP unit. This causes the next 16 bits to be loaded and stored, as in paragraphs 5-45 through 5-47. Then the routine repeats again for the third time.

5-49. However, on the third loop, when the count is 9, the signals which effect the data transfer are inhibited, so no data is stored. Also, the M-register is not incremented, so that cycles 8 through 10 merely read and write the location containing the second transferred word. The signals which are inhibited are: TIN4, TIN5, ISG, TS34, and IOI. This no-store cycle accomplishes the requirement of the STD instruction, to store only the first two words from the FPP unit.

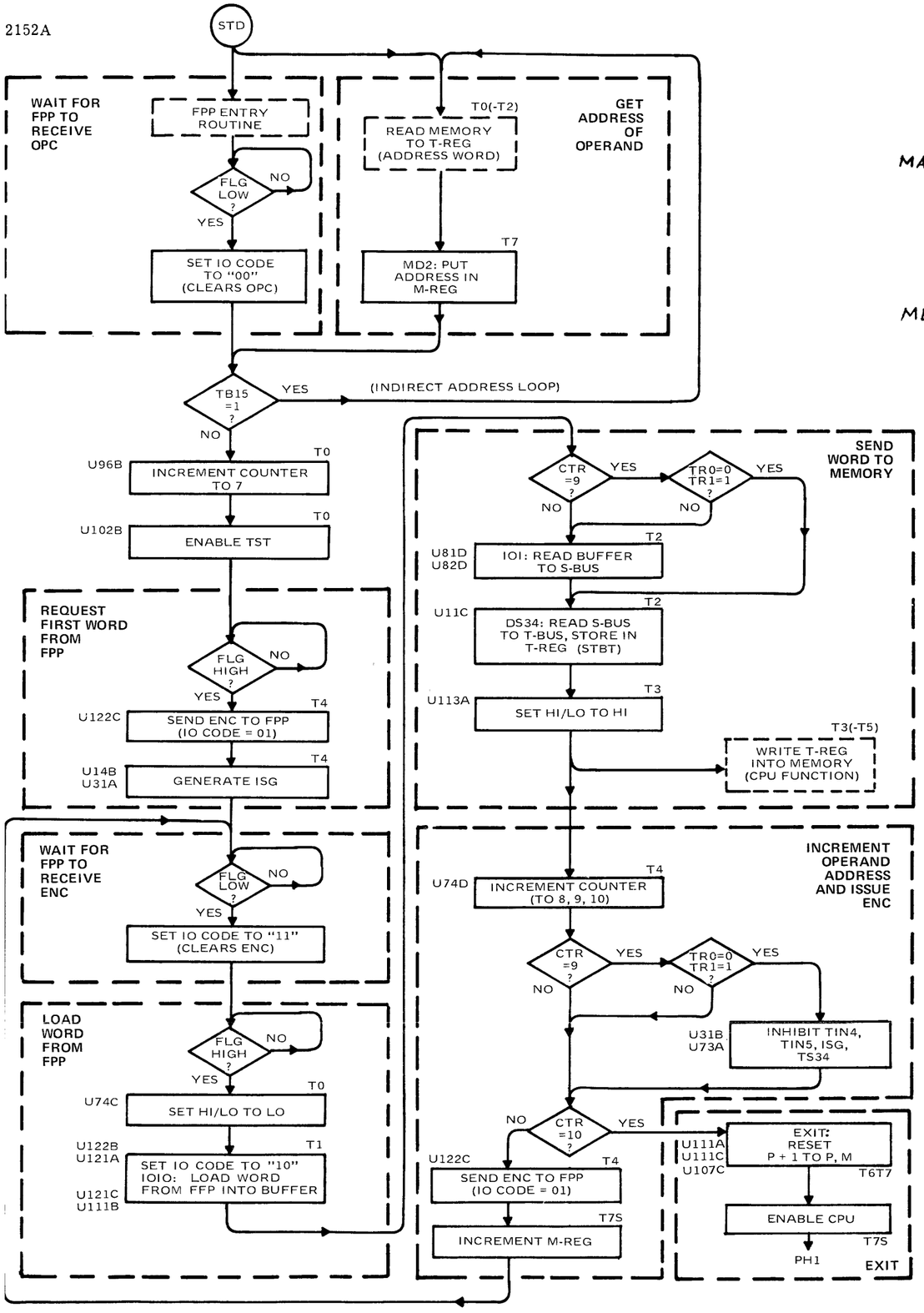
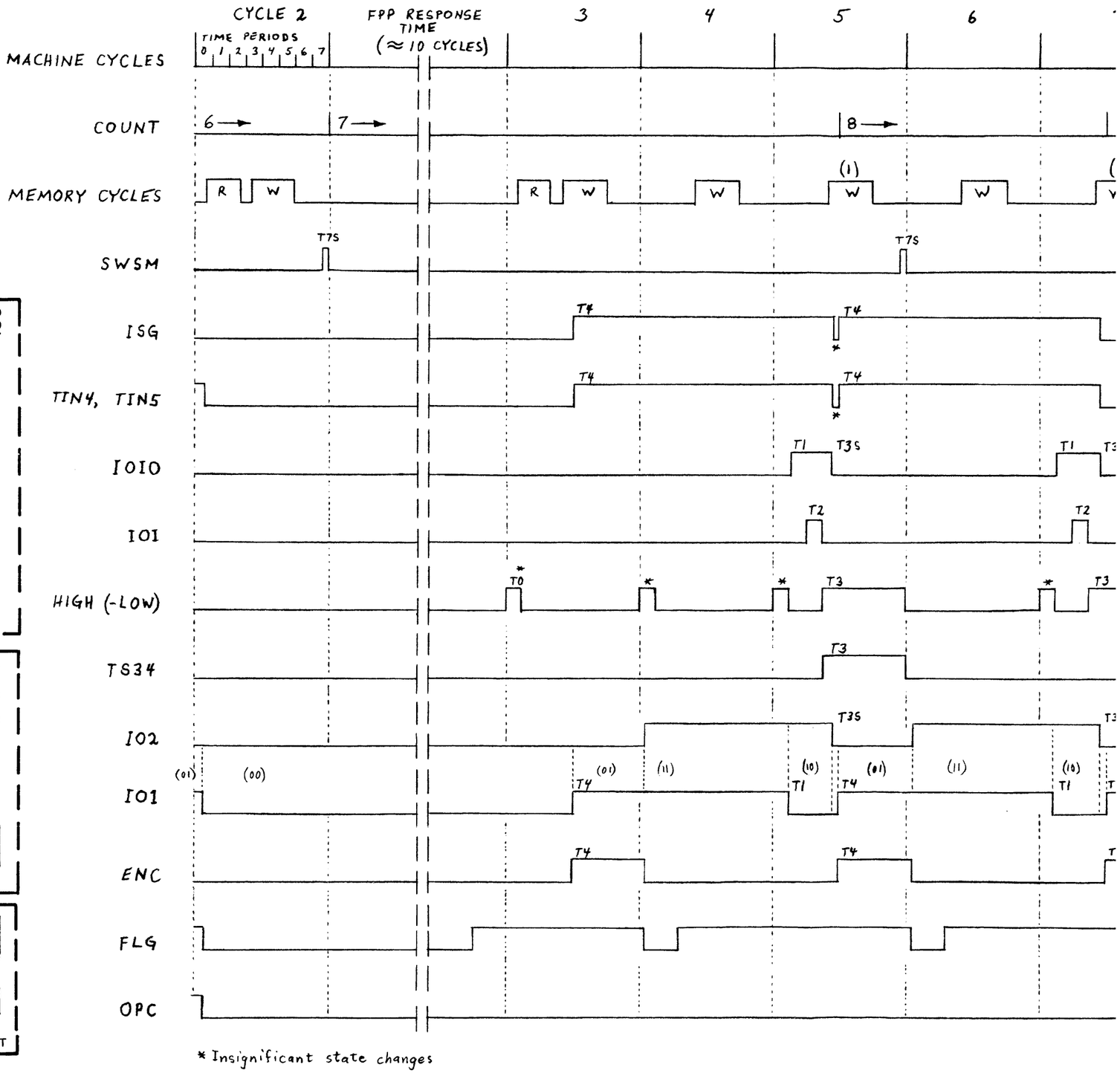


Figure 5-8. STD Flowchart



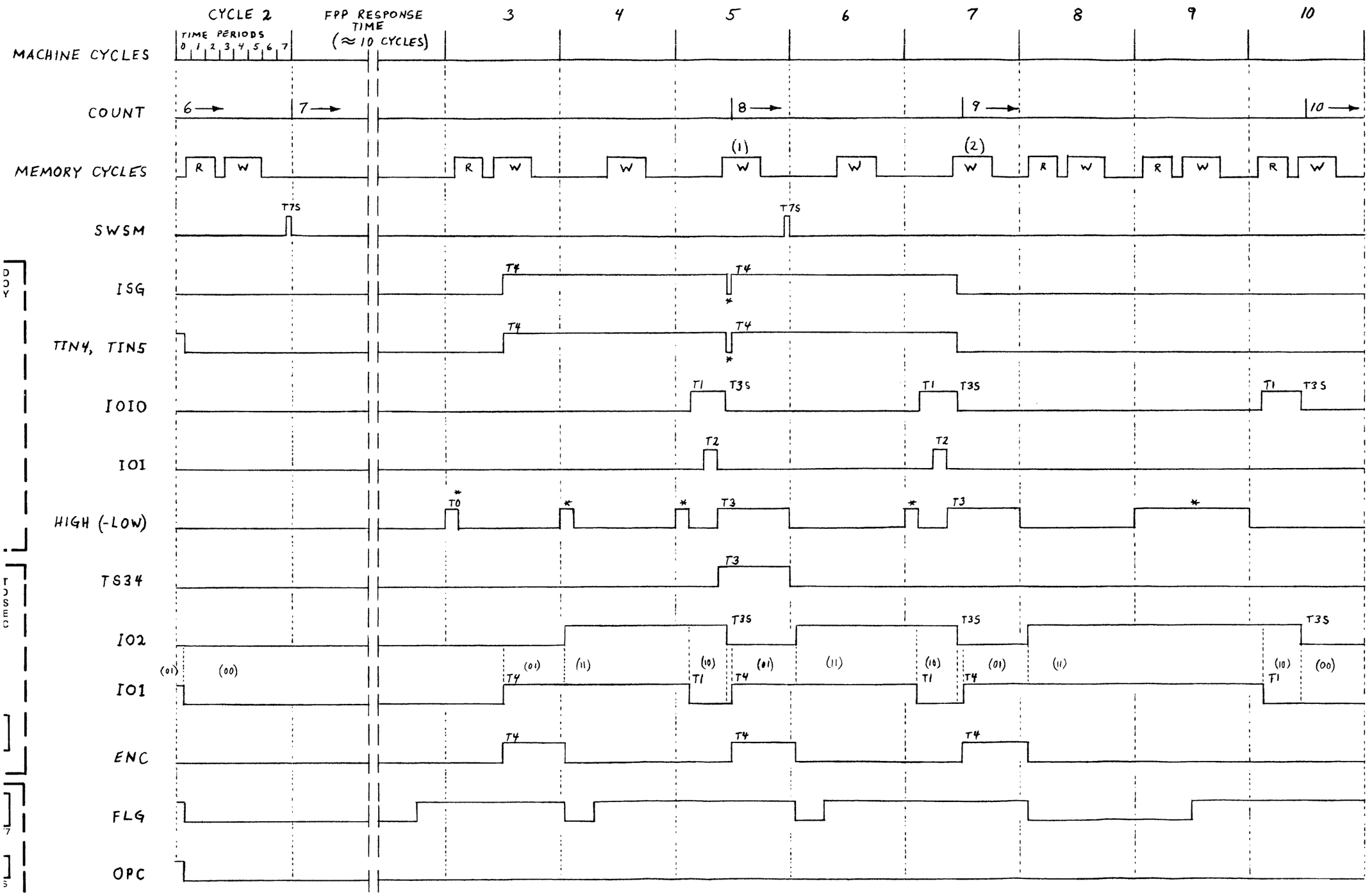


Figure 5-9. STD Timing

5-50. TLD OPERATION.

5-51. The TLD (triple load) routine fetches three words from memory and sends each one, as it is received, to the FPP unit. There are two variations unique to the LDF (load floating point) and LDD (load double integer) instructions. Both of these instructions have a two-word operand in memory, instead of three. For LDF the second word is fetched twice and reformatted, with 16 zeros, to convert the data to the triple-length format. (Format conversion is illustrated in section III.) For LDD a third word is actually fetched from the next memory location and sent to the FPP unit; however, the FPP unit will disregard this illegal word when it enters its LDD routine.

5-52. Following completion of the TLD routine, the instruction opcode is sent to the FPP unit by entering the FPF routine (floating point function, next page). The opcode will tell the FPP unit what to do with the three words it has received. For the three load instructions (LDX, LDF, LDD) this will consist simply of moving the data up from the FPP C-register to the FPP B-register. Other operations would be: multiply, divide, add, and subtract. There are 11 instructions in the TLD group. This page, however, describes only the triple-loading process.

5-53. During cycle 2, the cycle that immediately follows phase 1 (page 5-4), EAU obtains the address of the first word it is to fetch. During T0 through T2, the P123 signal causes memory to read out the address word to the T-register, and at T7 the word is transferred into the M-register. The transfer is accomplished by the MD2 signal, which is high during the count of 6. The MD 2 signal generates RTSB and ADF at T6T7 (see figure 7-4), which put the T-register onto the T-bus; then at T7, SWSM stores the T-bus into the M-register. If indirect addressing is indicated (TB15 = 1), the counter is prevented from incrementing, and another memory cycle occurs (not shown in the timing diagram), which fetches a new address word. Then, if TB15 is 0, the count increments to 7 at the end of T7S.

5-54. Cycle 3 reads the first word out of memory to the T-register. At T4 a TIN2 signal loads bits 0-7 into the output buffer. (See figure 7-6.) Since the computer backplane does not wire all T-register bits to the EAU slots, a shifting operation is now necessary to load the high byte (bits 8-15). The Shift flip-flop is set (at T6) which incidentally turns off P123 since another word from memory is not wanted yet. The Shift signal generates RTSB and SRMB (see figure 7-4), which perform the reading and shifting, and SWST. The SWST signal is pulsed by TS on the instruction decoder card in the CPU, resulting in a repetitive shift. The first two shifts occur in cycle 3. Then at the next T0, the HI-LO flip-flop switches to H1 in preparation for turning off the Shift signal at T4.

5-55. Cycle 4 continues the shifting process four more times, for a total of six. Only six shifts are required, rather than eight, because T-register bits 8 and 9 are available to the EAU timing card. The high byte (TR8-15) is therefore shifted to TR2-9. At T4 the Shift flip-flop is reset and a TIN3 signal loads the high byte into the output buffer. (See figure 7-4.)

5-56. Then at T5, if the Flag is high (FPP ready), the IO code is set to 01. Assuming for now that the count is still 7, the buffer-reading signals TIN4 and TIN5 are enabled (U92D now false), and ENC is sent to the FPP unit. This tells the FPP to load the 16 bits that the buffer is now reading out.

5-57. Meanwhile, the M-register is incremented in preparation for fetching the next word. (The count is still assumed to be 7, so TIN4, which permits incrementing to occur, is high.) Incrementing occurs by enabling TS34, which generates the appropriate signals on the EAU logic card (see figure 7-4): at T6T7 RMSB, ADF, and RB0 read and increment the M value, and at T7 SWSM stores the value back into the M-register. The HI/LO flip-flop toggles to LO at T0, in preparation for loading the next low byte. (LO enables P123 to read memory and enables TIN2, which is true at T4.)

5-58. During T0, in cycle 5, the FPP Flag goes low, indicating that it has loaded the first data word and is busy preparing for the next word. The low Flag clears the ENC signal by resetting IO1. With IO1 and IO2 both reset, TIN4 and TIN5 are also cleared (by U92D). The count now increments to 8, and the entire read-load process now repeats (to paragraph 5-54).

5-59. On the second loop (count 8, cycle 6), a check is made to see if the instruction is LDF (TR0= 0 and TR1 = 0). The identifying bits are stored in the Bit 0 and Bit 1 flip-flops. If the instruction is identified as LDF, TIN4 is inhibited from reading the low byte of the output buffer (see figure 7-6). This results in reading 8 zeros for the low byte, as required by the format conversion. (Refer to section III). Also, since TIN4 is low, the TS34 signal is inhibited, with the result that the M-register is not incremented. The second operand will therefore be fetched again in the third loop.

5-60. On the third loop (count 9, cycle 8), checks are made to see if the instruction is LDF (TR0 = 0, TR1 = 0) or LDD (TR0 = 0, TR1 = 1). For LDF, TIN5 is inhibited from reading the high byte of the output buffer, resulting in 8 zeros for the high byte. For LDD, both TIN4 and TIN5 are inhibited, resulting in an all-zero word. The final incrementing of the M-register in cycle 8 is insignificant since this register will be forced to P+1 in the FPF routine.

5-61. When the count goes to 10, and the last Flag goes high after about 2.2 microseconds (cycle 10), an OPC signal is generated. This signal reads out the opcode (still stored on the EAU interface card) and commands the FPP unit to load the opcode. This action initiates the FPF (floating point function) routine, described on the next page.

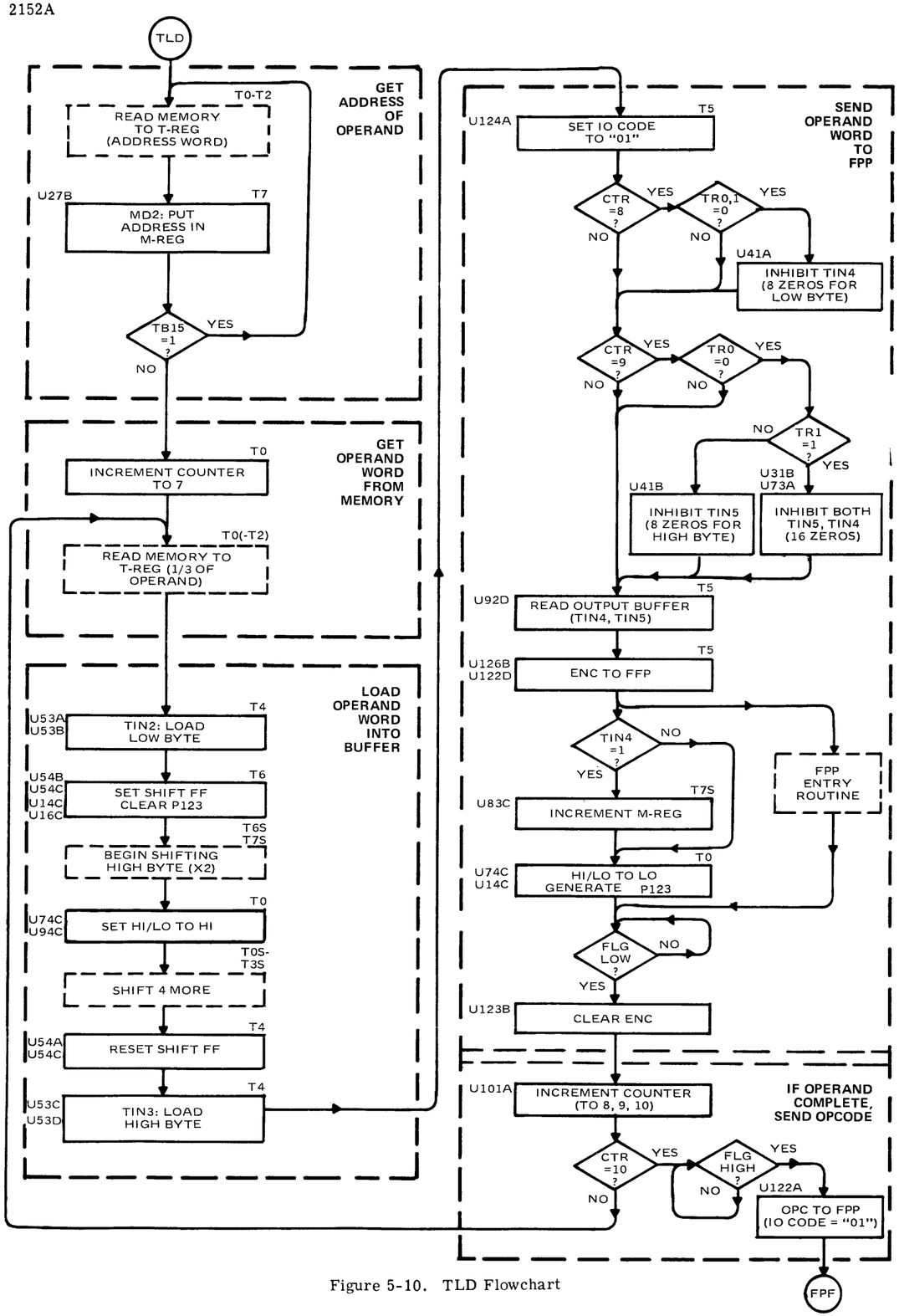
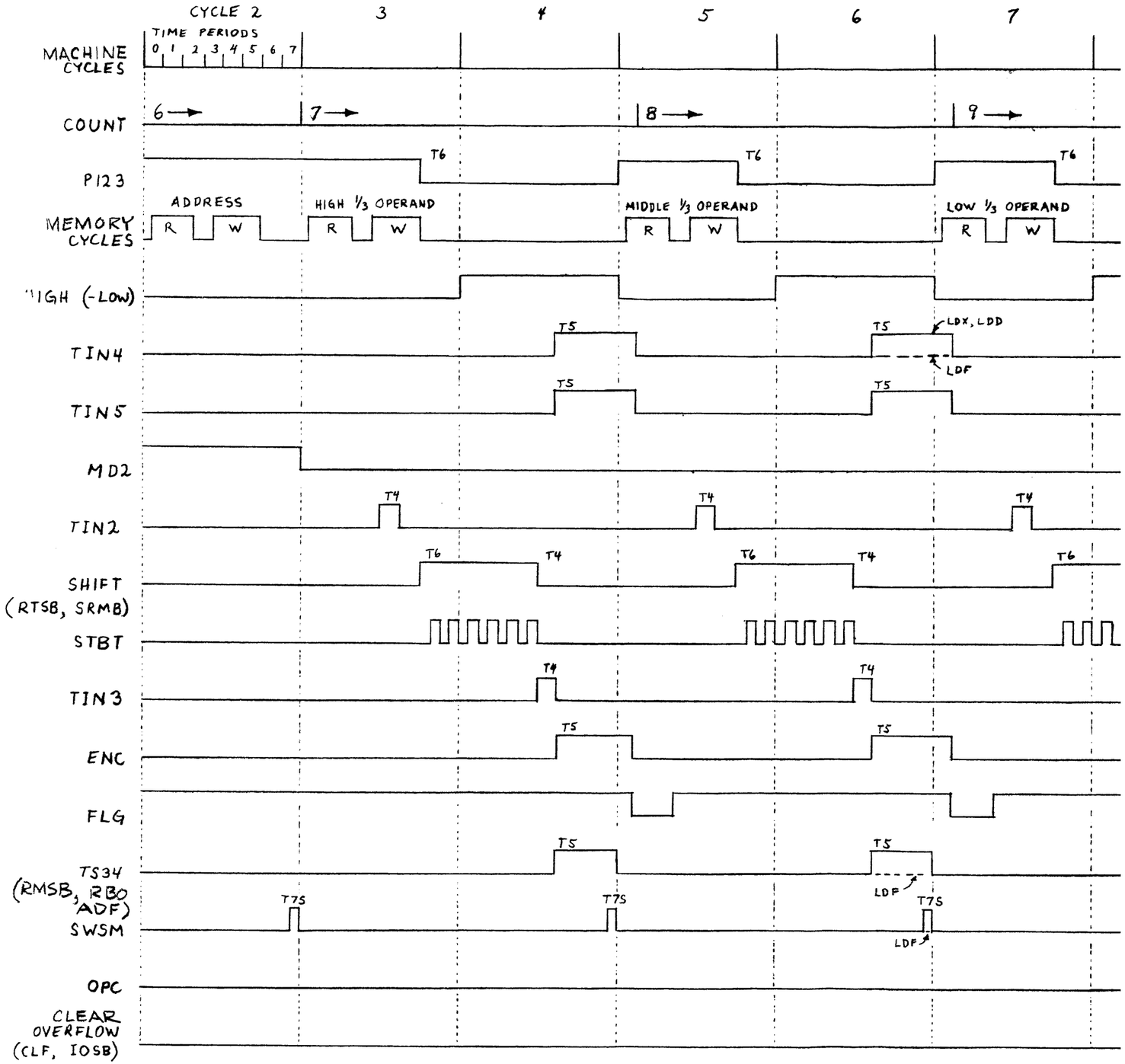


Figure 5-10. TLD Flowchart



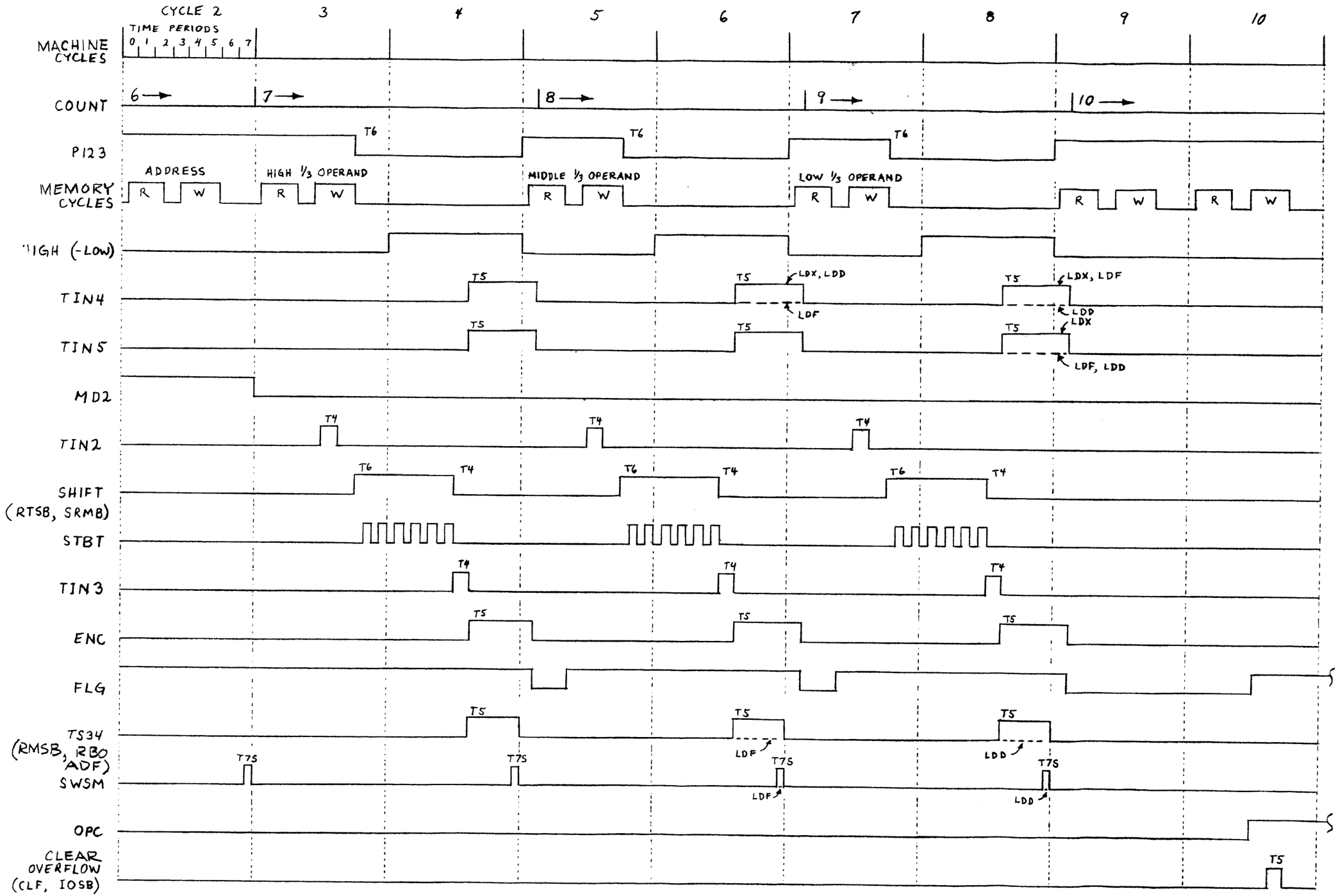


Figure 5-11. TLD Timing

5-62. FPF OPERATION.

5-63. The FPF (floating point function) routine provides an error-checking and exit sequence for the TLD and FPF groups of instructions. Before this sequence begins, the opcode has already been sent to the FPP unit. (Refer to pages 5-4 and 5-10.) The sequence begins when the FPP unit lowers the Flag, acknowledging receipt of the opcode. This sets the IO code (which was at 01) to 11. In this state EAU is waiting for a high Flag. The Flag will remain low as long as it is executing the function specified by the opcode. This may be a relatively long period, from about 10 to 200 microseconds.

5-64. When the Flag goes high, signifying completion of its computation, the Exit flip-flop is set at the next T4, and a check is made for a possible error. If the ERR signal from the FPP unit is true, it will load the error code into the input buffer. (See figure 7-6.) Table 5-4 lists the assigned error codes (only bits 8 through 15 are used). The ERR signal also generates the signals that transfer the input buffer contents to the computer A-register. The IOIO and IOI signals are enabled at T5, which, respectively, read the Buffer contents to the IOBI lines and transfer the IOBI lines to the S-bus in the CPU.

5-65. Then at T5, SWSA and EOFB are enabled. (See figure 7-4.) In the CPU, SWSA is strobed by TS to produce STBA (Store T-Bus in A). If an ERR (Error) signal is present, the IOIO and IOI signals cause the error code to be loaded into the A-register; otherwise, the A-register contents remain zero.

5-66. As a testable error indication, the ERR signal also sets the CPU Overflow flip-flop. This is done by enabling the OVD line, which generates IOSB (the I/O address of Overflow) and STF at T6T7.

5-67. The Exit signal also goes to the EAU logic card (see figure 7-4), where it generates the signals necessary to increment the P-register. At T6T7, RPRB, ADF, and SB0 read and increment the current value of P, and at T7 SWSM and SWSP store the values into the P- and M-registers. These registers are now ready to address the next instruction in memory.

5-68. Also occurring at T7 is the Reset signal, generated by the Exit signal. This clears the TLD, MAC, IO1, IO2, and Shift flip-flops.

5-69. The final action of EAU is to re-enable the CPU. This is done at the end of T7S by setting the EPH (Enable Phase) flip-flop, which ends the IIR signal to the CPU.

NOTE

This completes the discussion of the controller portion of the EAU logic. The section continues on page 5-15 with a discussion of the non-floating-point instructions.

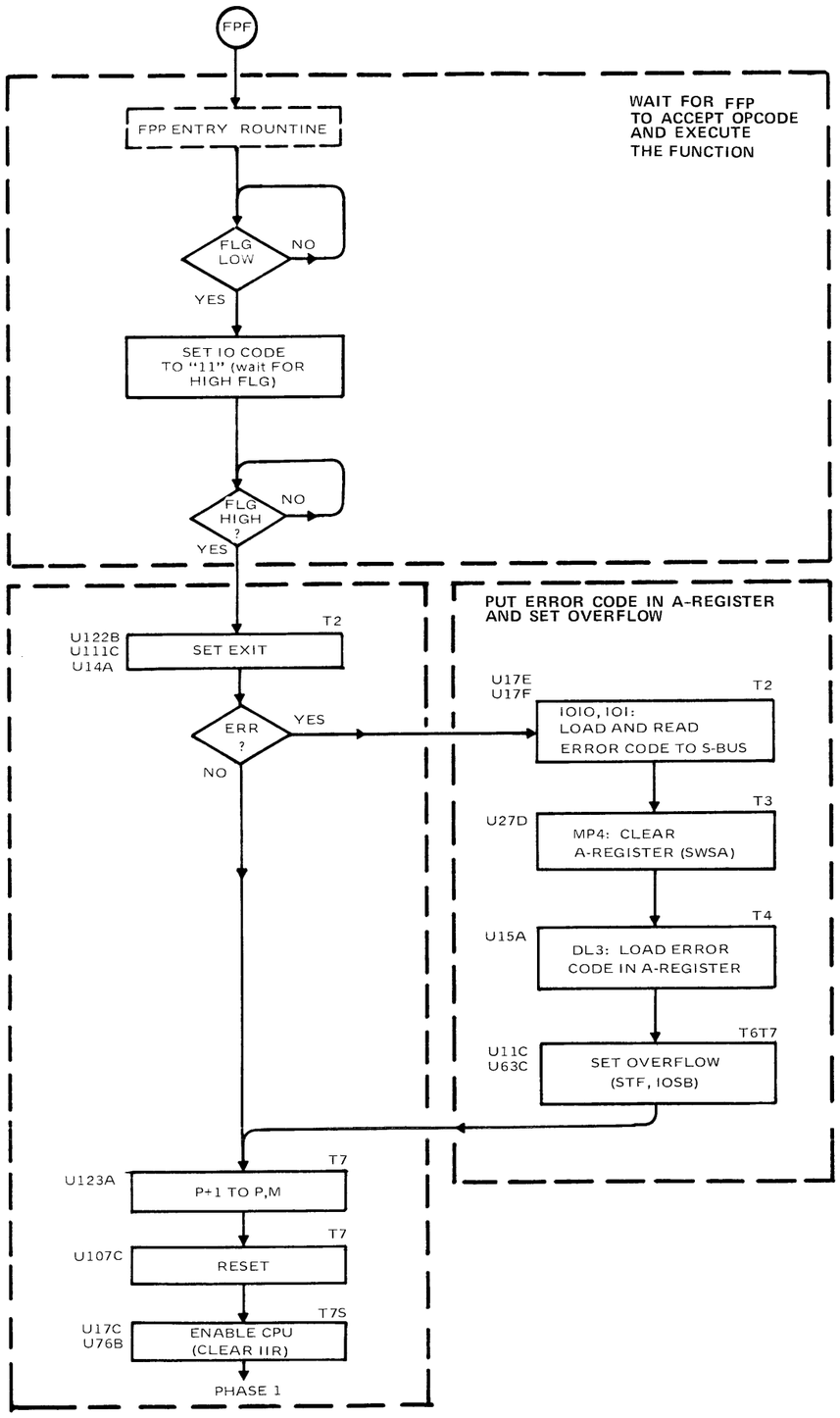


Figure 5-12. FPF Flowchart

Table 5-4. Error Codes

ERROR	CODE (A-Reg) 10	9	8	INSTRUCTION	CONDITION
0 No Response	0	0	0	Any, except: AS, LS, RO, MPY, DIV, DLD, DST	
1 Underflow	0	0	1	ADF, ADX, SBF, SBX, MPF, MPX, DVF, DVX	Result is in the range $0 < z < (1/2)\,2^{-128}$ or $0 > z \geq (-1/2)\,2^{-128}$
				CMX	$x = (1/2)\,2^{-128}$
				HCX, HSX	$x < -88$
				EXX	$x < -87.3$
2 Overflow	0	1	0	STD, FIX	$x < -2^{31}$ or $x \geq 2^{31}$
				ADF, ADX, SBF, SBX, MPF, MPX, DVF, DVX	Result is $\geq 2^{127}$ or $< -2^{127}$
				ABX, CMX	$x = -2^{127}$
				HCX, HSX	$x > 88$
				EXX	$x > 87.3$
				RNX	Result $= 2^{127}$
3 No Resolution	0	1	1	CSX, SNX, TNX	$\lvert x \rvert \geq 2^{38}$
4 Divide by Zero	1	0	0	DVF, DVX	$y = 0$
				TNX	$x = (2k+1)\pi/2$ for $k=0, \pm1, \pm2, \ldots$
5 Improper Variable	1	0	1	AHT	$\lvert x \rvert \geq 1$
				LNX	$x \leq 0$
				SRX	$x < 0$
6 Improper Opcode	1	1	0	Any	

Notes: x = Contents of X Register before execution
y = Memory contents
x = Contents of X Register after execution

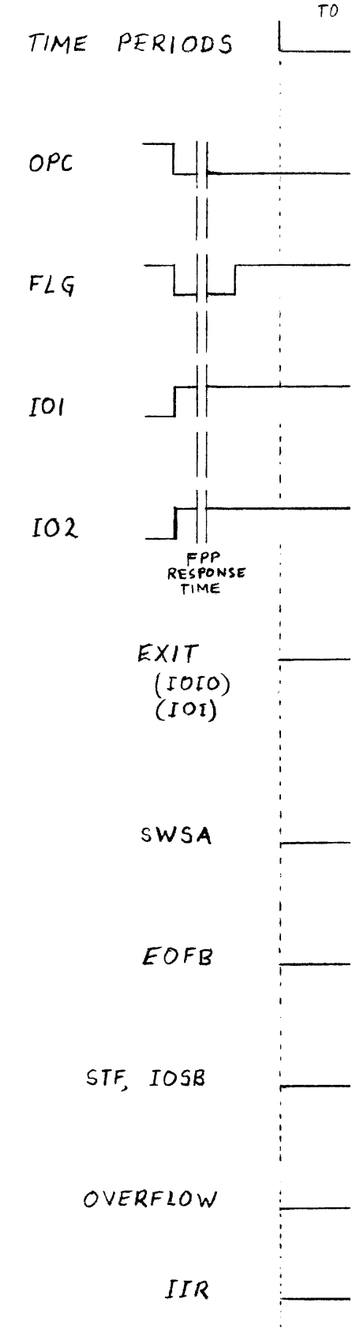


Table 5-4. Error Codes

ERROR	CODE (A-Reg) 10	9	8	INSTRUCTION	CONDITION
esponse	0	0	0	Any, except: AS, LS, RO, MPY, DIV, DLD, DST	
rflow	0	0	1	ADF, ADX, SBF, SBX, MPF, MPX, DVF, DVX	Result is in the range $0 < z < (1/2)\,2^{-128}$ or $0 > z \geq (-1/2)\,2^{-128}$
				CMX	$x = (1/2)\,2^{-128}$
				HCX, HSX	$x < -88$
				EXX	$x < -87.3$
flow	0	1	0	STD, FIX	$x < -2^{31}$ or $x \geq 2^{31}$
				ADF, ADX, SBF, SBX, MPF, MPX, DVF, DVX	Result is $\geq 2^{127}$ or $< -2^{127}$
				ABX, CMX	$x = -2^{127}$
				HCX, HSX	$x > 88$
				EXX	$x > 87.3$
				RNX	Result $= 2^{127}$
esolution	0	1	1	CSX, SNX, TNX	$\lvert x \rvert \geq 2^{38}$
e by Zero	1	0	0	DVF, DVX	$y = 0$
				TNX	$x = (2k+1)\pi/2$ for $k=0, \pm1, \pm2, \ldots$
oper Variable	1	0	1	AHT	$\lvert x \rvert \geq 1$
				LNX	$x \leq 0$
				SRX	$x < 0$
oper Opcode	1	1	0	Any	

Notes: x = Contents of X Register before execution
y = Memory contents
x = Contents of X Register after execution

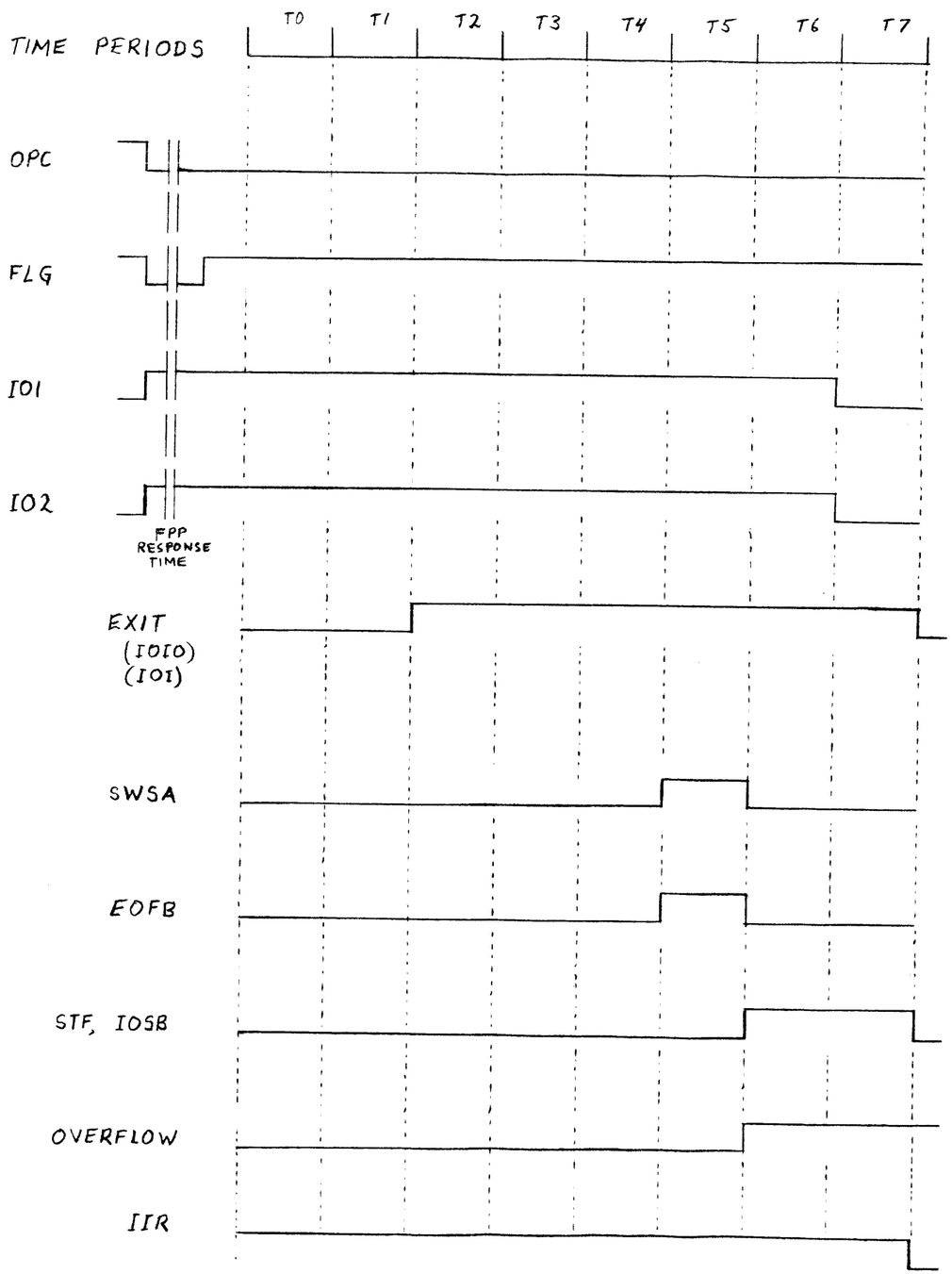


Figure 5-13. FPF Timing

5-70. EAU FUNCTIONS.

5-71. The remainder of this section provides theory of operation for those functions of the FPP which operate only on the computer A- and B-registers, and are not involved with the floating point processor unit. These functions are: integer multiply and divide, double load and double store, and long shifts and rotates. Table 5-5 lists the machine codes for these instructions. (Instruction definitions are given in section III of this manual.)

5-72. Figure 5-22 is a generalized block diagram of the extended arithmetic unit, integrated with the computer CPU block diagram. (Input/output is omitted since it is not involved in EAU operations.) As shown, the EAU hardware consists of two printed circuit cards: timing and logic. All of the signals entering or leaving these two boards are identified in figure 5-22. The following paragraphs (through 5-81) describe in general terms the functions that these signals are intended to accomplish.

5-73. STARTING THE EAU OPERATION. When the instruction register (I-reg) decodes an EAU group instruction from the T-register (bits 15, 14, 13, 12, 10; refer to table 5-5), a MAC signal is sent to the EAU timing card. This signal enables EAU.

5-74. First, the MAC signal loads the seven instruction-determining bits into the operation decoder, which decodes the type of operation to be performed (multiply, divide, rotate, etc.). The resulting decoded output does several things. For shifts and rotates, it loads TR0 through TR3 into the operation cycle counter, or, for the other instructions, it presets the counter to a value of 5 (ignoring TR0-3). (The significance of the number 5 will be explained later.) The decoded instruction signal also

Table 5-5. Extended Arithmetic Unit Machine Coding

INSTRUCTION	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MPY	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
DIV	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
DLD	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
DST	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
ASR	1	0	0	0	0	0	1	0	0	0	0	1	*n			
ASL	1	0	0	0	0	0	0	0	0	0	0	1	*n			
LSR	1	0	0	0	0	0	1	0	0	0	1	0	*n			
LSL	1	0	0	0	0	0	0	0	0	0	1	0	*n			
RRR	1	0	0	0	0	0	1	0	0	1	0	0	*n			
RRL	1	0	0	0	0	0	0	0	0	1	0	0	*n			

Shaded bits define EAU group (MAC signal); remaining bits determine specific instruction.

*n = number of shifts or rotates:

1 = 1 shift or rotate
2 = 2 shifts or rotates
3 = 3 shifts or rotates
4 = 4 shifts or rotates
5 = 5 shifts or rotates
6 = 6 shifts or rotates
7 = 7 shifts or rotates
8 = 8 shifts or rotates
9 = 9 shifts or rotates
10 = 10 shifts or rotates
11 = 11 shifts or rotates
12 = 12 shifts or rotates
13 = 13 shifts or rotates
14 = 14 shifts or rotates
15 = 15 shifts or rotates
0 = 16 shifts or rotates

8 4 2 1
Binary Value

generates an IIR (Inhibit Instruction Register) signal and a $\overline{\text{EPH}}$ ("not" Enable Phase) signal. Together, these two signals disable normal CPU operations, and EAU takes control.

5-75. EXECUTING EAU INSTRUCTIONS. Since memory is disabled ("not"-EPH), the T- and M-registers do not affect memory, so all five registers are available to EAU for temporary storage of data. There is one limitation, however, in that direct memory access (DMA) could insert a phase 5 cycle at the end of any EAU cycle, and thus destroy the contents of the T-register. Therefore all significant data in the T-register must be transferred elsewhere before the end of the cycle in which it occupies this register.

5-76. In addition to having complete control of the registers, EAU can also read or write in memory by generating a P123 signal. (ISG is necessary for the store operation, to inhibit loading memory data into the T-register.)

5-77. All of these operations (manipulating the registers and reading or writing in memory) occur under strict timing sequences determined by the timing encoding logic. Inputs to this logic include most of the basic timing signals of the computer (T0, T1, T2, etc.), plus the current count of the operation cycle counter, and the decoded instruction. Outputs to the EAU logic card comprise the 29 timing signals shown as direct connections between the two boards. The three shift signals and two memory signals (P123, ISG) are sent to the CPU. These 34 signals determine when and how the registers are to be manipulated, selecting any of the 5 Read signals, 5 Store signals, 3 Function signals, and 3 Shift signals, plus when to read or write in memory. For example, the MP1 through MP5 signals sequentially set up and then perform the operations of a multiply: first obtaining the address of the multiplier, then fetching the multiplier, then converting negative numbers to positive form, then doing the step-by-step shift and add series of operations that accomplish the multiply.

5-78. Shifts and carries into or out of the ends of registers are treated in different ways, depending on the operation being performed. The Sign, Link, and Carry flip-flops provide temporary storage and a means for manipulating these bits. Bit routing is provided to and from the high or low end of the CPU buses (RB15, RB14, RB0, SB0, C16, TB15, TB0).

5-79. Checking for overflow conditions is done on the logic card, and resultant indications are sent to the timing card (OVR, OASL, etc.). Depending on the type of operation, the CPU Overflow flip-flop may be set immediately or delayed until the end of the operation. The STF signal with IOSB (Overflow address) sets the Overflow flip-flop.

5-80. ENDING THE EAU OPERATION. As the EAU operation progresses, the operation cycle counter is incremented on each step. When the counter reaches the final count, it causes an Exit signal to be generated (via timing encoding). This terminates the IIR signal (and increments the P- and M-registers) and thus returns control to the CPU.

5-81. In the event that EAU should attempt to store a double word in an area of core protected by a memory protect option, an RSDS signal immediately clears the operation decoder and causes an Exit, as in the preceding paragraph. The CRS signal clears the EAU logic in the same way, except that it occurs under instruction control (CLC 0).

5-82. OPERATION CYCLE COUNTER.

5-83. Most of the logic circuits in the EAU hardware are relatively simple, consisting largely of gates that are activated in a set sequence. Therefore, most of this theory of operation will be presented on the basis of time sequences rather than circuit descriptions. However, the counter which generates the sequencing signals does require some explanation, in that its operation may not be readily apparent. The following paragraphs, through 5-92, describe the operation cycle counter (for brevity, frequently identified in this text as "the counter").

5-84. Figure 5-14 shows the counter logic. The counter itself consists of four J-K flip-flops, connected as a 4-bit binary counter. The use of J-K flip-flops permits accurate marking of time, by avoiding binary-state propagation delays at the instant of incrementing, inherent in simple binary counters. All flip-flops respond simultaneously to the incrementing clock signal, changing states as required on the trailing edge of the Clock.

5-85. Note that is is necessary for the two higher order flip-flops to use pin 7 of the preceding flip-flop as its input, rather than the normal set output at pin 13. If this were not done, OC2 would set prematurely on the first Clock after OC1 becomes set instead of waiting for OC0 to also become set. (Remember that in a binary counter, one flip-flop can be set only after all preceding flip-flops are set.) Referring to figure 5-15, note that pin 7 comes from a gate which (in addition to Clock) requires that the flip-flop's own set output is true and the preceding flip-flop (whose output appears at both pins 8 and 14) is also true. In a binary chain, a true output from pin 7 is therefore a signal that this flip-flop and all preceding flip-flops are set.

5-86. The normal clear state of the counter is when all four flip-flops are set (rather than clear), or when the count is 15 (decimal). This state is termed "counter at operational zero", and a low CTO0 signal in the logic signifies this condition. The counter is set to this state at T1 of every phase 1 (top gate in figure 5-14). The reason for using 15 instead of 0 for the clear state is that it permits the use of one's complement subtraction when (as will be discussed later) it is desired to put the negative value of the desired shift count into the Counter. One's complement arithmetic is simpler to implement in hardware than two's complement. All that is necessary (refer to next paragraph) is to use the positive binary value of the shift count to reset the corresponding flip-flops.

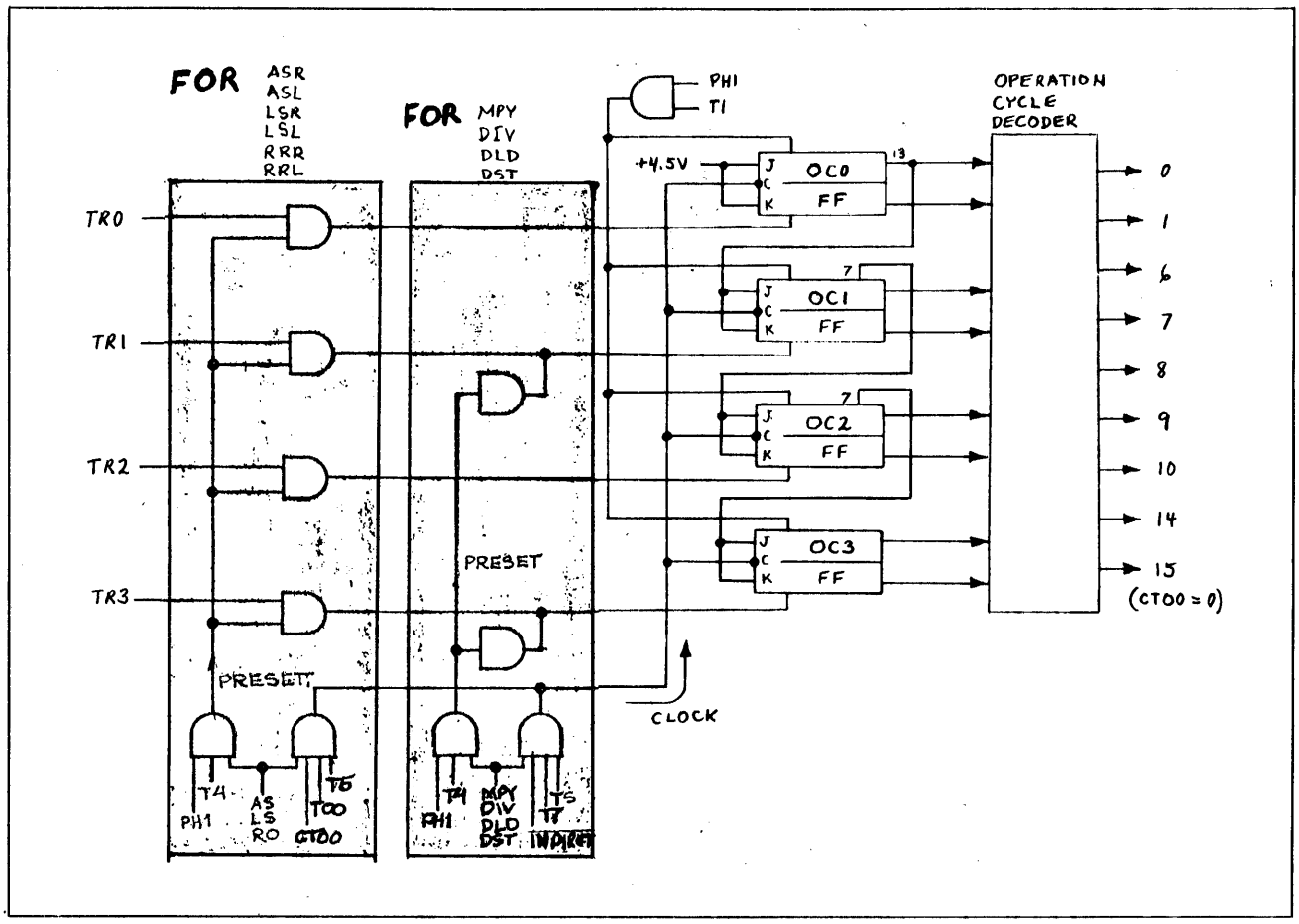


Figure 5-14. Presetting and Clocking the Counter

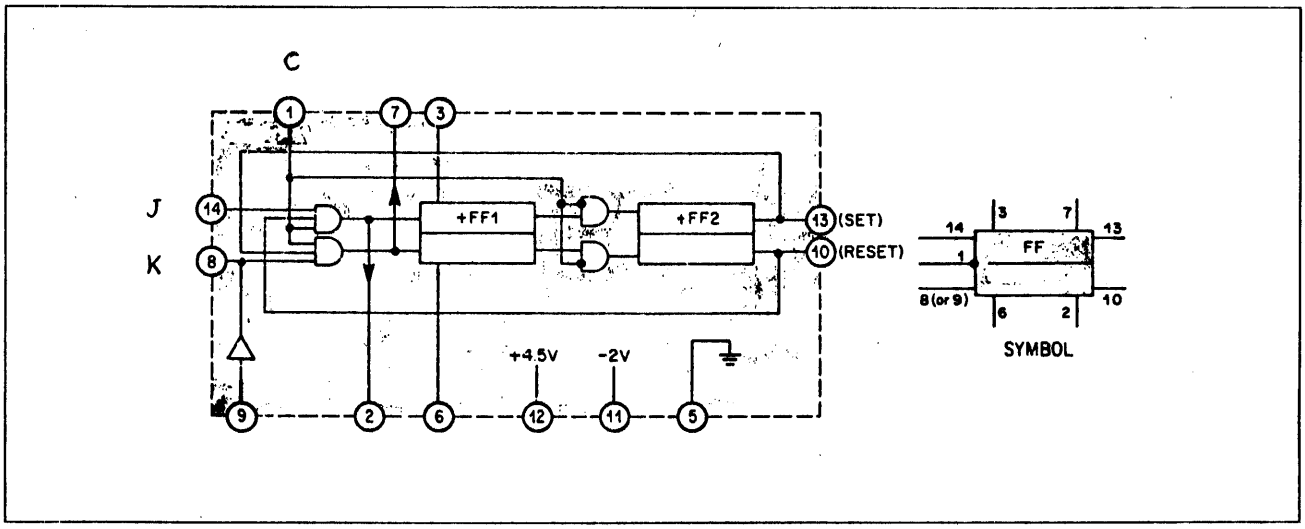


Figure 5-15. Equivalent Logic of J-K Flip-Flop

5-87. The counter operates under two different modes, identified by shaded blocks in figure 5-14. For the shifts and rotates, the counter is preset at T4 of phase 1 by strobing T-register bits 0-3 into the counter. (As shown in table 5-5, these bits represent the desired number of shifts to be accomplished.) Since the preset input is to the clear side of the flip-flops, the counter now contains the one's complement value of the shift count. For example, if the desired number of shifts is 8 (TR3 = 1), the counter will contain the value 7 (OC3 cleared, leaving OC0, 1, 2 set). Then, when the counter is incremented (clocked at every odd time interval by TOD) the counter advances from 7 toward the full count of 15. (TOD is true at every T1, T3, T5, or T7.) One shift is accomplished each time the counter increments; thus by the time the count reaches 15, the desired 8 shifts will have been accomplished.

5-88. For multiply, divide, double load, and double store, the counter is always preset to a value of 5 by clearing the OC1 and OC3 at T4 of phase 1. To some extent, 5 is an arbitrary number, chosen for ease of decoding the desired final count. Depending on which of the four instructions is being executed, it will be desired to know when 3 counts have elapsed (DLD and DST), or 11 counts (MPY), or 12 counts (DIV). Thus signals representing counter values of 8, 0 (rollover), and 1, respectively, are needed. These counter values are shown among those decoded by the operation cycle decoder, in figure 5-14. (The purposes of the other decoded counts will become apparent later in this section; note that not all counts need to be decoded.) Incrementing of the counter occurs at T7TS of every cycle during the execution, with the exception that when the logic obtains an address from memory, incrementing will be inhibited if the address is indirect. All EAU operations will be suspended until a direct address is found.

5-89. Tables 5-6 and 5-7 illustrate the count sequences for all EAU instructions. As explained in paragraph 5-87 for shifts and rotates, the one's complement value of the desired number of shifts is preset into the counter, which is then incremented (with each shift) until the count reaches 15. Table 5-6 illustrates this process. Note that up to 4 counts (shifts) occur in each machine cycle, with the result that a maximum of five and a minimum of two machine cycles are required to produce the full range of shifts. The first shift cannot occur until the last quarter of cycle 1 due to instruction decoding time and counter setup time; also one quarter of a cycle is required for the exit operations (re-enabling the CPU and incrementing P and

Table 5-6. Count Sequences for AS/LS/RO

T-REGISTER BITS 3 2 1 0	NO. OF SHIFTS	PRESET COUNTER VALUE	CYCLE # 1	2	3	4	5
0 0 0 1	1	14	14 15	Exit			
0 0 1 0	2	13	13 14	15 Exit			
0 0 1 1	3	12	12 13	14 15 Exit			
0 1 0 0	4	11	11 12	13 14	15 Exit		
0 1 0 1	5	10	10 11	12 13	14 15 Exit		
0 1 1 0	6	9	9 10	11 12	13 14	15 Exit	
0 1 1 1	7	8	8 9	10 11	12 13	14 15 Exit	
1 0 0 0	8	7	7 8	9 10	11 12	13 14	15 Exit
1 0 0 1	9	6	6 7	8 9	10 11	12 13	14 15 Exit
1 0 1 0	10	5	5 6	7 8	9 10	11 12	13 14 15 Exit
1 0 1 1	11	4	4 5	6 7	8 9	10 11	12 13 14 15 Exit
1 1 0 0	12	3	3 4	5 6	7 8	9 10	11 12 13 14 15 Exit
1 1 0 1	13	2	2 3	4 5	6 7	8 9	10 11 12 13 14 15 Exit
1 1 1 0	14	1	1 2	3 4	5 6	7 8	9 10 11 12 13 14 15 Exit
1 1 1 1	15	0	0 1	2 3	4 5	6 7	8 9 10 11 12 13 14 15 Exit
0 0 0 0	16	15	15 0	1 2	3 4	5 6	7 8 9 10 11 12 13 14 15 Exit

M), which is assigned to the final quarter of the last shifting cycle. Thus the minimum number of cycles is necessarily two, as shown by the top line of the table. The first column of table 5-6 shows the coding of the least significant bits of the instruction, which specify the desired number of shifts (second column). The third column shows the decimal value of the counter (as preset by a one's complement of column 1), and the remaining five columns show the incrementing sequence of the counter, each quarter cycle, for each of the 16 possible degrees of shift.

5-90. As explained in paragraph 5-88, for MPY, DIV, DLD, and DST the counter is preset to 5, then incremented at the end of each machine cycle (except for a possible inhibit due to indirect addressing). Table 5-7 illustrates the count sequence for these four instructions. Note that the number of machine cycles required varies according to the type of instruction (and, except for possible indirect phases, is fixed at that number, rather than variable, as for shifts and rotates). For multiply, this number is 12 cycles, for divide 13, and for double load and double store 4.

5-91. In contrast to the simple repetitive process of shifting, which requires little more than checking for a full count, operations such as multiply and divide are relatively complex. Thus at various stages of the count, unique signals are generated in order to accomplish specific unique functions in a set sequence. Table 5-7 lists these signals, identified by the cycle in which they occur. For example, MP1 is active in the first cycle of a multiply, MP2 in the second cycle, and MP3 in the third. MP4 (here the process becomes repetitive) is active for the fourth through eleventh cycles, and MP5 provides exit operations in the twelfth cycle. Note that MP5 occurs when the counter rolls over from 15 to 0; for divide, the counter will also roll over and advance to 1 in order to reach the D6 exit cycle in the thirteenth cycle.

5-92. The remainder of this section describes in detail the execution of each of the EAU instructions. Logic equations and flowcharts at the end of the section are to be used in conjunction with the descriptive text.

5-93. MULTIPLICATION.

5-94. Before going into execution sequence of the MPY instruction, it is first necessary to understand the theory of multiplication. The apparently simple process of multiplying two numbers actually involves a series of steps which, for convenience, are com-

Table 5-7. Count Sequences for MPY/DIV/DLD/DST

CYCLE #	COUNTER VALUE (DECIMAL)	COUNTER STATES				ENABLED TIMING SIGNAL			
		OC3	OC2	OC1	OC0	MPY	DIV	DLD	DST
1	5	0	1	0	1	MP1	D1	DL1	DS1
2	6	0	1	1	0	MP2 (MD2)	D2 (MD2)	DL2	DS2
3	7	0	1	1	1	MP3	D3	DL3	DS3
4	8	1	0	0	0	MP4	D4	DL4	DS4
5	9	1	0	0	1	MP4	D5		
6	10	1	0	1	0	MP4	D5		
7	11	1	0	1	1	MP4	D5		
8	12	1	1	0	0	MP4	D5		
9	13	1	1	0	1	MP4	D5		
10	14	1	1	1	0	MP4	D5		
11	15	1	1	1	1	MP4	D5		
12	0	0	0	0	0	MP5	D5		
13	1	0	0	0	1		D6		

Shaded areas indicate time during which EAU is multiplying or dividing. MP1-3 and D1-4 are initialization cycles. MP5 and D6 are exit cycles.

bined by convention into the shortest possible form. However, for a computer to duplicate this process, it is necessary to break the process down into its constituent steps. This is because the computer is a binary machine, and can handle no more than one step at a time. Paragraphs 5-95 through 5-111 discuss the derivation of the binary technique of multiplication used by EAU.

5-95. MULTIPLICATION THEORY.

5-96. DECIMAL MULTIPLICATION. Figure 5-16 illustrates the conversion of the conventional form of multiplication (box A) to a decimal equivalent of what EAU does (box E). Notice in all cases that, although the form of the computation varies, the numbers in the partial products and final product are consistently the same.

5-97. In box A, the first step toward viewing the operation in machine terms has already been taken. That is, individual numerals must be thought of as being positioned or moved within stationary columns. In the computer, numerals will be accumulated and moved in stationary registers. Shading of triplet columns in figure 5-16 emphasizes the columnar approach.

5-98. To analyze what is done in conventional multiplication, note that the multiplicand (2468) is successively multiplied by each digit of the multiplier, in order of the least significant to the most significant digit. Since each multiplier digit has a place value of ten times the preceding digit, each succeeding product is increased in value ten times by being shifted left one column position. The final step is to add all products resulting from each multiplication.

5-99. Thus there are three separate actions involved in multiplication. First, there are the individual multiplications of multiplier digit times multiplicand; then there is a shifting action to determine place value, and finally there is an addition to get the final product. Or, more concisely: multiply, shift, add.

5-100. Since the computer can add only two numbers at a time, partial products must be accumulated after each multiplication by a multiplier digit. Box B illustrates this action. First we must assure that the starting value is zero (assumed for mental calculations, but not necessarily true of machines). Then we multiply by the first multiplier digit (1) and add to the existing accumulated value (0) to get the first partial product (2468). Then multiply by the second multiplier digit (5), shift the result one position to the left, and add to get the second partial product (125,868). Finally, multiply by 9, shift two places to the left, and add to get the final product.

5-101. In Box C, the process is expanded to separate the multiplication and shift operation into two separate steps, as the computer must do. For machine use, there are two disadvantages to this form. One is that there is no shift in the first cycle; this would require additional hardware to treat the first cycle differently from the succeeding cycles. The other disadvantage is that the left-shifting is with respect to the previous addend, rather than to the existing accumulated value. Again, this would require more hardware to store information from the preceding cycle.

5-102. In box D, we shift the partial products right one position after adding, instead of shifting the addend left before adding. The net result is the same, but the process is now uniform from cycle to cycle: multiply, add to existing partial product, shift result right. The process is now in a form that the computer can use.

5-103. Box E illustrates how EAU multiplies, except that the example is still in human terms (decimal). It is assumed there are two 5-digit decimal registers (approximately equivalent to the two 16-bit binary registers in the computer), and the objective is to fill these two registers (combined) with the final product.

5-104. Initially, register B is zero, the multiplier is in register A, and the multiplicand is in some easily accessible third register. Placing the multiplier in register A is an efficiency measure to save hardware. After each digit of the multiplier is used, it is no longer needed, so the entire multiplier can be shifted right and thus discard the inactive digit. Since the partial product will also be shifting right, the two actions can occur in synchronism, with the result that the lower half of the accumulated product gradually displaces the multiplier in register A. An additional advantage is that the active multiplier digit always appears in a uniform position (least significant digit of A), thus simplifying the required hardware. The multiplication process occurs as follows.

5-105. The low order digit of the multiplier in register A is read, multiplied with the multiplicand, added to the existing value of register B, and then the result is stored in register B. Then both registers are shifted-right together, with the least significant digit of register B shifted into the most significant digit position of register A. The previously used multiplier digit, at the same time, gets shifted off the end of register A and is lost. After this entire sequence has been repeated five times, the final product is the 10-digit result existing in the combined registers. Note that although the multiplier actually consists of three digits, it is necessary to go through the motions two additional times (multiply by 0) since the multiplying register (A) is a five-digit register. The two shifts in cycles 4 and 5 correctly position the final product.

5-106. BINARY MULTIPLICATION. Figure 5-17 illustrates in binary form the same process of multiplication shown in box E of figure 5-16, using the same numerical values. This is exactly the sequence that EAU follows in executing the MPY instruction.

5-107. First note the similarities with the decimal form in figure 5-16. As before, the multiplier is in

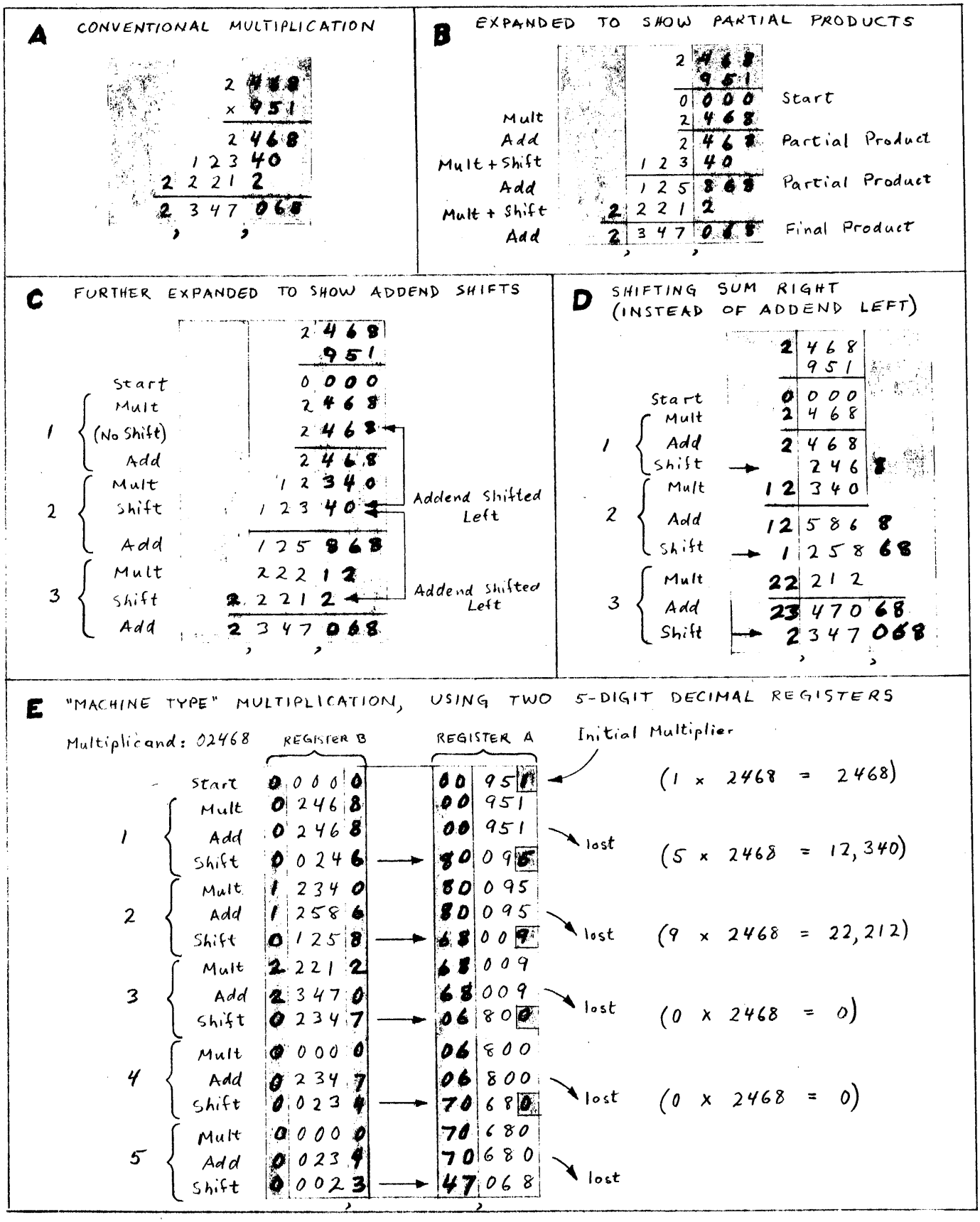


Figure 5-16. Decimal Multiplication

PROBLEM

2468 × 951 = 2,347,068

MULTIPLICAND (In M-Register)

DECIMAL EQUIVALENCE: SIGN 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1

0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

2048
256
128
32
4
2468

MULTIPLIER (In A-Register)

DECIMAL EQUIVALENCE: SIGN 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1

0	0	0	0	0	0	1	1	1	0	1	1	0	1	1	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

512
256
128
32
16
4
2
1
951

MULTIPLICATION PROCESS

	START	B-REGISTER	ACCUMULATING PRODUCT	A-REGISTER
		0 000 000 000	000 000 000	0 000 100 110 100 100
1	ADD	Don't Add		
	SHIFT	0 000 000	000 000 000	0 000 010 011 010 010
2	ADD	Don't Add		
	SHIFT	0 000 000	000 000 000	0 000 001 001 101 001
3	ADD	0 000 001	110 110 111	
	SHIFT	0 000 000	111 011 011	1 000 000 100 110 100
4	ADD	Don't Add		
	SHIFT	0 000 000	011 101 101	1 100 000 010 011 010
5	ADD	Don't Add		
	SHIFT	0 000 000	001 110 110	1 110 000 001 101
6	ADD	0 000 010	000 101 101	
	SHIFT	0 000 001	000 010 110	1 111 000 000 100 110
7	ADD	Don't Add		
	SHIFT	0 000 000	100 001 011	0 111 100 000 010 011
8	ADD	0 000 010	011 000 010	
	SHIFT	0 000 001	001 100 001	0 011 110 000 001 001
9	ADD	0 000 011	000 011 000	
	SHIFT	0 000 001	100 001 100	0 001 111 000 000 100
10	ADD	Don't Add		
	SHIFT	0 000 000	110 000 110	0 000 111 100 000 010
11	ADD	Don't Add		
	SHIFT	0 000 000	011 000 011	0 000 011 110 000 001
12	ADD	0 000 010	001 111 010	
	SHIFT	0 000 001	000 111 101	0 000 001 111 000 000
13	ADD	Don't Add		
	SHIFT	0 000 000	100 011 110	1 000 000 111 100 000
14	ADD	Don't Add		
	SHIFT	0 000 000	010 001 111	0 100 000 011 110 000
15	ADD	Don't Add		
	SHIFT	0 000 000	001 000 111	1 010 000 001 111 000
16	ADD	Don't Add		
	SHIFT	0 000 000	000 100 011	1 101 000 000 111 100

DECIMAL EQUIVALENCE: 2,097,152 131,072 65,536 | 32,768 16,384 4096 32 16 8 4

PRODUCT: 2,347,068

Figure 5-17. Binary Multiplication

the A-register, the B-register is clear, and the multiplicand is in a convenient third register (M-register). The number of operations, or cycles, is equal to the number of digit or bit positions in the multiplier (16 in this case).

5-108. The differences are only due to the fact that the mode is now binary instead of decimal. Note that the number of steps within each cycle is reduced from three to two (the multiply step is eliminated). This is due to the nature of the binary system, which has only the two digits 0 and 1. Multiplying the multiplicand by 0 results in 0; multiplying the multiplicand by 1 results in a value equal to the multiplicand. Therefore, in any cycle, the appearance of a 1 as the multiplier digit means: yes, add the multiplicand to the current accumulated product (1 x multiplicand); and 0 means: no, don't add the multiplicand (0 x multiplicand).

5-109. As an aid to understanding the process in human terms, figure 5-17 shows the decimal significance of each bit of the multiplicand and multiplier (and also the true bits of the product) to obtain the decimal equivalents of the binary numbers.

5-110. When the first cycle starts, the low order bit of the A-register is 0. Since 0 times the multiplicand is 0, we do not add in cycle 1. Then both registers are shifted-right together. This still results in a 0 as the multiplying digit, so in cycle 2 we still do not add. But the shift this time brings a 1 into the multiplying bit position (shaded), so in cycle 3 we do add (1 x multiplicand). Since the B-register has been zero up to this point, the addition results in a copy of the multiplicand. The next time a 1 is shifted into the multiplying position occurs in cycle 5; thus in cycle 6 the multiplicand is added to the value now existing in the B-register. Similarly, additions will occur in cycles 8, 9, and 12.

5-111. Note the advancement of the low half of the product (shaded), as the multiplier (unshaded) moves out of the A-register. The final product occupies both A- and B-registers, and has the same value as the previous decimal example.

5-112. MULTIPLY LOGIC.

5-113. MPY INSTRUCTION EXECUTION. The multiplication process cannot begin immediately when the MPY instruction occurs, because:

a. The multiplicand is in memory, which is an inconvenient location for repeated referencing.

b. One or both numbers may be negative.

c. Logic requires time to set up initial conditions.

5-114. Relative to item "a", the multiplicand needs to be fetched from memory and loaded into a register that is easy to reference. The M-register is the register assigned to this purpose. Negative numbers (item "6") need to be converted to the positive form, since the computer can add only in the positive direction. Also, the signs of the two numbers must be saved and compared, in order to determine the sign of the result. These two operations, plus logic initialization (item "C"), require three cycles of machine time.

5-115. Each cycle is identified as multiply cycle 1, multiply cycle 2, etc., and is activated by a corresponding unique signal (MP1, MP2, etc.). Multiply cycle 4 is repeated eight times, executing two add/shift operations. Therefore, the MP4 signal remains set for eight cycles before MP5 activates the final cycle, multiply cycle 5. This last cycle takes care of converting the answer to negative form if the compared signs indicate the result should be negative and returns control of the registers and memory to the CPU.

5-116. Figures 5-23 through 5-27, and tables 5-8 through 5-12 illustrate in detail the sequences of operations that execute the MPY instruction. The figures explain in flowchart form the intended action that the logic is meant to accomplish, and the tables list all the signals that accomplish each action. The tables, incidentally, provide a key to locating the specific gates (by reference designation) which are enabled in order to activate each signal listed in the tables.

5-117. The above mentioned figures and tables should be folded out for convenient reference during the following discussions of each multiply cycle (through paragraph 5-148). The logic diagrams, figures 7-2 and 7-4, should also be available.

5-118. MULTIPLY CYCLE 1. As indicated in figure 5-23, there are four major operations to be accomplished during the first cycle of an MPY instruction. These are:

a. Initialize the operation cycle counter, and look for the EAU group instruction signal, MAC.

b. Decode the MPY instruction, and set the Counter to 5, for MPY.

c. If the multiplier is negative, begin the conversion to positive form by complementing (the increment part of the 2's complement conversion occurs in the next cycle).

d. Prepare the EAU and the computer logic for reading memory in the next cycle. Remember that an EAU instruction always consists of two words: the instruction word and an address word which specifies the address of the multiplicand.

5-119. When any phase 1 begins, the instruction type is not known until the latter part of T2 when the computer instruction decoder decodes the instruction word read out of memory. During the time that the instruction word is being read out of memory, EAU clears the operation cycle counter (to the count of 15, as explained earlier in paragraph 5-86). This occurs at T1 of every phase 1 (see equation), regardless of whether the instruction is of the EAU type or not.

If the instruction is not of the EAU type, there is no MAC signal, and EAU remains disabled.

5-120. If there is a MAC signal from the CPU, the T-register bits which define the operation are clocked into the operation decoder at T3S. In the case of the MPY instruction, the MPY flip-flop will be set, and this begins the MP1 signal. At T4, the counter is set to 5, as explained earlier in paragraph 5-88.

5-121. Also at T4, the multiplier in the computer A-register is transferred to the B-register. In the process of doing so, the sign bit, which appears as bit 15 on the T-bus, is tested by gate U52B. If this bit is a "1", indicating a negative number, the SMR flip-flop will be set. Note the four logic equations for transferring A to B. The RARB signal reads A out to the R-bus, EOFB (Exclusive "OR", Buffered) passes the R-bus through the adder to the T-bus, unaltered since the S-bus is zero at this time, and SWSB (Switch Store in B) stores the T-bus in the B-register. The SMR signal is saved for use at T5, and also in cycles MP2 and MP5.

5-122. At T5 the multiplier in the B-register is read onto the R-bus by RBRB, complemented by CMFB, but stored back in the B-register (in this complemented form) only if SMR is set. Thus, if the number was positive, the value in the B-register will remain not complemented.

5-123. At T6T7, the CPU is permitted to increment the P- and M-registers. (MAC enables OPO in the computer.) The signals which accomplish this increment are listed in the EQUATION column of table 5-8, but equations are not given since these are not operations of the EAU option.

5-124. Then, at T7S, EAU disables the CPU by resetting the EPHX flip-flop, thus generating the IIR signal. At the end of T7S, which is the dividing line between the end of one cycle and the beginning of the next, the trailing edge of the Clock signal advances the counter to the count of 6. This enables the MP2 signal via gate U22C and thus permits a different set of signals to be activated during the next cycle, multiply cycle 2.

5-125. MULTIPLY CYCLE 2. In the second cycle, the main objective is to obtain the address of the multiplicand. The P- and M-registers have been incremented for this purpose in the preceding cycle. Since the possibility exists that the address fetched may be an Indirect address, provision is made to repeat the MP2 cycle as many times as necessary in order to obtain the final, direct address. Also, during this cycle, the conversion of a negative multiplier to positive form (begun in the preceding cycle) is completed, and the check is made to see whether an addition will be made in the first operation of the actual multiply routine (in multiply cycle 4).

5-126. Reading from memory is enabled by generating the P123 signal, which remains active from $\overline{T0}$ through T5. As shown in the logic equation, $\overline{T6T7}$ determines the time duration of P123, ending the signal when T6 starts. As usual, reading from memory places the word in the T-register.

5-127. During T3T4, the multiplier, presently in the B-register, is read out to the R-bus and, if SMR indicates that the sign was negative, is incremented and stored back in the B-register. Otherwise, if SMR is false, indicating a positive sign, the incremented value is not stored in the B-register, and thus the B-register retains its existing value.

5-128. At T4 the multiplier is on the T-bus due to the action described in the preceding paragraph. At this time, if the least significant bit (TB0) is "1", the MAF flip-flop will be set in preparation for the first multiplication add (in multiply cycle 4).

5-129. Incidentally, the Gate flip-flop, which produces the Gate signal used in the logic equations of the two preceding operations, is simply a means of assuring that the incrementing and MAF-setting operations occur only once in case MP2 is repeated for indirect addressing. The Gate flip-flop is initially set at T6 of the MP1 cycle. Then, if the first MP2 cycle finds an indirect bit on the T-bus (TB15) at time T6, the flip-flop will again be clocked. Since the MP1 input signal is now false, the flip-flop is latched to the clear state. The low Gate signal in succeeding MP2 cycles presents further incrementing of the B-register and setting of MAF. (Refer to the logic equations.)

5-130. At T6T7 the multiplicand address in the T-register is transferred to the M-register in preparation for reading the multiplicand (or possibly another address if the current one is indirect) from memory in the next cycle. At T7S the clock signal to the counter goes true only if TB15 is "0", indicating a direct address. Otherwise, the counter stays at its present count (6), thus causing a repetition of the MP2 cycle as a further search for a direct address. When the direct address is obtained, TB15 will be "0", and the counter will be clocked to the count of 7 at the trailing edge of T7S. This activates the MP3 cycle.

5-131. MULTIPLY CYCLE 3. The main objectives of the MP3 cycle are to fetch the multiplicand from memory and convert it, if negative, to the positive form. An incidental operation is to return the multiplier, presently in the B-register, back to its original position in the A-register.

5-132. At T0 the P123 signal is again activated in order to read memory. If the address was zero, the A-register is read, instead of memory, and loaded into the T-register. This is a function of the computer, so the relevant logic equations are not given; the active signals are listed as a guide for tracing out the action in the particular computer used with EAU.

5-133. At T2 the multiplier is transferred back to the A-register. The reason why the multiplier was transferred from A to B (in paragraph 5-121) and now from B back to A is to allow the A-register contents to be used, if desired, as both multiplier and multiplicand. This results in a mathematical square operation and is accomplished by referencing address 0

(A-register address) as the multiplicand (second word of the MPY instruction). If the temporary A-to-B transfer were not done, the act of converting a negative multiplier to positive form would destroy the original negative value. Thus the A-register contents remain unaltered throughout the MP1 and MP2 cycles until after the opportunity to read these contents as a multiplicand at T1 of the MP3 cycle.

5-134. Beginning at T3 the multiplicand, which has just been read into the T-register from either memory or the A-register, is transferred to the M-register throughout the multiplication process. This occurs in two steps: T-register to B-register, then B-register to M-register. The reason for the two-step process is to allow complementing and incrementing of the multiplicand if it is a negative number. At T3, the T-register is read onto the S-bus, passed through the adder by EOFB to the T-bus, and stored by SWSB in the B-register. If the sign bit on the T-bus indicates a negative number (TB15 = 1), the SMD flip-flop is set. Then at T5, depending on whether or not SMD was set, the B-register contents are transferred to the M-register in either complemented (CMFB) or uncomplemented (EOFB) form. Likewise, depending on the SMD state, the multiplicand is then incremented at T7.

5-135. In preparation for actual multiplication in the next cycle, the B-register is cleared. At T7S the counter is advanced to the count of 8, thus enabling the MP4 cycle.

5-136. MULTIPLY CYCLE 4. Each MP4 cycle accomplishes two add/shift operations. Therefore 8 complete MP4 cycles are necessary in order to obtain the 16 add/shift operations required by a 16-bit multiplier register.

5-137. The first add/shift operation occurs during T0 through T3. At T0T1, the multiplicand is read onto the S-bus and the current value of the high half of the accumulated product (initially zero) is read onto the R-bus. These values are added and routed to the T-bus. If the state of the MAF flip-flop indicates that addition should occur (refer to paragraph 5-128), the added result is stored back into the B-register. Otherwise, the B-register retains its existing value.

5-138. At T2 the B-register is read onto the R-bus, shifted right one position, and stored back in the B-register. During this process, the preshifted contents exist on the R-bus, allowing an opportunity to save the least significant bit (which otherwise would be lost due to the right shift). If RB0 is "1", it will set the Link flip-flop; otherwise the flip-flop, clocked by TEV and TS, will be cleared. (Note that the use of TEV in the Link flip-flop clock results in insignificant reading of RB0 at T0 and T4; however, since Link is read only at T3 and T7, the only significant readings of RB0 are those that occur at T2 and T6.)

5-139. At T3 the A-register is shifted. Its least significant bit is lost, the the saved bit from the B-register, now in the Link flip-flop, is read onto the T-bus (TB15). This bit is therefore stored into bit 15 of the A-register, simultaneously with storage of the shifted value of the A-register contents.

5-140. Since the new least significant bit of the multiplier exists on the T-bus at this time (T3), TB0 is read and sets the MAF flip-flop (if TB0 = 1) in preparation for the next add/shift operation.

5-141. The second add/shift operation of the MP4 cycle occurs during T4 through T7. The sequence of active signals is identical to those occurring from T0 through T3. Only the timing signals are different, advanced by four time periods. Thus T0T1 becomes T4T5, T2 becomes T6, etc. With only this minor difference, the process for the second add/shift operation is the same as described in paragraphs 5-137 through 5-140.

5-142. At the trailing edge of each T7S, the counter advances, first from 8 to 9, then 9 to 10, and so on, until the count rolls over from 15 to 0. At this point, eight MP4 cycles have occurred, and the multiplication process is therefore complete. The count of 0 activates the MP5 cycle.

5-143. MULTIPLY CYCLE 5. The MP5 cycle is an exit sequence, simply to convert the answer, if negative, to the negative form and return control to the CPU.

5-144. The combined outputs of the SMR and SMD flip-flops are connected in an "exclusive-or" gate arrangement. This gating will produce a Sign signal if either, but not both, of the flip-flops are set. This "exclusive-or" condition is consistent with the mathematical rule regarding positive or negative products from positive or negative multipliers and multiplicands. Thus if the Sign signal is true, the answer must be converted from the present positive form to its equivalent negative form.

5-145. At T0 the A-register is read and complemented, then stored back in the A-register if the Sign signal is true. At T1T2 the A-register is incremented by adding with a forced SB0, and stored if Sign is true. A possible carry out of the highest order adder (C16 signal) is saved in the Carry flip-flop. At T3 the B-register is also complemented but instead of incrementing with a forced SB0, the state of the Carry flip-flop determines whether SB0 is "1" or "0". This is a requirement of two's-complementing both A- and B-registers together as a single quantity. Thus by the end of T5 of the MP5 cycle, the final product exists in the combined A- and B-registers.

5-146. At T5 the Exit flip-flop is set in preparation for the final exit sequence (next paragraph), and at T6T7 the computer Overflow register is cleared. Clearing the Overflow register is merely a programming precaution since it is impossible to overflow a 32-bit product register by multiplying two 16-bit quantities. Thus any routine checks for Overflow, after programming an MPY instruction, will consistently find an indication of no overflow.

5-147. Also at T6T7, the P-register is incremented (added to a forced SB0) and stored in both P- and M-registers. This prepares the computer to read its next instruction in the following phase 1 cycle.

5-148. At T7S the EPHX flip-flop is set again, thus terminating the IIR signal. This enables the computer instruction register and phase control logic. At T7 the operation decoder is clocked, and since there are presently no T-register inputs, the operation decoder flip-flops (specifically the MPY flip-flop) are cleared. This completes the MPY sequence of operations.

5-149. DIVISION.

5-150. The process of division, when broken down into its constituent steps, is considerably more complex than multiplication. Since a thorough, step-by-step derivation of the technique of binary division is largely academic to the context of this manual, only the steps significant to understanding the EAU hardware will be discussed. If further theoretical information is necessary, most computer design textbooks contain a discussion of binary division; EAU uses the method known as "restoring division". Paragraphs 5-151 through 5-175 of this section describe the important features of this technique.

5-151. DIVISION THEORY.

5-152. DECIMAL DIVISION. Figure 5-18 illustrates the conversion of the conventional form of division (box A) to a decimal equivalent of what EAU does (box C). In the conventional form, we first estimate how many times 13,500 will "go into" the leftmost digits of the dividend. The number of digits isolated for this partial dividend is also a trial-and-error guess, to some extent. Then we visually line up the quotient digit for a multiplication, which is performed to determine the next partial dividend. This next partial dividend consists of the remainder resulting from a subtraction of the exact value of 5 times the divisor from the former partial dividend, plus the next digit "brought down" from the dividend. This process continues until all digits of the dividend have been utilized as a partial dividend.

5-153. Thus it can be seen that the form of conventional division is unsuitable for use by a computer; there can be no preliminary estimating or visual aligning. The process must consist of a fixed number of simple, repetitive steps.

5-154. Box B illustrates one step in converting the division process to a machine form. Here, the process is viewed as a series of subtractions. That is, the problem is viewed in its simplest terms: how many times can 13,500 be subtracted from 79,000,000? This means, actually, that the divisor is a negative number.

5-155. The concept of isolating parts of the dividend as a partial dividend is retained, except that every digit is tried for a valid subtraction, one by one. There is no preliminary estimated positioning.

5-156. In step 1 of box B, the dividend and divisor are seen as existing in two adjacent registers. To bring the first digits into alignment for the first subtraction attempt, step 2 pre-shifts the divisor to the right. (Note in the conventional form that there is a also a shifting to the right.) Obviously, filling in assumed zeros, the divisor is too large to "go into" the dividend. Therefore, a non-significant zero is registered in the quotient (see bottom of box B), and the divisor is again shifted one place to the right.

5-157. In steps 3, 4, and 5 the divisor remains too large after each shift, and so each subtraction attempt adds one more zero to the quotient. Finally, in step 6, valid subtraction begins. It is seen that the divisor can be subtracted five times from the dividend; therefore the number 5 is registered in the quotient. These five subtractions reduce the dividend to 11,500,000.

5-158. Now the divisor is again shifted right (step 7), and again successive subtractions begin. This time the divisor can be subtracted 8 times, giving an 8 in the quotient and a new remaining dividend of 700,000.

5-159. In steps 8 and 9, five and one subtractions occur (respectively), producing a final quotient of 00005851 and a remainder of 11,500.

5-160. The process described above would place prohibitive demands on hardware. For example, the divisor, only 5 digits long, ultimately uses 13 register positions. Box 6 shows the form used by EAU, except that the example here is still in decimal numbers.

5-161. Box C assumes that there are two adjacent six-digit registers. Initially, the dividend occupies both of these registers as a 12-digit value. The six-digit divisor is in some easily accessible third register. Notice that the shifting is to the left instead of to the right as in conventional division. As explained in the multiplication example, shifting the partial product (or dividend in this case) in the opposite direction gives the same result as shifting the addend (or divisor) in the conventional direction. In both cases the reason for the reversal is hardware efficiency.

5-162. As in box B the first step is to pre-shift in preparation for the first subtraction trial. From then on, the process is repetitive through six cycles, one for each digit of the quotient. The quotient is shown shaded, gradually accumulating in register A. The dividend is correspondingly reduced from its initial 12-digit value to a 6-digit remainder in register B.

5-163. In step 1 the divisor is too large to subtract from register B. (Register A is not considered since the divisor is assumed to have trailing zeros.) Therefore nothing is subtracted from register B; i.e., it is restored to the value it had before the subtraction attempt. Now both register B and regis-

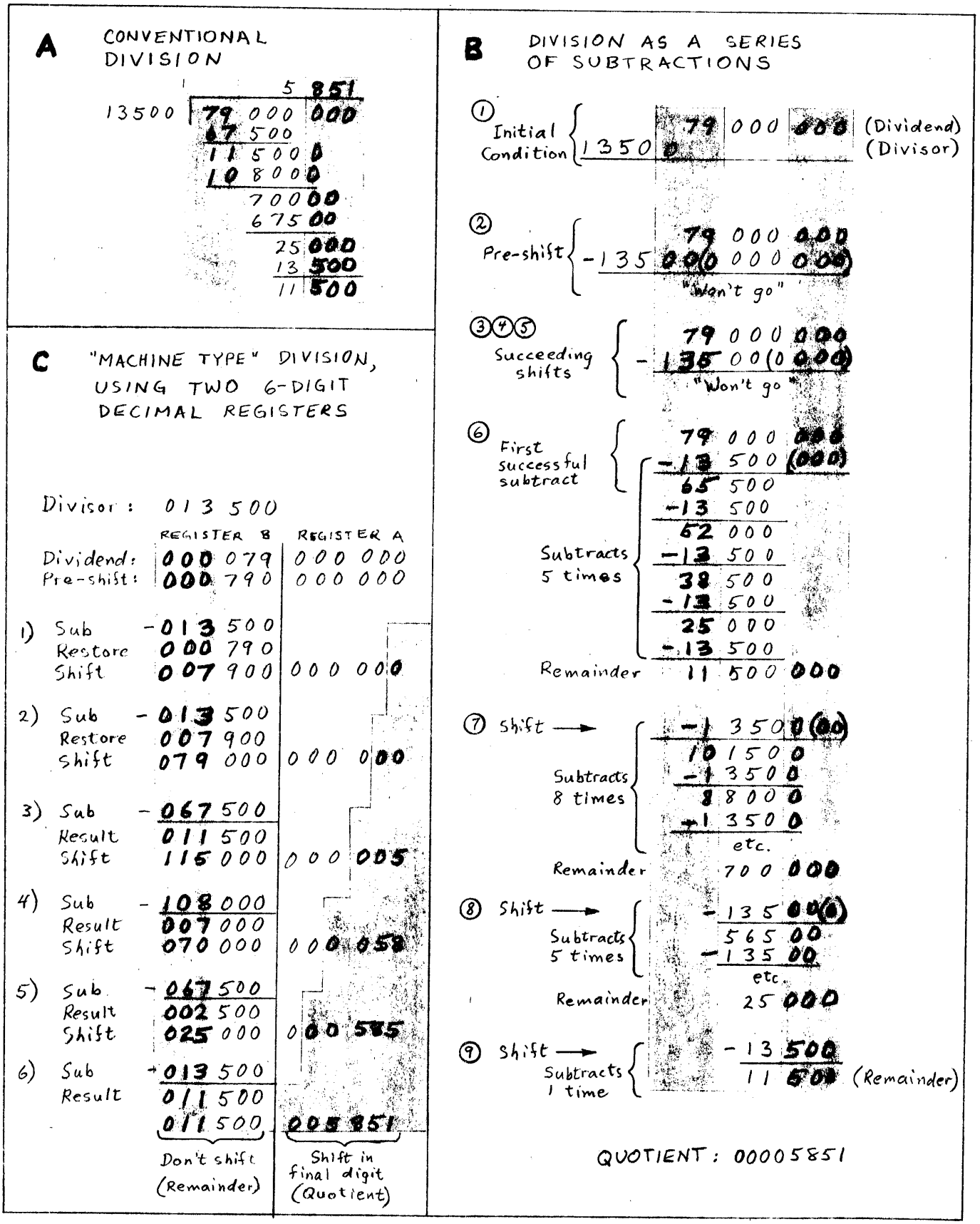


Figure 5-18. Decimal Division

ter A are shifted left one position, with the first quotient digit shifted into the least significant position. This quotient digit is 0 since the divisor subtracted zero times. The same thing happens in step 2.

5-164. Then, in step 3, the divisor subtracts five times (5 x 13,500 = 67,500), leaving a quantity of 11,500 in register B. When the registers are shifted left, the digit 5 is shifted into register A. Similarly, in steps 4 and 5, the divisor subtracts eight and five times, causing an 8 and a 5 to be shifted into register A.

5-165. The final step (step 6), however, is different, After performing the subtraction and determining the final quotient digit (1 in this case), register A is shifted whereas register B is not. This is because register A requires the shift in order to enter the final digit into the least significant digit position. But if register B were also shifted, the value of the remainder would incorrectly be increased by a factor of 10. The end result, as shown, leaves the quotient 5851 in register A and the remainder 11,500 in register B.

5-166. BINARY DIVISION. Figure 5-19 illustrates in binary form the same process of division shown in box C of figure 5-18. The numerical values of dividend, divisor, quotient, and remainder are the same as before. This is the sequence that EAU follows in executing the DIV instruction.

5-167. As before, the dividend initially occupies two registers and is now 32 binary digits long. The quotient is gradually shifted into the A-register, and the remainder will be left in the B-register. There is a pre-shift before the computation begins, and in the final step (16th) the quotient in the A-register is shifted (to shift in the final bit), whereas the remainder in the B-register is not shifted (to avoid incorrectly doubling its value).

5-168. The divisor, which initially resides in memory, must be converted to a negative number if it is not already negative. In the present example, the numbers are assumed to be positive; therefore the value is complemented and incremented before loading into the M-register. The process of subtracting divisor from dividend is therefore reduced to simply adding the B- and M-registers.

5-169. Also, due to the nature of the binary system, it is not necessary to count how many times the divisor can be subtracted, as was the case in the decimal example. The divisor will either subtract once or not at all; i.e., only a 1 or a 0 can be entered into the A-register as a quotient digit.

5-170. The decimal equivalence of each bit of the dividend is given in figure 5-19; adding the value indicated for each true bit of the binary value (1) produces the sum of 79,000,000. Similarly, the divisor, quotient, and remainder can be summed up in the same way.

5-171. The way that EAU detects whether or not the divisor will subtract is to check for a carry out of the last bit of the adder. (In the computer this signal is identified as C16,) In two's complement subtraction this bit will be true if the subtraction is valid. In this case the computer will complete the addition of the B- and M-registers, and a 1 is entered into the least significant bit position of the A-register. This 1 indicates that the divisor did subtract once, and it becomes part of the accumulating quotient. Otherwise, if there is no carry, the addition is aborted ("don't add"), and a 0 is entered into the quotient.

5-172. It happens in this example (since bits 12 and 13 of the two's complemented divisor are zeros) that bit 14 of the partial dividends in the B-register alone determines whether or not addition occurs. If this bit is a 1 (shown shaded in nine cycles), there will be a carry, so addition does occur in the succeeding cycle. However, remember that the carry out of bit 15 is the significant factor; looking at one bit only, as in this example, will not always be reliable.

5-173. When the division process begins, and the dividend is pre-shifted one position to the left, bit 14 of the B-register is 0. Therefore there is no addition in cycle 1. Cycle 1 simply shifts the two registers one place to the left, entering a 0 into the least significant bit position of the A-register. This 0 will become a leading zero in the final quotient.

5-174. Cycles 2 and 3 produce the same results. That is, bit 14 of the B-register determines that there is no addition, and zeros are shifted into the quotient. However, in cycle 3, bit 14 of the B-register is a 1 after shifting. Thus in cycle 4, the divisor in the M-register is added (actually subtracted since the divisor is negative) to the B-register. The result (remainder) is shown on the add line for cycle 4. Then both registers are shifted left, with a 1 shifted into the quotient. This 1 will become the most significant bit of the final quotient.

5-175. Thereafter, bit 14 of the B-register will be become a 1 eight more times, causing additions to occur in cycles 6, 7, 9, 10, 12, 13, 15, and 16. In the final cycle, as explained previously, the quotient receives its final shift, and the remainder is not shifted since its value is already correct. The numerical values of the end result are the same as in the earlier decimal example.

5-176. DIVIDE LOGIC.

5-177. DIV INSTRUCTION EXECUTION. The DIV instruction requires four cycles of initialization before actual division begins. This is one more than is required for multiply because divide checks for a possible overflow result before starting the division and because the double-length dividend requires more time for negative-positive conversion than the single-length multiplication quantities.

5-178. The DIV instruction requires that the divisor be in negative form and the dividend in positive form before the division begins. This is so the computer

PROBLEM

$$79{,}000{,}000 \div 13{,}500 = 5851 + \frac{11{,}500}{13{,}500}$$

DIVIDEND

(79,000,000)

DECIMAL EQUIVALENCE: SIGN, 1,073,741,824, 536,870,912, 268,435,456, 134,217,728, 67,108,864, 33,554,432, 16,777,216, 8,388,608, 4,194,304, 2,097,152, 1,048,576, 524,288, 262,144, 131,072, 65,536, 32,768, 16,384, 8192, 4096, 2048, 1024, 512, 256, 128, 64, 32, 16, 8, 4, 2, 1

0	0	0	0	0	1	0	0	1	0	1	1	0	1	0	1	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

B-REGISTER A-REGISTER

DIVISOR

(13,500)

IN MEMORY:

0	0	1	1	0	1	0	0	1	0	1	1	1	1	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SUBTRACTION REQUIRES NEGATIVE FORM OF DIVISOR (2's COMPLEMENT), OBTAINED AS FOLLOWS:

IN M-REGISTER:

COMPLEMENT	1	1	0	0	1	0	1	1	0	1	0	0	0	0	1	1
INCREMENT	1	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DIVISION PROCESS

		B-REGISTER					A-REGISTER					
PRE-SHIFT		0	000	100	101	101	010	1	110	001	110	000	000
1	ADD		Don't	Add									
	SHIFT	0	001	001	011	010	101	1	100	011	100	000	000
2	ADD		Don't	Add									
	SHIFT	0	010	010	110	101	011	1	000	111	000	000	000
3	ADD		Don't	Add									
	SHIFT	0	100	101	101	010	111	0	001	110	000	000	000
4	ADD	0	001	011	010	011	011						
	SHIFT	0	010	110	100	110	110	0	011	100	000	000	001
5	ADD		Don't	Add									
	SHIFT	0	101	101	001	101	100	0	111	000	000	000	010
6	ADD	0	010	010	110	110	000						
	SHIFT	0	100	101	101	100	000	1	110	000	000	000	101
7	ADD	0	001	011	010	100	100						
	SHIFT	0	010	110	101	001	001	1	100	000	000	001	011
8	ADD		Don't	Add									
	SHIFT	0	101	101	010	010	011	1	000	000	000	010	110
9	ADD	0	010	010	111	010	111						
	SHIFT	0	100	101	110	101	111	0	000	000	000	101	101
10	ADD	0	001	011	011	110	011						
	SHIFT	0	010	110	111	100	110	0	000	000	001	011	011
11	ADD		Don't	Add									
	SHIFT	0	101	101	111	001	100	0	000	000	010	110	110
12	ADD	0	010	011	100	010	000						
	SHIFT	0	100	111	000	100	000	0	000	000	101	101	101
13	ADD	0	001	100	101	100	100						
	SHIFT	0	011	001	011	001	000	0	000	001	011	011	011
14	ADD		Don't	Add									
	SHIFT	0	110	010	110	010	000	0	000	010	110	110	110
15	ADD	0	011	000	011	010	100						
	SHIFT	0	110	000	110	101	000	0	000	101	101	101	101
16	ADD	0	010	110	011	101	100						
	SHIFT	0	010	110	011	101	100	0	000	101	101	101	101

REMAINDER: 11,500

QUOTIENT: 5851

Figure 5-19. Binary Division

can perform the repetitive subtractions by adding, assuming that the divisor is the two's complement (negative) form. Therefore, if the dividend is negative or if the divisor is positive, these numbers must be converted (complemented and incremented). The dividend, which is assumed to be in the B- and A-registers, begins to be converted in the first cycle. The divisor, which is initially in memory; is converted as soon as it is brought out to the registers. This will occur in the third cycle (plus one cycle for each indirect addressing phase that may be used). After conversion, the divisor will reside in the M-register.

5-179. Each cycle of the DIV instruction is identified as divide cycle 1, divide cycle 2, etc., and is activated by a corresponding unique signal (D1, D2, etc.). Divide cycle 2 may be repeated several times if indirect addressing is used. Divide cycle 5 is invariably repeated eight times to achieve the required 16 add/shift operations (two per cycle). Thus D5 remains set for eight cycles before D6 activates the exit sequence. The exit cycle converts the quotient to negative form if the compared signs of the dividend and divisor indicate that the result should be negative. (Note, however, that the remainder always has the same sign as the dividend, which is the quantity we are subtracting from, regardless of the sign of the quotient.)

5-180. Figures 5-28 through 5-33, and tables 5-13 through 5-18 illustrate in detail the sequences of operations that execute the DIV instruction. These tables and figures should be folded out for convenient reference during the following discussion of each divide cycle. The logic diagrams, figures 7-2 and 7-4, should also be available.

5-181. DIVIDE CYCLE 1. As indicated in figure 5-28, there are four major operations to be accomplished in the first cycle of a DIV instruction. These are:

a. Initialize the operation cycle counter, and look for EAU group instruction signal MAC.

b. Decode the DIV instruction and set the Counter to 5 for DIV.

c. If the dividend is negative, begin the conversion to positive form by complementing the A-register half (the conversion will continue during the next two cycles).

d. Prepare EAU and the computer logic for reading memory in the next cycle (to get the address of the divisor).

5-182. Phase 1 begins by reading an instruction out of memory. This occurs during T0 through T2. Meanwhile, at T1, EAU clears its operation cycle counter (to the count of 15, as explained earlier in paragraph 5-86). By the latter part of T2, the computer instruction decoder has identified the instruction group. If the instruction is of the EAU group, a MAC signal enables EAU.

5-183. The MAC signal allows the T-register bits, which define the operation, to be clocked into the operation decoder at T3S. In the case of the DIV instruction, TR8 and TR11 cause the DIV flip-flop to be set. This begins the D1 signal. At T4 the counter is set to 5 as explained earlier in paragraph 5-88.

5-184. Also at T4, the B-register part of the dividend is read out to the R-bus in order to identify the sign bit. (The B-register contents are not altered.) The EOFB signal routes the data through the CPU adder to the T-bus. Then, if bit 15 of the T-bus is a "1" (meaning a negative dividend), the SDD flip-flop will be set. Thus the dividend sign is saved as an SDD signal for use at T5 and also in cycles D2, D3, and D6.

5-185. At T5 the A-register part of the dividend is read onto the R-bus by RARB and complemented by CMFB. However, this complemented form will be stored back in the A-register only if SDD had been set at T4. Thus if the number was positive, the value in A will remain not complemented.

5-186. At T6T7 the CPU is permitted to increment the P- and M-registers. (MAC enables OPO in the computer.) The effective signals are listed in the EQUATION column of table 5-13, but equations are not given since these are functions of the associated computer.

5-187. Then at T7S EAU disables the CPU by resetting the EPHX flip-flop, thus generating the IIR signal. At the end of T7S, which is the dividing line between cycles, the trailing edge of the clock signal advances the counter to the count of 6. This enables the D2 signal via gate U25C, thus activating divide cycle 2.

5-188. DIVIDE CYCLE 2. The second cycle, which may be repeated several times if indirect addresses are read, obtains the address of the divisor from memory. The P- and M-registers have been incremented for this purpose in the preceding cycle. Also, during this cycle, the process of converting a negative dividend continues by incrementing the A-register.

5-189. Reading from memory is enabled by generating the P123 signal, which remains active from T0 through T5 (terminated by T677). The address word is read into the T-register during T0 through T2.

5-190. During T3T4 the A-register is incremented by reading its contents onto the R-bus, forcing a "1" onto the S-bus (SB0) and adding the two together (ADF) onto the the T-bus. The result is stored back into the A-register only if SDD was set in the preceding cycle. If there is a carry out of bit 15 of the adder (C16), the Carry flip-flop will be set at T4. The Gate signal ensures that the incrementation occurs only once, in case D2 is repeated for indirect addressing. The Gate flip-flop is set at T6 of phase 1 and is reset at T6 of the next cycle if TB15 is true.

5-191. At T6T7 the divisor address in the T-register is transferred to the M-register, in preparation for

reading the divisor (or possibly another address if the current one is indirect) from memory in the next cycle. The T-register contents are read by RTSB, routed through the CPU adder by ADF, and stored in the M-register by SWSM. The Gate flip-flop will be cleared if TB15 causes a repeat of the D2 cycle.

5-192. At T7S the Clock signal to the counter goes true only if TB15 is "0", indicating a direct address. Otherwise, the counter stays at its present count (6), thus causing a repetition of the D2 cycle as a further search for a direct address. When the direct address is obtained, TB15 will be "0", and the counter will be clocked to the count of 7 at the trailing edge of T7S. This activates the D3 cycle.

5-193. DIVIDE CYCLE 3. The D3 cycle fetches the divisor from memory and completes the conversion of a negative dividend to positive form.

5-194. At T0 the P123 signal is again enabled in order to read memory. The divisor will be in the T-register by the end of T2 and will remain there until T6. Meanwhile, the B-register part of the dividend is converted to positive form if it was negative. At T3 the B-register contents are read and complemented, and then stored back in the B-register if SDD is true. At T4T5 the content of the Carry flip-flop is used to increment the B-register. This is a requirement of two's complementing two registers as if the contents were a single quantity. Carry may be a "0" or a "1", depending on whether there was a carry out of the A-register. (Refer to paragraph 5-190.) The B-register is read onto the R-bus by RBRB, and Carry forces a "1" or a "0" onto the S-bus (SB0), and ADF adds the two onto the T-bus; the result is stored back in the B-register if SDD is true.

5-195. During T6 and T7 the T- and P-register contents are moved as a precaution against the possibility of a DMA transfer destroying the T-register contents at the end of this cycle. At T6, the P-register contents are read onto the R-bus and stored into the M-register. Then at T7 the T-register is read onto the S-bus and stored into the P-register. The EOFB signal, which is high for both T6 and T7, routes both sets of data through the CPU adder to the T-bus. While the divisor is available on the T-bus during T7, sign bit TB15 is read and will set the SDV flip-flop if TB15 is a "1". Thus if SDV is set, the divisor is a negative number; if not set, the divisor is a positive number. At T7S the counter is advanced to the count of 8, thus enabling the D4 cycle.

5-196. DIVIDE CYCLE 4. The D4 cycle accomplishes three functions: converts a positive divisor to negative form (the need for this was explained earlier in paragraph 5-178); checks if an overflow will result if the division is attempted; and pre-shifts the dividend (Refer to paragraphs 5-156 and 5-162).

5-197. During T0 the divisor, presently in the P-register, is transferred to the T-register. In the process of doing so, the data is either complemented (if SDV is not set) or transferred unaltered (if SDV is set). This performs half of the conversion of positive divisor to negative form; the second half occurs during T2T3. During T1 the original contents of the P-register, presently in the M-register, are restored to the P-register.

5-198. During T2T3, the divisor is transferred from the T-register to the M-register. In the process of doing so, the data is either incremented (by a forced RB0 if SDV is not set), or transferred unaltered (if SDV is set). Thus the end result guarantees a negative number as the divisor in the M-register.

5-199. While the divisor is available on the T-bus during T3, the first overflow check is made. If TB15 is a "0", this indicates that the divisor was all-zero, and the OVR flip-flop is set. (If the divisor was zero, complementing would cause all-ones, and incrementing would cause roll-over back to zero.)

5-200. The second overflow check is made at T4T5. The B- and M-registers (most significant half of the dividend, and the divisor) are added together as a trial subtraction. The result is not stored. If, however, bit 15 is not a "1", as it should be for two's complement subtraction, this indicates that the dividend is too large for the divisor. A "0" as TB15 will cause the OVR flip-flop to be set at T5S.

5-201. Incidentally, if the OVR flip-flop had been set due to a zero divisor, the clock at T5S (for the second check) might cause OVR to toggle back to the reset state. Gate U92A uses the OVR signal to retain the set state during T5 to avoid this possibility.

5-202. After the OVR flip-flop is set from either cause (paragraph 5-199 or 5-200), the Exit flip-flop will also be set, at T5S. This aborts the DIV instruction by immediately beginning the exit sequence (paragraph 5-218).

5-203. Assuming that no overflow was indicated, the DIV instruction execution continues at T6 by shifting the A- and B-register contents one bit position to the left. The SLMB and SL14B signals remain high throughout T6 and T7, enabling the A-register to be shifted during T6 and the B-register during T7. The content of the Carry flip-flop, which might have been set in the D2 cycle, is read onto bit 0 of the T-bus and thus becomes the least significant bit in the A-register. The Link flip-flop is used to save the bit shifted out of bit 15 of the A-register, so that it can be moved into bit 0 of the B-register at T7. Since the Link flip-flop is clocked by the TEV (Time Even) signal, bit 15 is also read at earlier times in the cycle; however, only the reading at T6 is significant since the content is used only at T7. Bit 15 of the B-register, the original sign bit, is discarded.

5-204. At T7S the counter is advanced to the count of 9, thus enabling the D5 cycle.

5-205. DIVIDE CYCLE 5. Each D5 cycle accomplishes two subtract/shift operations. Therefore 8 complete D5 cycles are necessary in order to obtain the 16 subtract/shift operations required by a 16-bit divisor register.

5-206. The first subtract/shift operation occurs during T0 through T3. At T0T1, the B- and M-register contents are read and added together onto the T-bus. If the addition does not result in a carry out of bit 15 of the adder, this means that the subtraction is not valid (divisor larger than dividend bits). In this case the result is not stored, the B-register retains its present value, and the Carry flip-flop is cleared. However, if the addition does result in a carry, the result is stored in the B-register and the Carry flip-flop is set ("1").

5-207. At T2, the A-register is read onto the R-bus and shifted left one position onto the T-bus. At the same time, the Carry flip-flop content is read onto bit 0 of the T-bus. Then the 16 shifted bits are stored back into the A-register. Meanwhile, the original bit 15 of the A-register (RB15) is clocked into the Link flip-flop. This bit will become bit 0 of the B-register.

5-208. At T3 the B-register is shifted. Its least significant bit is lost, but the saved bit from the A-register (now in the Link flip-flop) is read onto the T-bus (TB0). The shifted 16-bit result is stored back into the B-register.

5-209. The second subtract/shift operation of the D5 cycle occurs during T4 through T7. The sequence of active signals is identical to those occurring from T0 through T3. However, the shift B operation (refer to equations) introduces the D5L8 term; this is for the purpose of omitting this shift on the last (8th) loop. Otherwise, for the first seven loops, the second subtract/shift operation is the same as described in paragraphs 5-206 through 5-208; the only difference is that the timing signals are advanced by four time periods (i.e., T0T1 becomes T4T5, etc.).

5-210. Note that since the Link flip-flop is clocked by TEV, insignificant readings of RB0 occur at T0 and T4. Only those readings that occur at T2 and T6 are actually used.

5-211. At the trailing edge of each T7S the counter advances, first from 9 to 10, then 10 to 11, and so on, until the count rolls over from 15 to 0. The count of 0 indicates that this is the last loop, and accordingly disables the D5L8 signal. Thus the final shift of the B-register at T7 is omitted since B is neither read nor stored. This leaves the remainder unaltered (refer to paragraphs 5-165, 5-167 and 5-175.) Then the count of 1, at the end of the eighth loop, activates the D6 cycle. The quotient now exists in the A-register and the remainder in the B-register. Both are in positive form.

5-212. DIVIDE CYCLE 6. The D6 cycle converts the quotient and/or remainder to negative form (if required), checks for quotient overflow, and performs the exit sequence. The reason why possible overflow is again checked is that the earlier precheck (in cycle D4) ascertained only that the answer could be contained in 16 bits. The final checks in the D6 cycle ensure that the answer is correct for 15 bits plus sign.

5-213. The combined outputs of the SDD and SDV flip-flops are connected in an "exclusive-or" gate arrangement. This gating will produce a Sign signal if either, but not both, of the flip-flops are set. This "exclusive-or" condition is consistent with the mathematical rule that declares the quotient will be negative if either the dividend or the divisor (exclusively) was negative. Thus if the Sign signal is true, the quotient must be converted from the present positive form to its equivalent negative form.

5-214. At T0 the A-register is read and complemented, then stored back in A if SIGN is true. At T1T2 the A-register is incremented by adding with a forced SB0, and stored if SIGN is true. The quotient is now in its final form.

5-215. While the quotient is on the R- and T-buses during T2, the final overflow check is made. If Sign is low, indicating that a positive answer is expected (and therefore the non-incremented value on the R-bus is the final answer), RB15 is checked. If RB15 is a "1", implying a negative result, an overflow has occurred, and the OVR flip-flop will be set.

5-216. If Sign is high, indicating that a negative answer is expected (and therefore the incremented value on the T-bus is the final answer), TB15 and C16 are checked. If both TB15 and C16 are "0", an overflow has occurred, and the OVR flip-flop will be set. The C16 bit is included in the check to avoid an erroneous overflow indication for a legitimate zero quotient (fractional result). In this case TB15 (and the rest of the T-bus) will be "0", and C16 will be "1" as a result of the complement and increment.

5-217. During T3 through T5 the remainder (independent of the quotient's sign) will be converted to negative form if the original dividend was negative. (Refer to paragraph 5-179.) At T3 the B-register contents are read and complemented, than stored back in B if the SDD flip-flop is set. At T4T5 the B-register is again read, then incremented by adding a forced SB0, and the final result stored back in the B-register. The remainder is now in its final form.

5-218. At T5 the Exit flip-flop is set, thus starting the exit sequence. First, during T6T7 the computer Overflow flip-flop is either cleared or set, depending on whether or not the OVR flip-flop was set at T2S. The CLF and IOSB signals will clear Overflow; the STF and IOSB signals will set Overflow.

5-219. Also during T6T7, the P-register is incremented by adding to a forced SB0, and the result is stored in both P- and M-registers. This prepares the computer to read its next instruction in the following phase 1 cycle.

5-220. At T7S the EPHX flip-flop is set again, thus terminating the IIR signal. This enables the computer instruction register and phase control logic. At T7 the operation decoder is clocked, and since there are presently no T-register inputs, the DIV flip-flop is now cleared. This completes the DIV sequence of operations.

5-221. DOUBLE LOAD.

5-222. The purpose of the DLD (Double Load) instruction is to transfer two consecutive memory words into the A- and B-registers. The instruction is executed in a minimum of four machine cycles (one cycle added for each level of indirect addressing which may be used). The first two cycles read the double-word instruction out of memory (instruction and operand address), and the last two cycles read the double-word operand out of memory (to the A- and B- registers).

5-223. The four double load cycles are identified as DL1, DL2, DL3, and DL4, activated (respectively) by counts 5, 6, 7, and 8 of the operation cycle counter. The following descriptions, through paragraph 5-234, describe the operations occurring in each of the four cycles. Figures 5-34 and 5-35 and tables 5-19 and 5-20 illustrate the sequences of operations that execute the DLD instruction. These figures and tables should be folded out for convenient reference during the following discussions. The logic diagrams, figures 7-2 and 7-4, should also be available.

5-224. DOUBLE LOAD CYCLE 1. Figure 5-34 shows both the DL1 and DL2 cycles. The DL1 cycle includes the upper two blocks. (Actually, since the type of instruction is not known until the end of T3 in phase 1, the operations unique to DLD do not begin until T3S.) Phase 1 begins to read the instruction out of memory during T0. Meanwhile, at T1, EAU clears its operation cycle counter (to the count of 15, as explained earlier in paragraph 5-86). By the latter part of T2, the computer instruction decoder has identified the instruction group. If the instruction is of the EAU group, a MAC signal enables EAU.

5-225. The MAC signal, which will be true for DLD, allows the T-register bits that define the DLD instruction (see equation) to be clocked into the operation decoder at T3S. This causes the DLD flip-flop to be set.

5-226. The DLD signal enables P123, which will remain high throughout all four cycles since memory will be addressed in every cycle. At T4 the counter is set to 5, as explained earlier in paragraph 5-88.

5-227. Nothing further happens in EAU until the end of T7. However, during T6T7, the CPU is allowed to increment the P- and M-registers. (MAC enables OPO (One Phase Operation) which increments P and M at the end of Phase 1.) Then at T7S EAU disables the CPU by resetting the EPHX flip-flop, thus generating the IIR signal. At the trailing edge of T7S, the counter is advanced to the count of 6. This starts the DL2 cycle. (Note that specific DL1 and DL2 signals are not necessary, as the operations occurring during these two cycles can be accomplished by other available signals.)

5-228. DOUBLE LOAD CYCLE 2. The DL2 cycle (which may be repeated several times if indirect addresses are read) obtains the address of the operand from memory. The P- and M-registers have been incremented for this purpose in the preceding cycle.

5-229. During T0 through T2 memory reads the address word into the T-register. Then during T6T7 the T-register contents are read onto the S-bus by RTSB, routed through the CPU adder by ADF, and stored in the M-register by SWSM. The MD2 (Multiply/Divider cycle 2) signal is used to activate these signals since the purpose is similar to the fetching of addresses by the MPY and DIV instructions.

5-230. At T7S the clock signal to the counter goes true only if TB15 is "0", indicating a direct address. Otherwise, the counter stays at its present count (6), thus causing a repetition of the DL2 cycle as a further search for a direct address. When the direct address is obtained, TB15 will be "0", and the counter will be clocked to the count of 7 at the trailing edge of TS. This activates the DL3 cycle.

5-231. DOUBLE LOAD CYCLE 3. Figure 5-35 shows both the DL3 and DL4 cycles. The DL3 cycle (upper block in the figure) fetches the first data word and loads it into the A-register.

5-232. During T0 through T2 memory reads the data word into the T-register. Then at T4 the RTSB, EOFB, and SWSA signals transfer the T-register contents to the A-register. This completes the first half of the double load. Now the M-register is incremented (during T6T7) in order to address the next data word in memory. Incrementing is accomplished by reading the M-register onto the S-bus, forcing a "1" on the R-bus (R80), adding the two together onto the T-bus, and storing the result back into the M-register. At T7S, the counter is advanced to the count of 8, thus enabling the DL4 cycle.

5-233. DOUBLE LOAD CYCLE 4. The DL4 cycle fetches the second data word and loads it into the B-register. During T0 through T2 the word is read into the T-register, and at T4 the RTSB, EOFB, and SWSB signals transfer it to the B-register. This completes the double load operation. At T5 the Exit flip-flop is set, beginning the sequence which returns control to the CPU.

5-234. During T6T7, the P-register is incremented by reading its contents onto the R-bus and adding with a forced "1" on the S-bus (SB0). The result on the T-bus is stored into both the P- and M-registers. This gives the computer the address of its next instruction. Then at T7S the Exit signal clears the IIR signal by setting the EPHX flip-flop and also clears the DLD flip-flop by clocking the operation decoder (which has no T-register inputs at this time). With DLD low, the P123 signal is also terminated.

5-235. DOUBLE STORE.

5-236. The DST (Double Store) instruction transfers the contents of the A- and B-registers to two consecu-

tive memory locations. The instruction is executed in a minimum of four machine cycles (one cycle added for each level of indirect addressing which may be used). The first two cycles read the double-word instruction out of memory (instruction and operand address), and the last two cycles transfer the A- and B-register contents to memory.

5-237. The four double store cycles are identified as DS1, DS2, DS3, and DS4, activated (respectively) by counts 5, 6, 7, and 8 of the operation cycle counter. Paragraphs 5-238 through 5-249 describe the operations occurring in each of the four cycles. Figures 5-36 and 5-37 and tables 5-21 and 5-22 illustrate the sequences of operations that execute the DST instruction. These figures and tables should be folded out for convenient reference during the following discussions. The logic diagrams, figures 7-2 and 7-4, should also be available.

5-238. DOUBLE STORE CYCLE 1. Figure 5-36 shows both the DS1 and DS2 cycles. The DS1 cycle includes the upper two blocks. (Actually, since the type of instruction is not known until the end of T3 in phase 1, the operations unique to DS1 do not begin until T3S.) Phase 1 begins to read the instruction out of memory during T0. Meanwhile, at T1 EAU clears its operation cycle counter (to the count of 15, as explained earlier in paragraph 5-86). By the latter part of T2, the computer instruction decoder has identified the instruction group. If the instruction is of the EAU group, a MAC signal enables the EAU option.

5-239. The MAC signal allows the T-register bits which define the DST instruction (see equation) to be clocked into the operation decoder at T3S. This causes the DST flip-flop to be set.

5-240. The DST signal enables P123, which will remain high throughout all four cycles since memory will be addressed in every cycle. At T4 the counter is set to 5, as explained earlier in paragraph 5-88. Nothing further happens in EAU until the end of T7.

5-241. However, during T6T7 the CPU is allowed to increment the P- and M-registers. (MAC enables OPO (One Phase Operation) which increments P and M at the end of phase 1.) Then at T7S, EAU disables the CPU by resetting the EPHX flip-flop, thus generating the IIR signal. At the trailing edge of T7S, the counter is advanced to the count of 6. This starts the DS2 cycle. (Note that specific DS1 and DS2 signals are not necessary, as the operations occurring during these two cycles can be accomplished by other available signals.)

5-242. DOUBLE STORE CYCLE 2. The DS2 cycle (which may be repeated several times if indirect addresses are read) obtains the address from memory that will specify where to store the data. The P- and M-registers have been incremented for this purpose in the preceding cycle.

5-243. During T0 through T2 memory reads the address word into the T-register. Then during T6T7 the T-register contents are read onto the S-bus by RTSB, routed through the CPU adder by ADF, and stored in the M-register by SWSM. The MD2 (Multiply/Divide cycle 2) signal is used to activate these signals since the purpose is similar to the fetching of addresses by the MPY and DIV instructions.

5-244. At T7S the Clock signal to the counter goes true only if TB15 is "0", indicating a direct address. Otherwise, the counter stays at its present count (6), thus causing a repetition of the DS2 cycle as a further search for a direct address. When the direct address is obtained, TB15 will be "0", and the counter will be clocked to the count of 7 at the trailing edge of T7S. This activates the DS3 cycle.

5-245. DOUBLE STORE CYCLE 3. Figure 5-37 shows both the DS3 and DS4 cycles. The DS3 cycle (upper block in the figure) reads the A-register and transfers the contents to memory.

5-246. At T0 an ISG signal is generated, which will remain high throughout these last two cycles of the DST instruction. The purpose of ISG is to prevent reading memory contents into the T-register during the normal read-memory cycle time (T0 through T2). Instead, at T2 the A-register contents are transferred into the T-register in preparation for writing into memory (at T3 through T5). The RARB, EOFB, and SWST signals accomplish the A- to T-register transfer. The normal write cycle of the computer then stores the data into memory.

5-247. Now the M-register is incremented (during T6T7) in order to address the next consecutive location. Incrementing is accomplished by reading the M-register onto the S-bus, forcing a "1" onto the R-bus (RB0), adding the two together onto the T-bus, and storing the result back into the M-register. At T7S the counter is advanced to the count of 8, thus enabling the DS4 cycle.

5-248. DOUBLE STORE CYCLE 4. The DS4 cycle reads the B-register and transfers the contents to the location now addressed by the M-register. At T2 the RBRB, EOFB, and SWST signals transfer the B-register contents to the T-register. Then, during T3 through T5 the computer writes the T-register contents into memory. This completes the double store operation. At T5 the Exit flip-flop is set, beginning the sequence which returns control to the CPU.

5-249. During T6T7 the P-register is incremented by reading its contents onto the R-bus and adding with a "1" on the S-bus (SB0). The result on the T-bus is stored into both the P- and M-registers. This gives the computer the address of its next instruction. Then at T7S the Exit signal clears the IIR by setting the EPHX flip-flop and also clears the DST flip-flop by clocking the operation decoder (which has no T-register inputs at this time). With DST low, the P123 signal is also terminated.

5-250. SHIFTS AND ROTATES.

5-251. GENERAL.

5-252. There are six "shift/rotate" instructions in the EAU group. The purpose of these instructions is to shift the combined contents of the A- and B-registers right or left in one of three conventional modes of shifting: arithmetic shifting, logical shifting, or rotation. The number of bit positions of shift, from 1 to 16, is specified by four bits of the instruction word. (Refer to section III.) Since there is no reference to memory, the instruction is contained in one 16-bit word, rather than a double-word as for MPY, DIV, DLD, and DST.

5-253. The process of shifting is simply a matter of repetitively moving the data bits, in parallel, left or right in the registers, one bit position per computer time period, until the specified number of shifts have been achieved. Since there are two registers, each is shifted in alternate time periods, and a complete shift of the two registers therefore occupies two time periods (typically 3.2 or 4.0 microseconds, depending on computer clock timing).

5-254. Understanding the shift/rotate logic requires an understanding of the effects of the shift signals and the usage of the Link and CARX (Carry) flip-flops. Figure 5-20 illustrates the effects accomplished by the three shift control signals, and figure 5-21 shows how these signals and the two linking flip-flops are used in each of the six instructions.

5-255. As shown in figure 5-20, there are two left-shifting signals and one right-shifting signal. The right-shifting signal, SRMB (Shift Right Magnitude, Buffered), shifts bits 1 through 15 of the data on the R-bus to bit positions 0 through 14 on the T-bus.

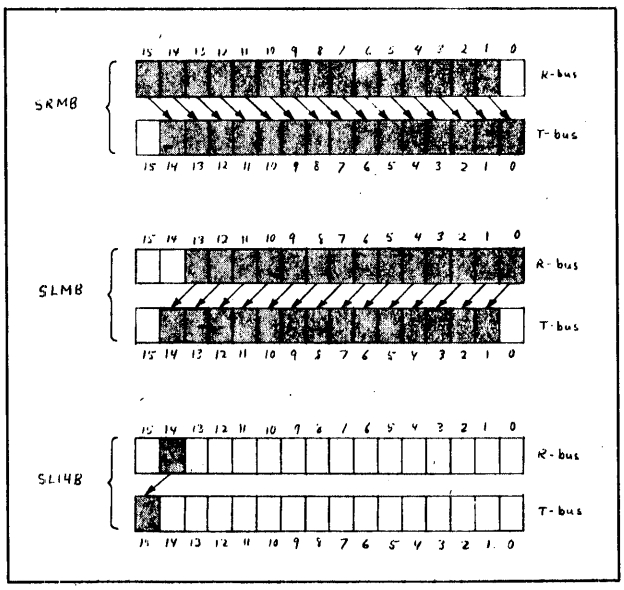


Figure 5-20. Effects of Shift Signals

Since the normal succeeding action is to store the T-bus contents into the A- or B-register, bit 0 of the R-bus will be lost unless a separate action saves this bit (RB0) in the Link or CARX flip-flop. Also, unless some specific action is taken regarding bit 15 of the T-bus, TB15 will be a "0".

5-256. When shifting left, bit 14 of the R-bus would automatically move into bit 15 of the T-bus. In the case of an arithmetic left shift (ASL), this is not desired. Thus bit 14 of the R-bus is separately controlled by SL14B (Shift Left Bit 14, Buffered). The SLMB signal (Shift Left Magnitude, Buffered) shifts bits 0 through 13 of the R-bus into bits 1 through 14 of the T-bus. Unless separate actions are taken regarding bits 0 and 15 of the T-bus, these bits will be "0". The SL14B signal, if used, shifts bit 14 of the R-bus onto bit 15 of the T-bus.

5-257. Figure 5-21 compares the three different types of shifts, showing both the right and left versions of each. Some general notes about the figure: in each case, the B-register is on the left, the A-register on the right. The brackets for SRMB and SLMB enclose R-bus bits to be shifted (rather than post-shift T-bus bits). In each case, the shaded register is the first to be shifted; this occurs during even time periods (T0, T2, T4, or T6). The unshaded register is shifted next, during odd time periods (T1, T3, T5 or T7).

5-258. In the arithmetic shifts, ASR and ASL, the sign bit (bit 15 of the B-register) must remain unaltered. This is accomplished by gating RB15 to TB15 when the B-register is shifted. When shifting right, the B-register is shifted first, during TEV (Time Even), and bit 0 is saved in the Link flip-flop. During TOD (Time Odd) the A-register is shifted. At the same time, the Link content is gated to TB15 so that the former bit 0 of the B-register becomes bit 15 of the A-register. When shifting left, the A-register is shifted first, during TEV, and bit 15 is saved in the Link flip-flop. Since nothing is done about bit 0, it will be a "0". During TOD the B-register is shifted left (all except bit 14, the only case where this is true), while the Link flip-flop content is shifted into bit 0.

5-259. The logical shifts, LSR and LSL, are simpler than the arithmetic shifts since the sign bit does not require special treatment. A "0" is shifted into bit 15 of the B-register when shifting right (by simply not gating anything to TB15), and similarly a "0" is shifted into bit 0 of the A-register when shifting left (no gating to TB0).

5-260. The rotates require use of both the Link and CARX flip-flops since there is an end-around shift involved. When rotating, no bits are lost, and no "0"s are entered. Before rotation begins, the CARX flip-flop is initially loaded with a copy of A0 (when rotating right) or B15 (when rotating left). Then during each TEV, CARX is shifted into the register being shifted, while Link receives the bit shifted out of the other end of the register. During TOD, the reverse is true: the Link content enters one end of the register being shifted, while CARX receives the

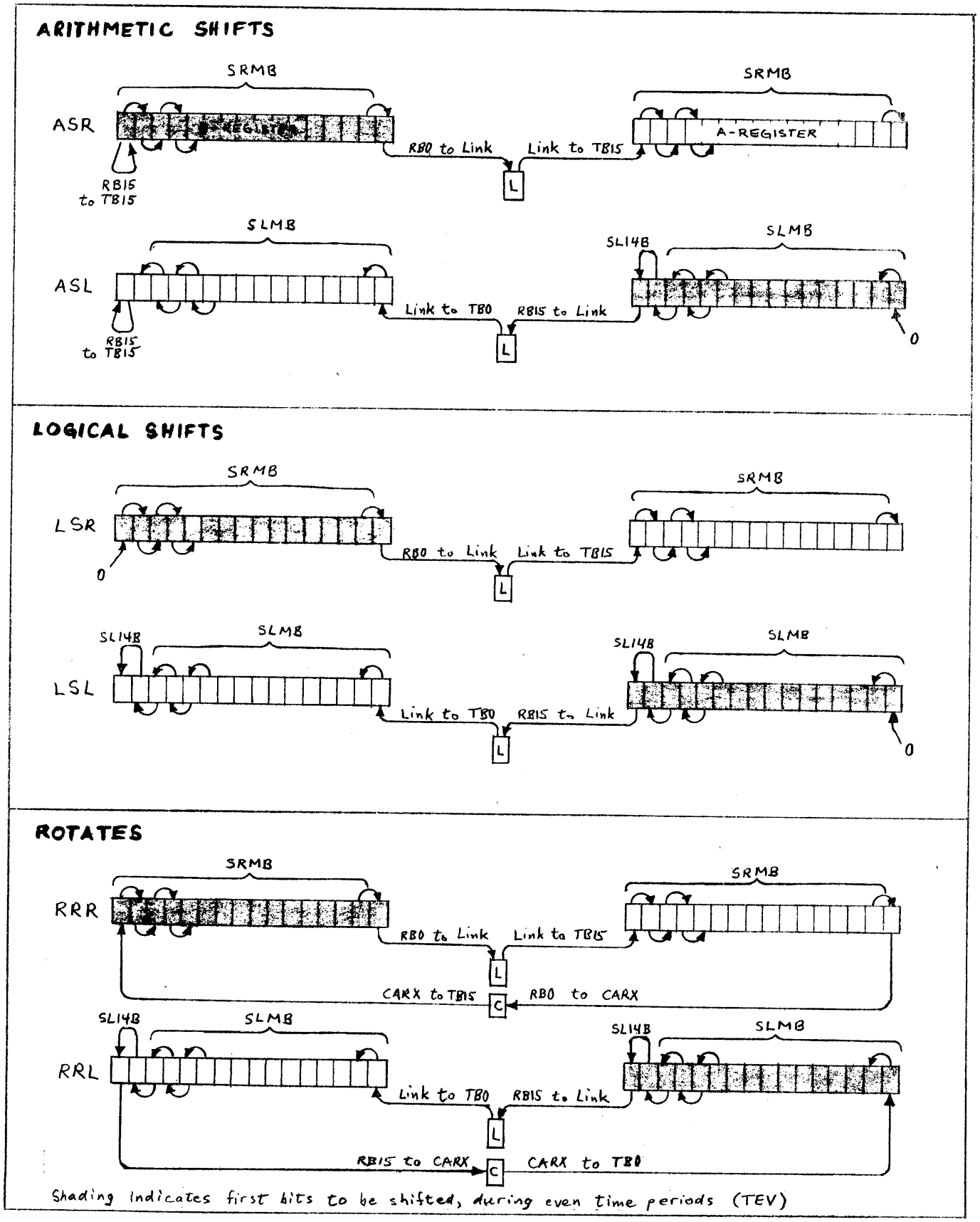


Figure 5-21. Shift Operations

bit shifted out of the other end. The CARX content at the end of TOD, therefore, is a preparation for the next rotate, if there is to be one.

5-261. The following paragraphs (to the end of this section) describe the logic which executes the three types of shifts. Figures 5-38 through 5-43 and tables 5-23 through 5-28 illustrate the operations involved. These tables and figures should be folded out for convenient reference during the following discussions. The logic diagrams, figures 7-2 and 7-4, should also be available. As background information on the count sequences for shifts and rotates, paragraphs 5-87 and 5-89 should be reviewed.

5-262. ARITHMETIC SHIFTS.

5-263. As shown in figure 5-38, phase 1 begins to read the instruction out of memory during T0. Memory reading continues through T2. Meanwhile, at T1 EAU clears its operation cycle counter (to the count of 15, as explained in paragraph 5-86). By the latter part of T2, the computer instruction decoder has identified the instruction group. If the instruction is of the EAU group, a MAC signal enables EAU. The MAC signal allows TR4 to be clocked into the operation decoder at T3S. This causes the AS flip-flop to be set.

5-264. Since an ASL operation could cause an overflow condition (the only shift instruction which may do so), the CPU Overflow flip-flop is cleared at T3S by generating CLF and IOSB. Then at T4 T-register bits 0 through 3 are gated into the operation cycle counter to set the counter to the complement of the number of desired shifts. (Refer to paragraph 5-87.)

5-265. Changing the count value to some value other than the clear count (15) automatically enables the CTO0 line. With AS, this enables the SRCS (Shift-Rotate Counted Started) signal. The SRCS is the primary signal that enables most of the shift logic gates on the EAU logic card. However, if the desired shift count is 16, TR0 through TR3 will be all zeros (table 5-5), so the counter will retain its cleared count (15). In this case it is necessary to enable SRCS by some other means for the first shift until the count gets started. This is done by "anding" the reset outputs of the IIRX and EPHX flip-flops (gate U72D). This temporarily enables the CTO0 line, from T6 (when the EPHX flip-flop is cleared; refer to next paragraph) until the end of T7S (when the IIRX flip-flop is set).

5-266. At T5S, EAU disables the CPU by resetting the EPHX flip-flop, thus generating the IIR signal. This is earlier than occurs for the MPY, DIV, DLD, and DST instructions, which disable the CPU at T7S. The reason for the earlier disabling in the case of the shifts and rotates is to prevent the CPU from incrementing the P- and M-registers during T6T7. Since the shift-rotate instructions are not double-word instructions, no further reference to memory will be made until the execution is completed. The P- and M-registers will be incremented only once, at the end of the last cycle.

5-267. During T6T7 of phase 1 the first shift is executed. Table 5-23 shows the equations for the right shift, and table 5-24 shows the equations for the left shift. The TEV operations occur during T6, and the TOD operations occur during T7.

5-268. When shifting right, the B-register is first read out to the R-bus by RBRB (during TEV), and bits 1 through 15 are shifted right by SRMB to TB0 through TB14. RB15 is gated to TB15, and the result is stored back in the B-register. Meanwhile, RB0 is clocked into the Link flip-flop. Then, during TOD the A-register is read and shifted by RARB and SRMB, Link is gated to TB15, and the result is stored back in the A-register.

5-269. When shifting left the A-register, during TEV, is first read out by RARE, bits 0 through 14 are shifted to TB1 through TB15 by SLMB and SL14B, and TB0 takes the value of "0". The result is stored back in the A-register, and RB15 is saved in the Link flip-flop. Then, during TOD only bits 0 through 13 of the B-register are shifted to the T-bus (TB1 through TB14) while RB15 is gated to TB15. SL14B is not enabled, in this case. The result is stored back in the B-register by RBRB.

5-270. If RB14 and RB15 differ during TOD, the shift will cause a significant data bit to be lost ("1" for positive numbers, "0" for negative numbers.) This is an error condition and is so indicated by having the "exclusive-or" of RB14 and RB15 generate an OASL (Overflow due to Arithmetic Shift Left) signal.

5-271. At the end of TOD the counter is incremented. If the count is less than 15, the shifting will continue in the next cycle. If the count is 15 (meaning only one shift was desired), the second cycle is still necessary in order to provide the exit sequence.

5-272. When the next cycle begins (see figure 5-39), the shifting becomes repetitive, with one shift occurring during T0T1, the next in T2T3, then T4T5, and T6T7. The active signals are exactly the same as described above (paragraphs 5-268 through 5-271). Before each shift begins, a check of the current count is made. If the count is 15, CTO0 goes low, thus inhibiting SRCS and preventing any further shifts. Also, if OASL is true, the computer Overflow flip-flop is set by enabling STF and IOSB.

5-273. The Exit flip-flop is set at T5 if the count is either 14 or 15. This is accomplished by not using the least significant bit of the binary count value in the exit gating. The reason is to permit one last shift to occur during T4T5, while simultaneously setting the Exit flip-flop at T5. Since the actual exit sequence does not begin until T6, the parallel operations permit three full shifts plus exit in any cycle.

5-274. The exit sequence during T6T7 is the same as for all previous instructions. The P-register is read onto the R-bus by RPRB, incremented by SB0 and ADF, and stored into the P- and M-registers. At T7S the CPU is again enabled, by setting the EPHX flip-flop and thus terminating the IIR signal.

5-275. LOGICAL SHIFTS.

5-276. Figures 5-40 and 5-41 and tables 5-25 and 5-26 illustrate the operations for the logical shifts, LSR and LSL. The operations are exactly as described for the arithmetic shifts, paragraphs 5-262 through 5-274, with the following exceptions:

a. The sign bit is not restored (omit RB15 gating to TB15 in paragraphs 5-268 and 5-269).

b. When shifting the B-register left, SL14B is used (instead of inhibited, as in paragraph 5-269).

c. There is no overflow checking (omit paragraph 5-270, and the references to overflow in paragraph 5-264 and 5-272).

d. The LS flip-flop is set by TR5 (cf AS by TR4 in paragraph 5-263).

5-277. ROTATES.

5-278. Figures 5-42 and 5-43 and tables 5-27 and 5-28 illustrate the operations for the rotate instructions, RRR and RRL. These operations are similar in most respects to the previous description of the arithmetic shifts, paragraphs 5-262 through 5-274, and so is somewhat abbreviated. Where appropriate, references to earlier descriptive details are given.

5-279. As usual, the instruction is read out of memory during T0 through T2, while the operation cycle counter is cleared at T1. The MAC signal, decoded during T2, enables EAU by allowing TR6 to set the RO flip-flop at T3S. At T4 the counter is set to the complement of the desired shift count, thus enabling CTO0 and consequently SRCS. This enables the shifts. (See comment in paragraph 5-265 regarding initial shift if the desired shift count is 16.)

5-280. During T5 of phase 1 of the rotate instructions, an ROT5 signal is generated. The purpose of this signal is to enable the presetting of the CARX flip-flop. The need for this was explained earlier in paragraph 5-260. The presetting is accomplished by reading either the A-register (for right rotation) or the B-register (for left rotation) onto the R-bus. Then the value of RB0 (right) or RB15 (left) is loaded into the CARX flip-flop. Since no "store" operation occurs, the A- and B-registers retain their current values.

5-281. At T5S the IIR signal is generated, thus disabling the CPU. During T6T7 the first rotation occurs. Table 5-27 shows the equations for the right rotate, and table 5-28 shows the equations for the left rotate. The TEV operations occur during T6, and the TOD operations occur during T7.

5-282. When rotating right, the B-register is first read out to the R-bus by RBRB (during TEV), and bits 1 through 15 are shifted right by SRMB to TB0 through TB14. The content of the CARX flip-flop (which was preset to have the value of A0) is gated onto TB15. The result is stored back in the B-register. Meanwhile, RB0 is clocked into the Link flip-flop. Then, during TOD the A-register is read and shifted by RARB and SMRB, Link is gated to TB15, and the result is stored back in the A-register. In preparation for the next rotate, TB0 is saved in the CARX flip-flop.

5-283. When rotating left, the A-register is read out by RARB during TEV, and bits 0 through 14 are shifted to TB1 through TB15 by SLMB and SL14B. The content of the CARX flip-flop (which was preset to have the value of B15) is gated onto TB0. The result is stored back in the A-register. Meanwhile, RB15 is clocked into the Link flip-flop. Then, during TOD the B-register is read onto the R-bus, and SLMB and SL14B shift bits 0 through 14 to T-bus bits 1 through 15. The content of the Link flip-flop is gated onto TB0, and the result is stored back in the B-register. In preparation for the next rotate, TB15 is saved in the CARX flip-flop.

5-284. At the end of TOD the counter is incremented. If the count is less than 15, the rotations will continue in the next cycle. If the count is 15, the next cycle will simply provide the exit sequence.

5-285. Assuming that additional rotations have been coded in the instruction, the rotate operations become repetitive in the next cycle. Four rotations can occur: during T0T1, T2T3, T4T5, and T6T7. (See figure 5-43.) The active signals are exactly as described above (paragraphs 5-282 through 5-284). Before each rotate begins, a check of the current count is made. If the count is 15, CTO0 goes low, thus inhibiting SRCS and preventing any further shifts. The exit sequence is the same as described in paragraphs 5-273 and 5-274.

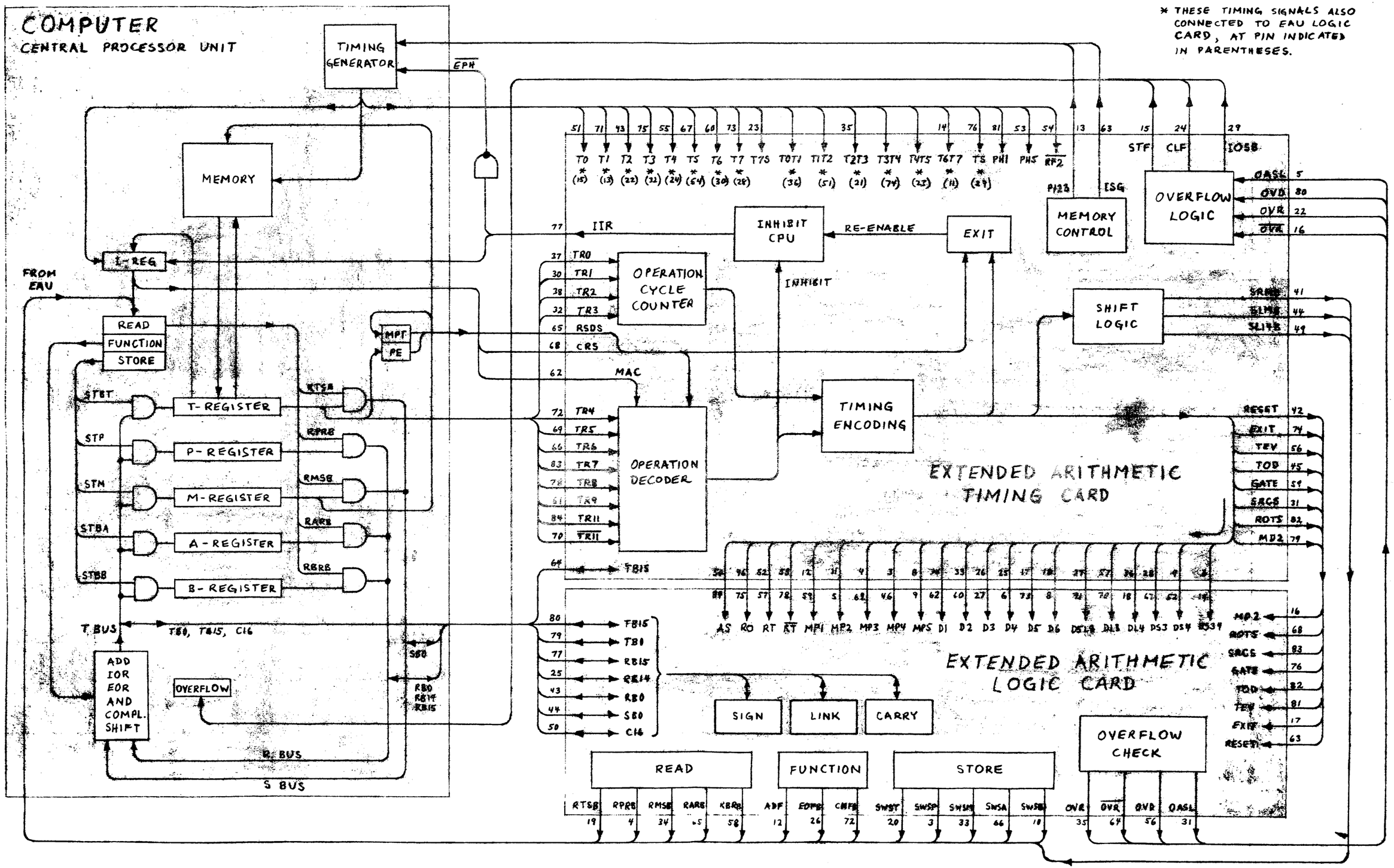


Figure 5-22. EAU Block Diagram

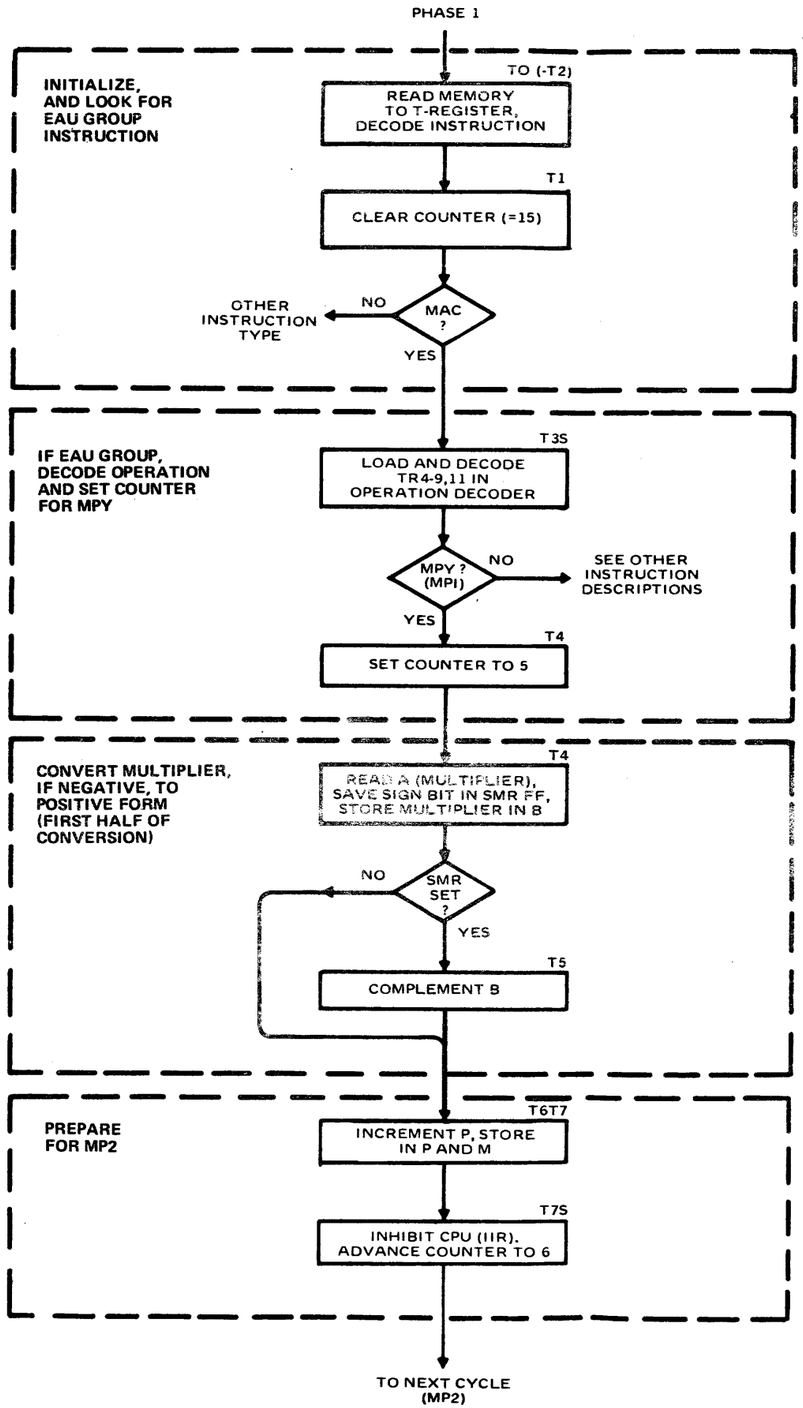


Figure 5-23. Multiply Cycle 1

Table 5-8. MP1 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T1 Clear Counter CTR(15) = PH1 · T1	T	U42C
T3S Decode MPY MPY = MAC · TR7 · $\overline{TR11}$ Clock = PH1 · T3 · TS	T T	U106, U114B U85C
T4 Set Counter CTR(5) = PH1 · T4 · MPY	T	U97C
T4 A to B RARB = MP1 · T4 EOFB = MP1 · T4 SWSB = MP1 · T4 SMR = MP1 · T4 · TB15	L L L L	U96B U46A U14D U52B
T5 Complement B RBRB = MP1 · T5 CMFB = MP1 · T5 SWSB = MP1 · T5 · SMR	L L L	U85C U106B U61A
T6T7 Increment P, M See CPU Instruction Decoder (RPRB, SBD, ADF, STP, STM)		
T7S Inhibit CPU IIR = MAC · T7 · TS	T	U77, U76D
T7S Advance Counter Clock = MPY · T7 · TS · ($\overline{TB15 \cdot M02}$)	T	U96B

* T = EAU Timing Card
L = EAU Logic Card

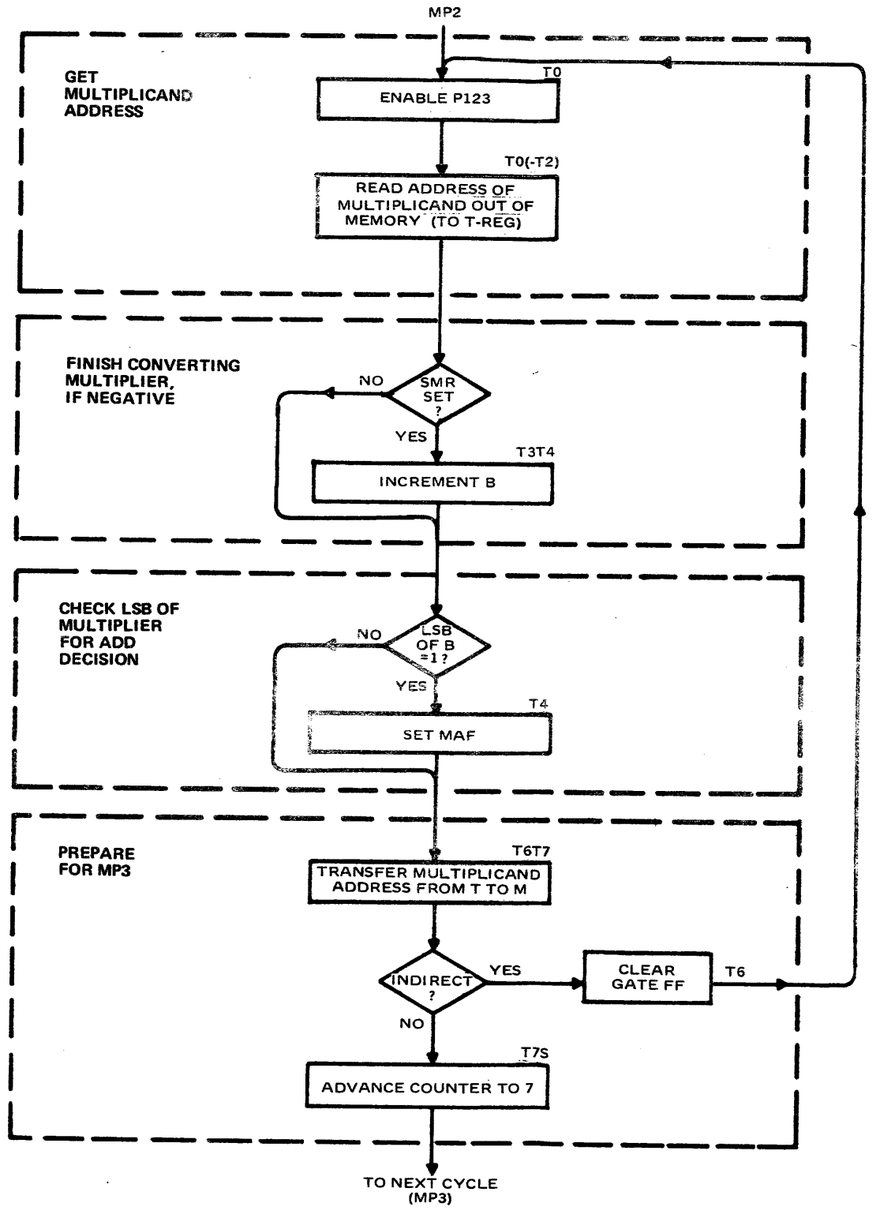


Figure 5-24. Multiply Cycle 2

Table 5-9. MP2 Logic Equations

EQUATION	CARD *	REFERENCE DESIGNATION
T0 Enable Memory P123 = MPY · CTR(6) · $\overline{T6T7}$	T	U15C
T3T4 Increment B RBRB = MP2 · T3T4 SBI = MD2 · T3T4 · GATE · SMR ADF = MD2 · T3T4 SWSB = MP2 · T4 · GATE · SMR	L L L L	U87D U51B U26D U51A
T4 Set Multiplication Add Function MAF = TB0 · MP2 · GATE · T4	L	U23A
T6T7 T to M RTSB = MD2 · T6T7 ADF = MD2 · T6T7 $\overline{SWSM}$ = MD2 · T7 $\overline{GATE}$ = TB15 · T6 T7S Advance Counter Clock = MPY · T7 · T5 · $\overline{(TB15 \cdot MD2)}$	L L L T T	U35C U25D U55C U55D U96B

*T = EAU Timing Card
L = EAU Logic Card

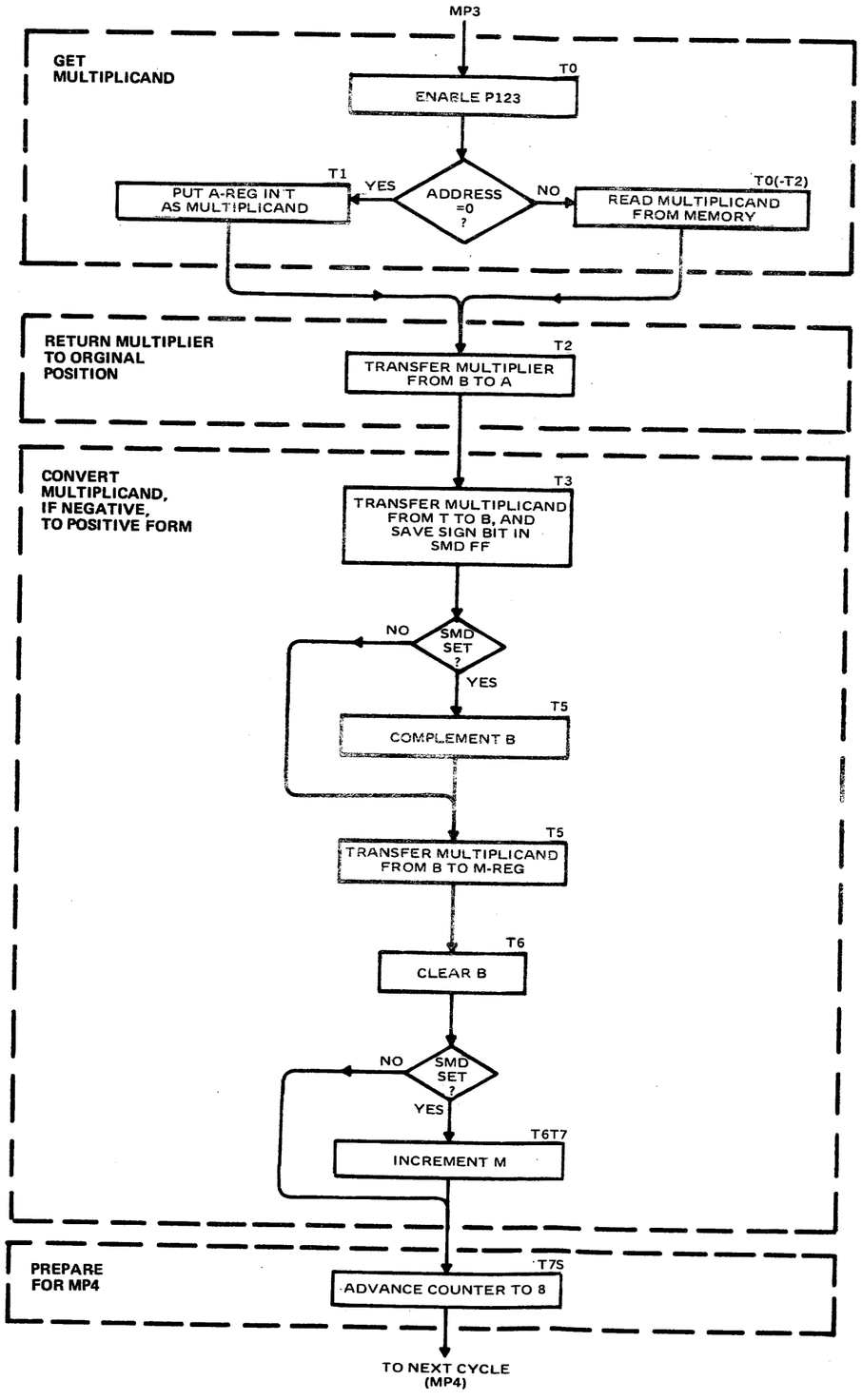


Figure 5-25. Multiply Cycle 3

Table 5-10. MP3 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T0 Enable Memory P123 = MPY · CTR(7) · $\overline{T6T7}$ T1 A to T See CPU Instruction Decoder (AAF, RARB, EOF, STBT)	T	U15C
T2 B to A RBRB = MP3 · T2 EOFB = MP3 · T2T3 SWSA = MP3 · T2	L L L	U84A U15B U107D
T3 T to B RTSB = MP3 · T3 EOFB = MP3 · T2T3 SWSB = MP3 · T3 SMD = MP3 · T3 · TS · TB15	L L L L	U35D U15B U14C U52D, U53A
T5 B or $\overline{B}$ to M RBRB = MP3 · T5 Either { EOFB = MP3 · T5 · $\overline{SMD}$ Either { CMFB = MP3 · T5 · SMD SWSM = MP3 · T5	L L L L	U85D U45C U106C U56D
T6 Clear B SWSB = MP3 · T6	L	U77D
T7 Increment M RMSB = MP3 · T6T7 RBO = MP3 · T6T7 ADF = MP3 · T7 SWSM = MP3 · T7 · SMD	L L L L	U64B U64A U15A U55B
T7S Advance Counter Clock = MPY · T7 · TS · ($\overline{TB15 \cdot MD2}$)	T	U96B

* T = EAU Timing Card
L = EAU Logic Card

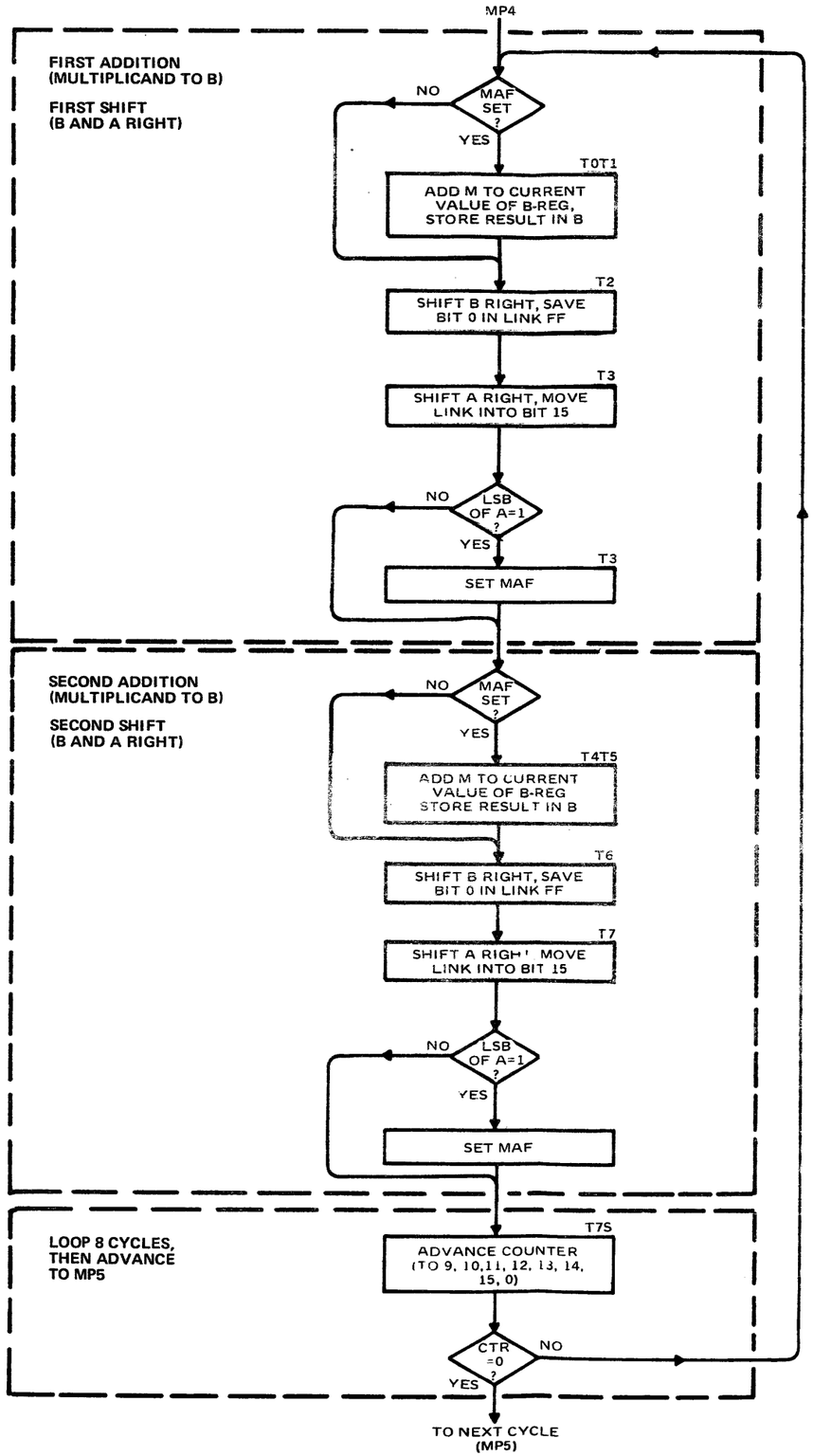


Figure 5-26. Multiply Cycle 4

Table 5-11. MP4 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T0T1 Add		
RMSB = MP4 · T0T1	L	U54A
RBRB = MP4 · T0T1	L	U74C
ADF = MP4 · T0T1	L	U14A
SWSB = MP4 · T1 · MAF	L	U76A
T2 Shift B		
RBRB = MP4 · T2	L	U87A
SRMB = MP4 · T2T3	T	U72A
SWSB = MP4 · T2	L	U77A
LINK = MP4 · TEV · TS · RB0	L	U123D, U124A
T3 Shift A		
RARB = MP4 · T3	L	U91D
SRMB = MP4 · T2T3	T	U72A
TB15 = MP4 · T3 · LINK	L	U126C
SWSA = MP4 · T3	L	U94B
MAF = MP4 · T3 · TB0	L	U33B
T4T5 Add		
RMSB = MP4 · T4T5	L	U54B
RBRB = MP4 · T4T5	L	U74C
ADF = MP4 · T4T5	L	U14A
SWSB = MP4 · T5 · MAF	L	U76A
T6 Shift B		
RBRB = MP4 · T6	L	U87B
SRMB = MP4 · T6T7	T	U72A
SWSB = MP4 · T6	L	U77B
LINK = MP4 · TEV · TS · RB0	L	U123D, U124A
T7 Shift A		
RARB = MP4 · T7	L	U91D
SRMB = MP4 · T6T7	T	U72A
TB15 = MP4 · T7 · LINK	L	U126C
SWSA = MP4 · T7	L	U94B
MAF = MP4 · T7 · TB0	L	U33B
T7S Advance Counter		
Clock = MPY · T7 · TS · $\overline{(TB15 \cdot MD2)}$	T	U96B

*T = EAU Timing Card
L = EAU Logic Card

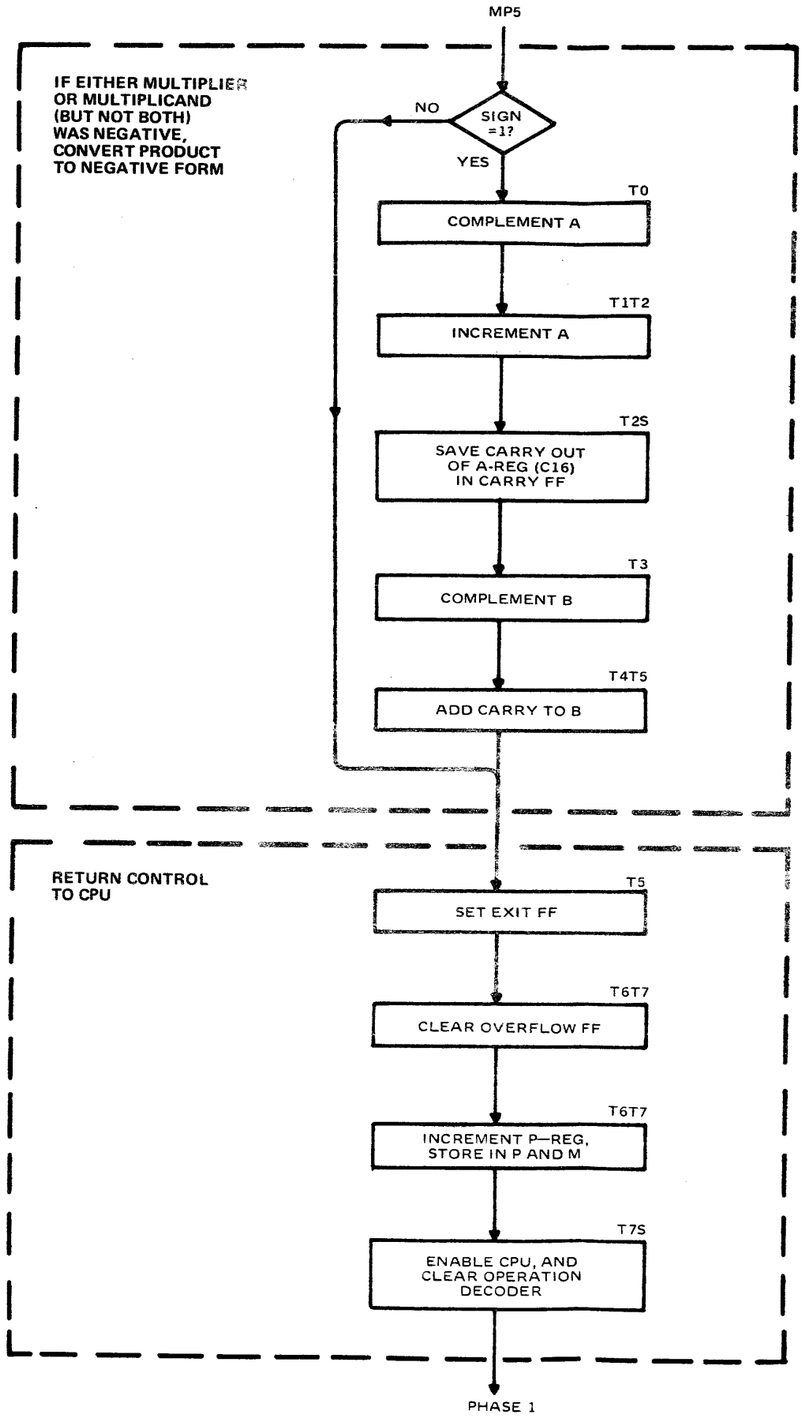


Figure 5-27. Multiply Cycle 5

Table 5-12. MP5 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T0 Complement A		
RARB = MP5 · T0	L	U95B
CMFB = MP5 · T0	L	U95A
SWSA = MP5 · T0 · SIGN	L	U94A
T1T2 Increment A		
RARB = MP5 · T1T2	L	U95C
SBO = MP5 · T1T2	L	U65A
ADF = MP5 · T1T2	L	U26A
SWSA = MP5 · T2 · SIGN	L	U94A
CARRY = MP5 · T2 · T5 · C16	L	U102D, U103D, U113B, U113A
T3 Complement B		
RBRB = MP5 · T3	L	U84C
CMFB = MP5 · T3	L	U104C
SWSB = MP5 · T3 · SIGN	L	U75B, U77C
T4T5 Carry into B		
RBRB = MP5 · T4T5	L	U74C, U24B
SBO = MP5 · T4T5 · CARRY	L	U65B, U33D
ADF = MP5 · T4T5	L	U14A, U24B
SWSB = MP5 · T5 · SIGN	L	U75A, U75B
T5 Set Exit		
Exit = MP5 · T5	T	U16B, U93A
T6T7 Clear Overflow		
CLF = MP5 · T6T7	T	U64B, U61D
IOSB = MP5 · T6T7	T	U62F, U61D
T6T7 Increment P, M		
RPRB = Exit · T6T7	L	U15C
SBO = Exit · T6T7	L	U33A
ADF = Exit · T6T7	L	U25C
SWSP = Exit · T7	L	U16C
SWSM = Exit · T7	L	U55D
T7S Enable CPU		
$\overline{IIR}$ = Exit · T7 · T5	T	U76B
$\overline{MPY}$ = Exit · T7	T	U107C

* T = EAU Timing Card
L = EAU Logic Card

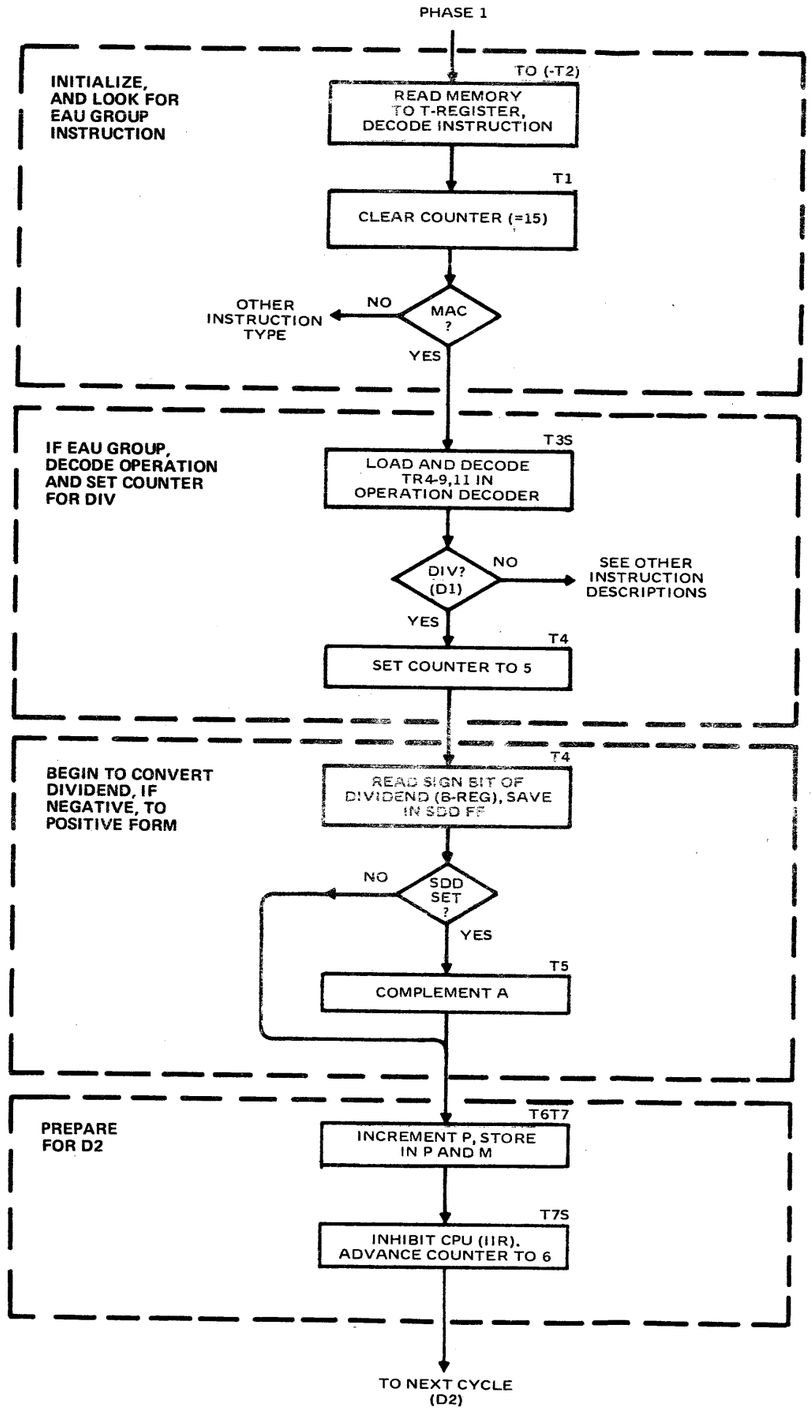


Figure 5-28. Divide Cycle 1

Table 5-13. D1 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T1 Clear Counter		
CTR(15) = PH1 · T1	T	U42C
T3S Decode DIV		
DIV = MAC · TRB · $\overline{TRTI}$	T	U106
Clock = PH1 · T3 · TS	T	U85C
T4 Set Counter		
CTR(5) = PH1 · T4 · DIV	T	U97C
T4 Save Dividend Sign		
RERB = D1 · T4	L	U85A
EOFB = D1 · T4 · TS	L	U44B
SDD = D1 · T4 · TB15	L	U52A
T5 Complement A		
RARB = D1 · T5	L	U96C
CMFB = D1 · T5	L	U106A
SWSA = D1 · T5 · SDD	L	U94C
T6T7 Increment P, M		
See CPU Instruction Decoder (RPRB, SBO, ADF, STP, STM)		
T7S Inhibit CPU		
IIR = MAC · T7 · TS	T	U77, U76D
T7S Advance Counter		
Clock = DIV · T7 · TS · ($\overline{TB15}$ · MD2)	T	U96B

* T = EAU Timing Card
L = EAU Logic Card

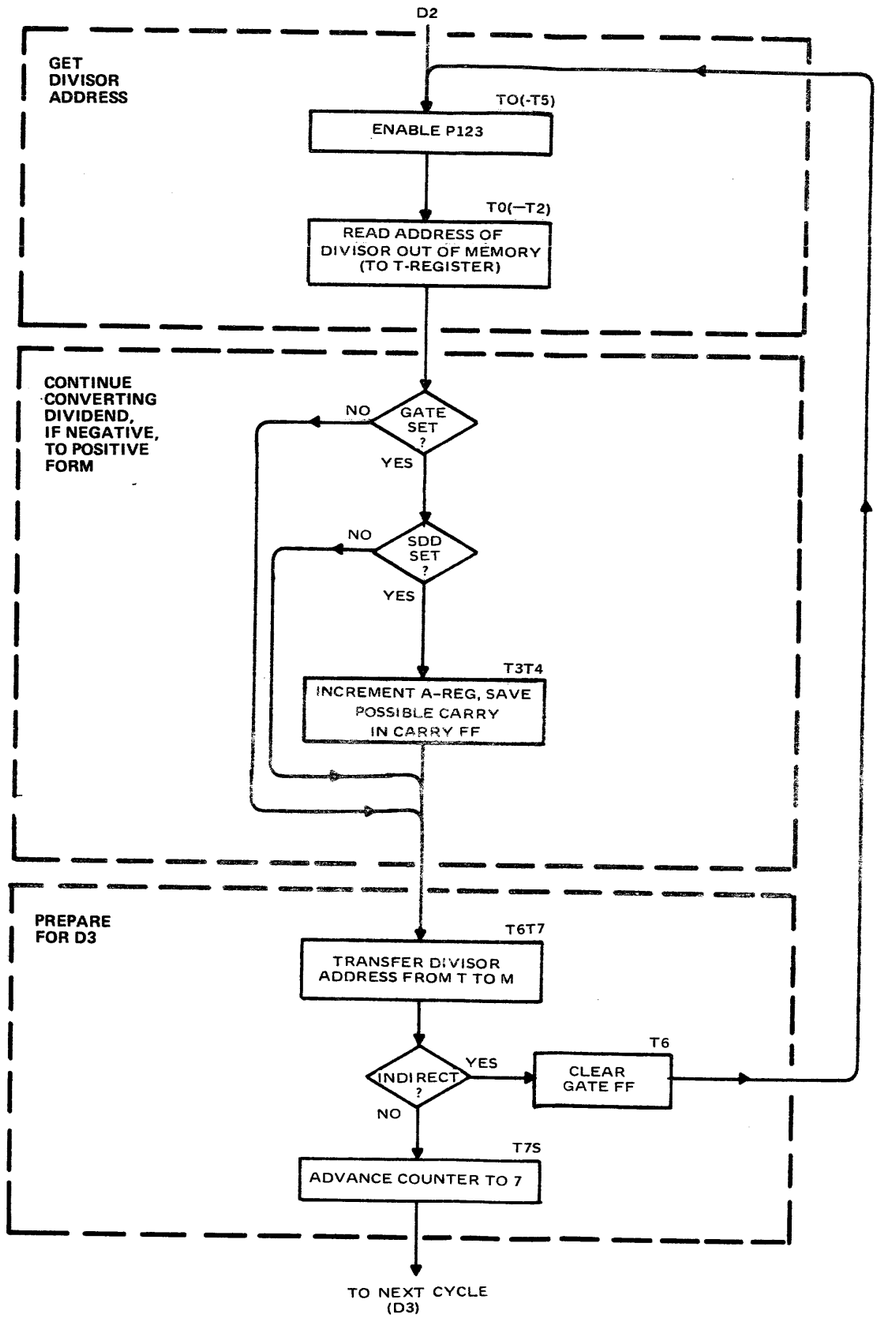


Figure 5-29. Divide Cycle 2

Table 5-14. D2 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T0 Enable Memory		
P123 = DIV · CTR(6) · $\overline{T6T7}$	T	U15C
T3T4 Increment A		
RARB = D2 · T3T4	L	U95D
SBO = MD2 · T3T4 · GATE · SDD	L	U51B
ADF = MD2 · T3T4	L	U26D
SWSA = D2 · T4 · GATE · SDD	L	U93A
T4 Save Carry		
Carry = D2 · T4 · GATE · C16	L	U122C, U113A, U102A, U103D
T6T7 T to M		
RTSB = MD2 · T6T7	L	U35C
ADF = MD2 · T6T7	L	U25D
SWSM = MD2 · T7	L	U55C
$\overline{GATE}$ = T815 · T6	T	U55D
T7S Advance Counter		
Clock = DIV · T7 · T5 · ($\overline{T815 \cdot MD2}$)	T	U96B

*T = EAU Timing Card
L = EAU Logic Card

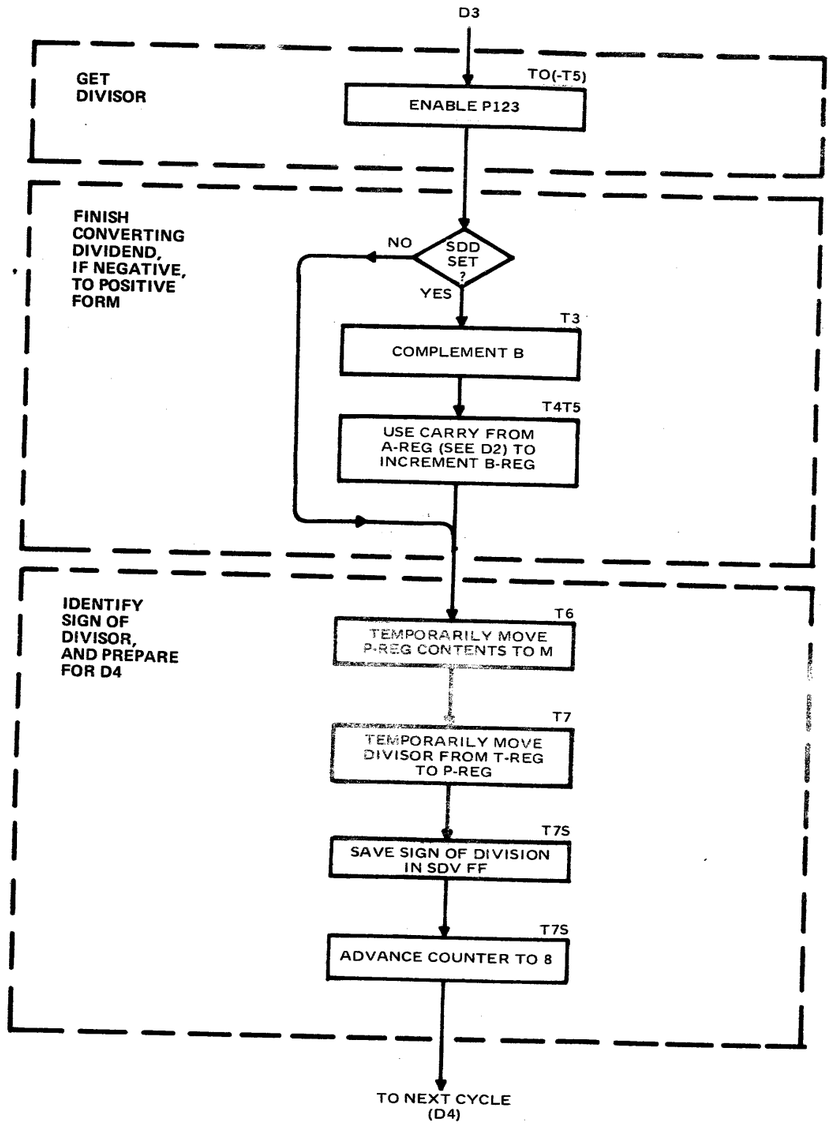


Figure 5-30. Divide Cycle 3

Table 5-15. D3 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T0 Enable Memory		
P123 = DIV · CTR(7) · $\overline{T6T7}$	T	U15C
T3 Complement B		
RBRB = D3 · T3	L	U84B
CMFB = D3 · T3	L	U106D
SWSB = D3 · T3 · SDD	L	U75C, U77C
T4T5 Increment B		
RBRB = D3 · T4T5	L	U74C, U14B
SBO = D3 · T4T5 · Carry	L	U33C, U65B
ADF = D3 · T4T5	L	U14A, U14B
SWSB = D3 · T5 · SDD	L	U75C, U75A
T6 P to M		
RPRB = D3 · T6	L	U15D
EOFB = D3 · T6T7	L	U45B
SWSM = D3 · T6	L	U56A
T7 T to P		
RTSB = D3 · T7	L	U35B
EOFB = D3 · T6T7	L	U45B
SWSP = D3 · T7	L	U16D
SDV = D3 · T7 · T5 · $\overline{T8I5}$	L	U52C, U53A
T7S Advance Counter		
Clock = DIV · T7 · T5 · ($\overline{T8I5 \cdot MD2}$)	T	U96B

*T = EAU Timing Card
L = EAU Logic Card

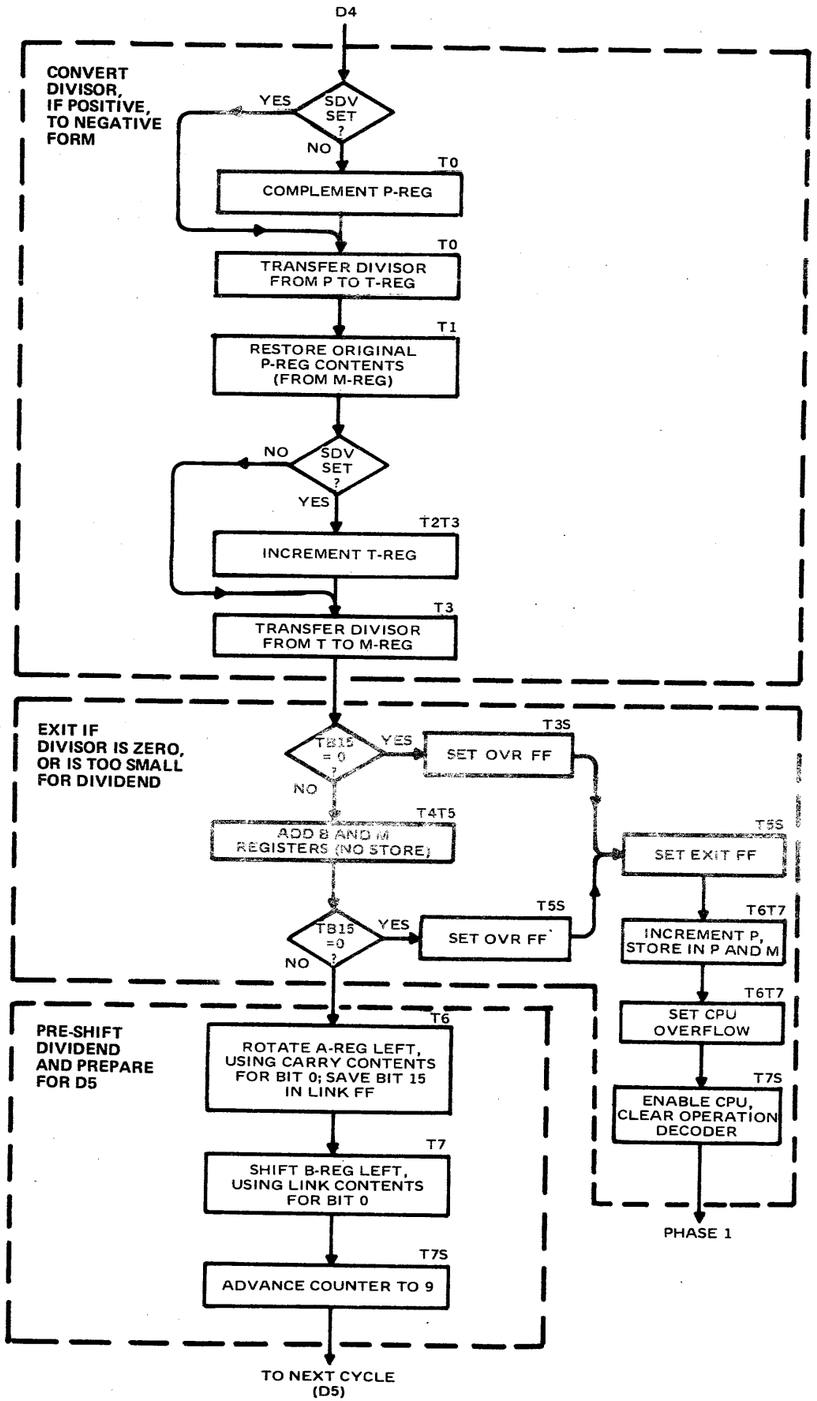


Figure 5-31. Divide Cycle 4

Table 5-16. D4 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T0 P or $\overline{P}$ to T		
RPRB = D4 · T0	L	U16B
Either { EOFB = D4 · T0 · SDV	L	U44C
Either { CMFB = D4 · T0 · $\overline{SDV}$	L	U61C
SWST = D4 · T0	L	U36B
T1 M to P		
RMSB = D4 · T1	L	U54D
EOFB = D4 · T1	L	U45A
SWSP = D4 · T1	L	U16A
T2T3 T or T+1 to M		
RTSB = D4 · T2T3	L	U36C
RB0 = D4 · T2T3 · $\overline{SDV}$	L	U66C
ADF = D4 · T2T3	L	U26B
SWSM = D4 · T3	L	U56B
T3S Zero Divisor Check		
OVR = D4 · T3 · T5 · $\overline{TB15}$	L	U83B, U83A, U82A
T4T5 Small Divisor Check		
RBRB = D4 · T4T5	L	U74C, U24A
RMSB = D4 · T4T5	L	U54C
ADF = D4 · T4T5	L	U14A, U24A
OVR = D4 · T5 · T5 · $\overline{TB15}$	L	U83C, U83A, U82A
T5 Exit		
Exit = D4 · T5 · OVR	T	U61A, U14D
(See D6 T6T7 for Exit sequence)		
T6T7 Shift A, B		
SLMB = D4 · T6T7 · $\overline{EXIT}$	T	U64F, U66C
SL14B = D4 · T6T7 · $\overline{EXIT}$	T	U65B, U66C
RARB = D4 · T6 · $\overline{OVD}$	L	U91D, U105A/D, U104B
TB0 = D4 · T6 · $\overline{OVD}$ · Carry	L	U115C, U105D, U104B
SWSA = D4 · T6 · $\overline{OVD}$	L	U94B, U105A/D, U104B
LINK = D4 · TEV · T5 · RB15	L	U123A, U124A
RBRB = D4 · T7 · $\overline{OVD}$	L	U74B, U104A, U104B
TB0 = D4 · T7 · $\overline{OVD}$ · LINK	L	U116A, U104B
SWSB = D4 · T7 · $\overline{OVD}$	L	U74D, U104A, U104B
T7S Advance Counter		
Clock = DIV · T7 · T5 · ($\overline{TB15}$ · MD2)	T	U96B

* T = EAU Timing Card
L = EAU Logic Card

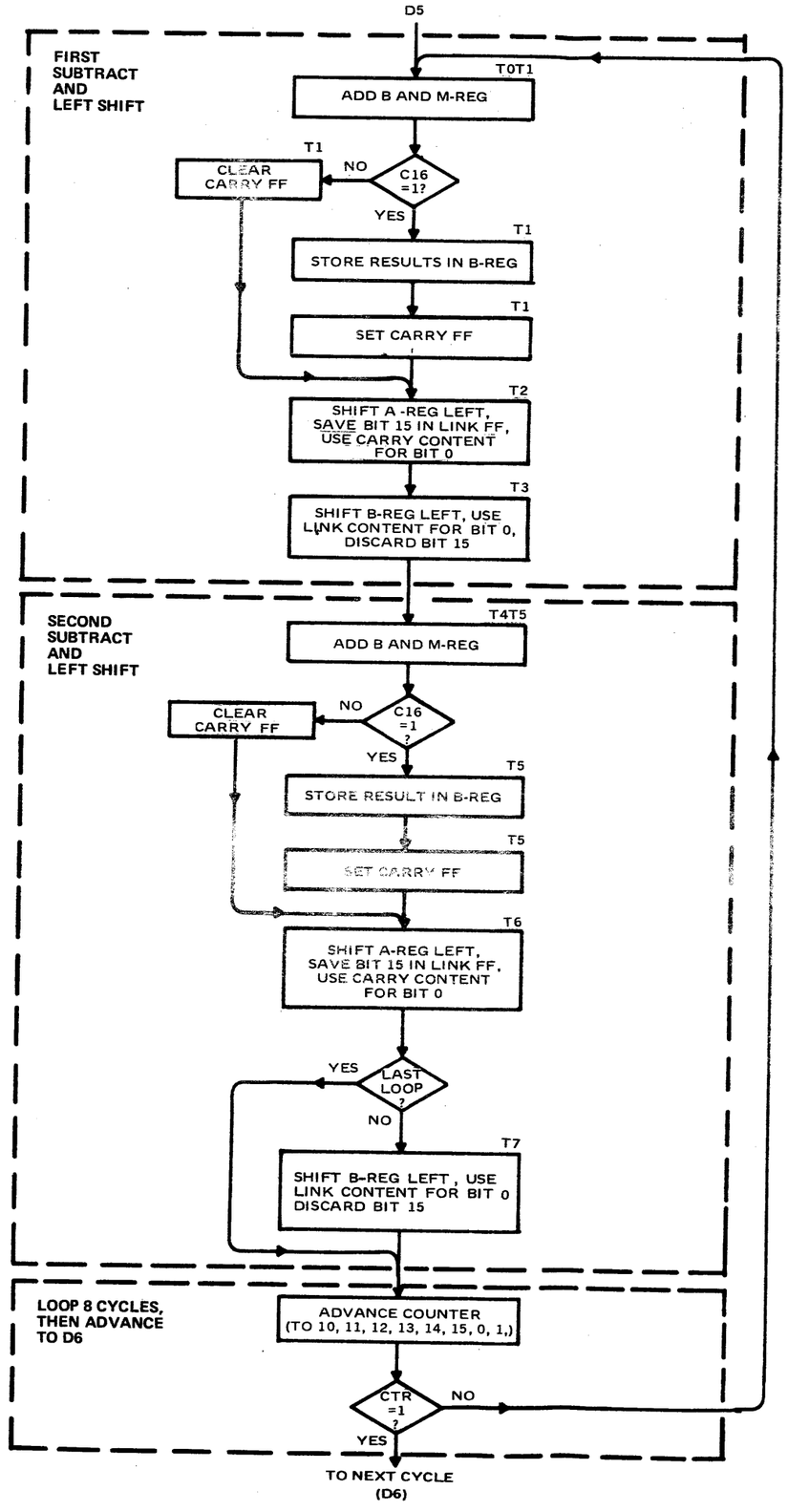


Figure 5-32. Divide Cycle 5

Table 5-17. D5 Logic Equations

EQUATION	CARD *	REFERENCE DESIGNATION
T0T1 Add B and M		
RBRB = D5 · T0T1	L	U74C, U24D, U34B
RMSB = D5 · T0T1	L	U54A, U34B
ADF = D5 · T0T1	L	U14A, U24D, U34B
SWSB = D5 · T1 · C16	L	U76C
Carry = D5 · T1 · C16 · TS	L	U113D, U113A, U103D, U102B
T2T3 Shift A and B		
SLMB = D5 · T2T3	T	U64F, U66B, U62A
SLI4B = D5 · T2T3	T	U65B, U66B, U62A
RARB = D5 · T2	L	U91D, U105A, U105B
TB0 = D5 · T2 · Carry	L	U115C, U105B
SWSA = D5 · T2	L	U94B, U105A, U105B
LINK = D5 · TEV · TS · RB15	L	U123B, U124A
RBRB = D5 · T3	L	U84D
TB0 = D5 · T3 · LINK	L	U116C
SWSB = D5 · T3	L	U14C, U34D
T4T5 Add B and M		
RBRB = D5 · T4T5	L	U74C, U24C, U34B
RMSB = D5 · T4T5	L	U54B, U34B
ADF = D5 · T4T5	L	U14A, U24C, U34B
SWSB = D5 · T5 · C16	L	U76C
Carry = D5 · T5 · C16 · TS	L	U113B/A, U103D, U102B
T6 Shift A		
SLMB = D5 · T6T7	T	U64F, U66B, U62B
SLI4B = D5 · T6T7	T	U65B, U66B, U62B
RARB = D5 · T6	L	U91D, U105A, U105C
TB0 = D5 · T6 · Carry	L	U115C, U105C
SWSA = D5 · T6	L	U94B, U105A, U105C
LINK = D5 · TEV · TS · RB15	L	U123B, U124A
T7 Shift B		
SLMB = D5 · T6T7	T	U64F, U66B, U62B
SLI4B = D5 · T6T7	T	U65B, U66B, U62B
RBRB = D5L8	L	U74B, U91B
TB0 = D5L8 · LINK	L	U115B
SWSB = D5L8	L	U74D, U91B
(D5L8 = D5 · $\overline{CTR(0)}$ · T7)	T	U26C
T7S Advance Counter		
Clock = DIV · T7 · TS · ($\overline{TB15 \cdot MD2}$)	T	U96B

* T = EAU Timing Card
L = EAU Logic Card

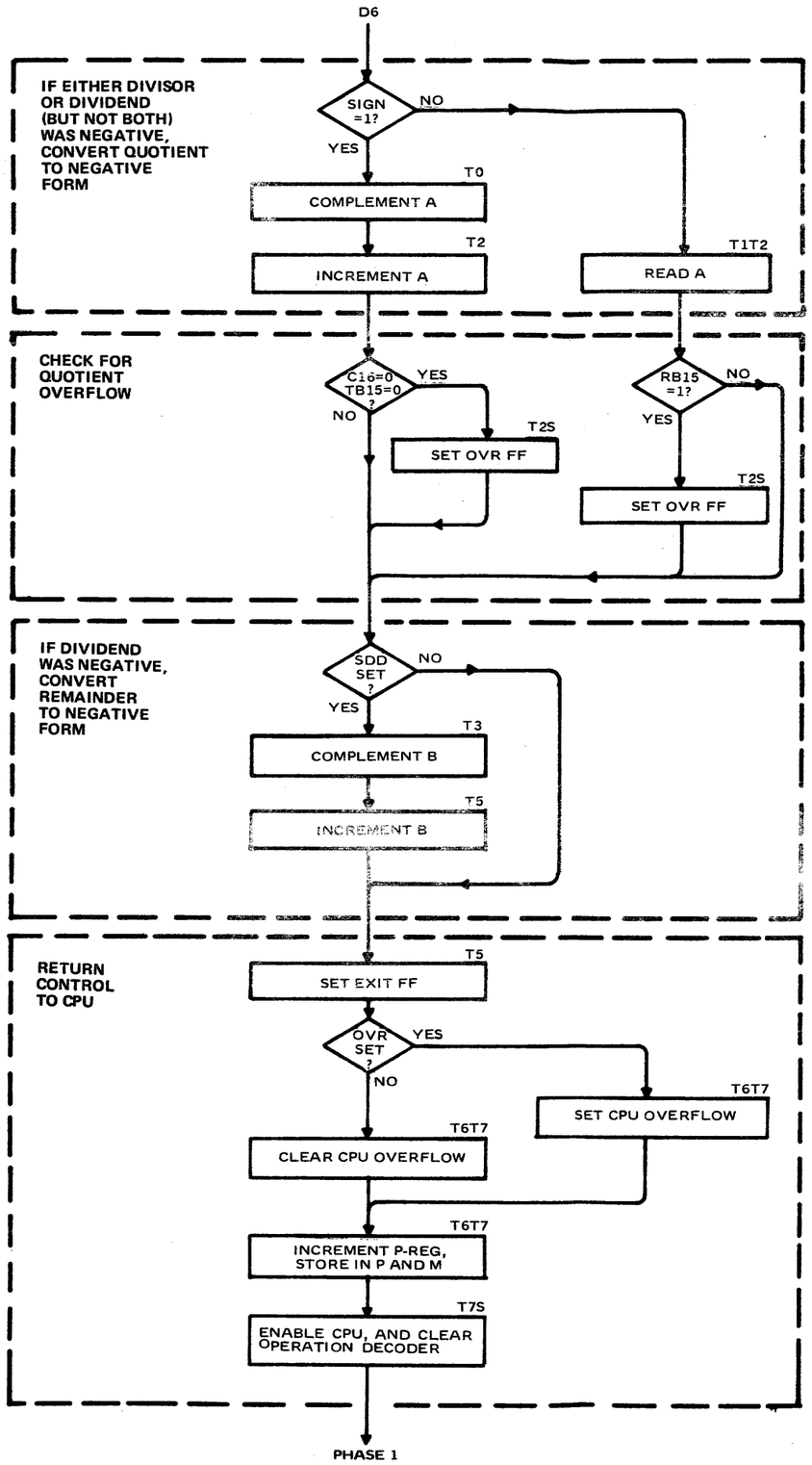


Figure 5-33. Divide Cycle 6

Table 5-18. D6 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T0 Complement A		
RARB = D6 · T0	L	U95B, U34A
CMFB = D6 · T0	L	U95A, U34A
SWSA = D6 · T0 · SIGN	L	U94A, U34A
T1T2 Increment A		
RARB = D6 · T1T2	L	U95C, U34A
SB0 = D6 · T1T2 · SIGN	L	U65C
ADF = D6 · T1T2	L	U26C
SWSA = D6 · T2 · SIGN	L	U94A, U34A
T2S Positive-Quotient Overflow		
OVR = D6 · T2 · T5 · $\overline{SIGN}$ · RB15	L	U92C, U83A, U83D
T2S Negative-Quotient Overflow		
OVR = D6 · T2 · T5 · SIGN · $\overline{C16}$ · $\overline{TB15}$	L	U93B, U83A, U83D
T3 Complement B		
RBRB = D6 · T3	L	U84C, U34A
CMFB = D6 · T3	L	U104C, U34A
SWSB = D6 · T3 · SDD	L	U77C, U75D
T4T5 Increment B		
RBRB = D6 · T4T5	L	U74C, U24B, U34A
SB0 = D6 · T4T5	L	U65B, U23B
ADF = D6 · T4T5	L	U14A, U24B, U34A
SWSB = D6 · T5 · SDD	L	U75C, U75D
T5 Set Exit		
Exit = D6 · T5	T	U21A, U14D
T6T7 Clear or set Overflow		
Either CLF = D6 · T6T7 · $\overline{OVR}$	T	U64B, U61C, U61B
Either IOSB = D6 · T6T7 · $\overline{OVR}$	T	U62F, U61C, U61B
Either STF = D6 · T6T7 · OVR	T	U62D, U72B, U61B
Either IOSB = D6 · T6T7 · OVR	T	U62E, U72B, U61B
T6T7 Increment P, M		
RPRB = Exit · T6T7	L	U15C
SB0 = Exit · T6T7	L	U33A
ADF = Exit · T6T7	L	U25C
SWSP = Exit · T7	L	U16C
SWSM = Exit · T7	L	U55D
T7S Enable CPU		
$\overline{IIR}$ = Exit · T7 · T5	T	U76E
$\overline{DIV}$ = Exit · T7	T	U107C

* L = MA Timing Card
T = MA Timing Card

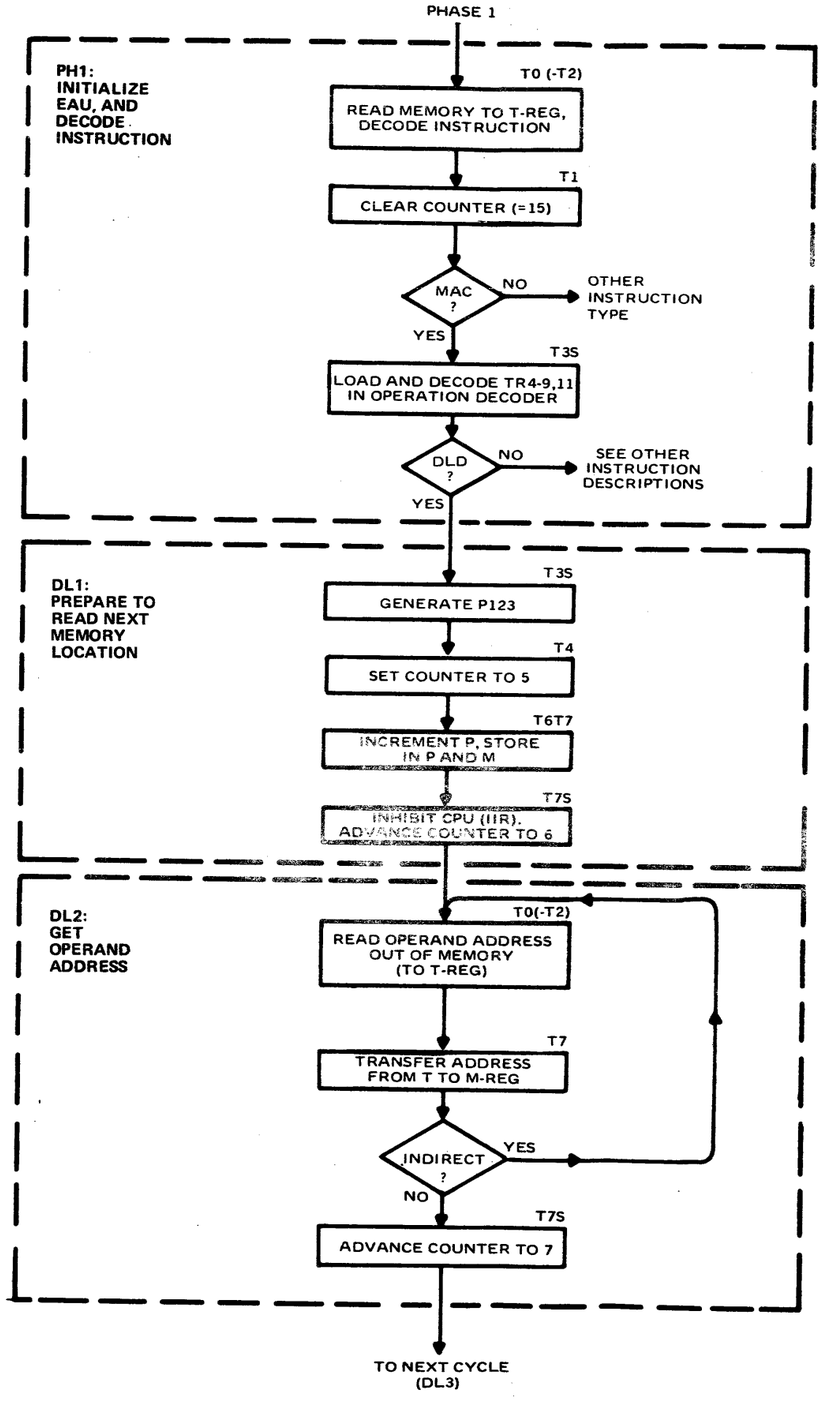


Figure 5-34. Double Load Cycles 1 and 2

Table 5-19. DL1/2 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T1 Clear Counter		
$CTR(15) = PH1 \cdot T1$	T	U42C
T3S Decode DLD		
$DLD = MAC \cdot TR11 \cdot \overline{TR9} \cdot \overline{TR8}$	T	U106
$Clock = PH1 \cdot T3 \cdot TS$	T	U85C
T3S Generate P123		
$P123 = DLD$	T	U74F, U93D
T4 Set Counter		
$CTR(5) = PH1 \cdot T4 \cdot DLD$	T	U97C
T7S Inhibit CPU		
$IIR = MAC \cdot T7 \cdot TS$	T	U77, U76D
T7S Advance Counter		
$Clock = DLD \cdot T7 \cdot TS \cdot (\overline{TB15} \cdot MD2)$	T	U96B
T7 T to M		
$RTSB = MD2 \cdot T6T7$	L	U35C
$ADF = MD2 \cdot T6T7$	L	U25D
$SWSM = MD2 \cdot T7$	L	U55C
$(MD2 = DLD \cdot CTR(6))$	T	U27B
T7S Advance Counter		
$Clock = DLD \cdot T7 \cdot TS \cdot (\overline{TB15} \cdot MD2)$	T	U96B

*T = EAU Timing Card
L = EAU Logic Card

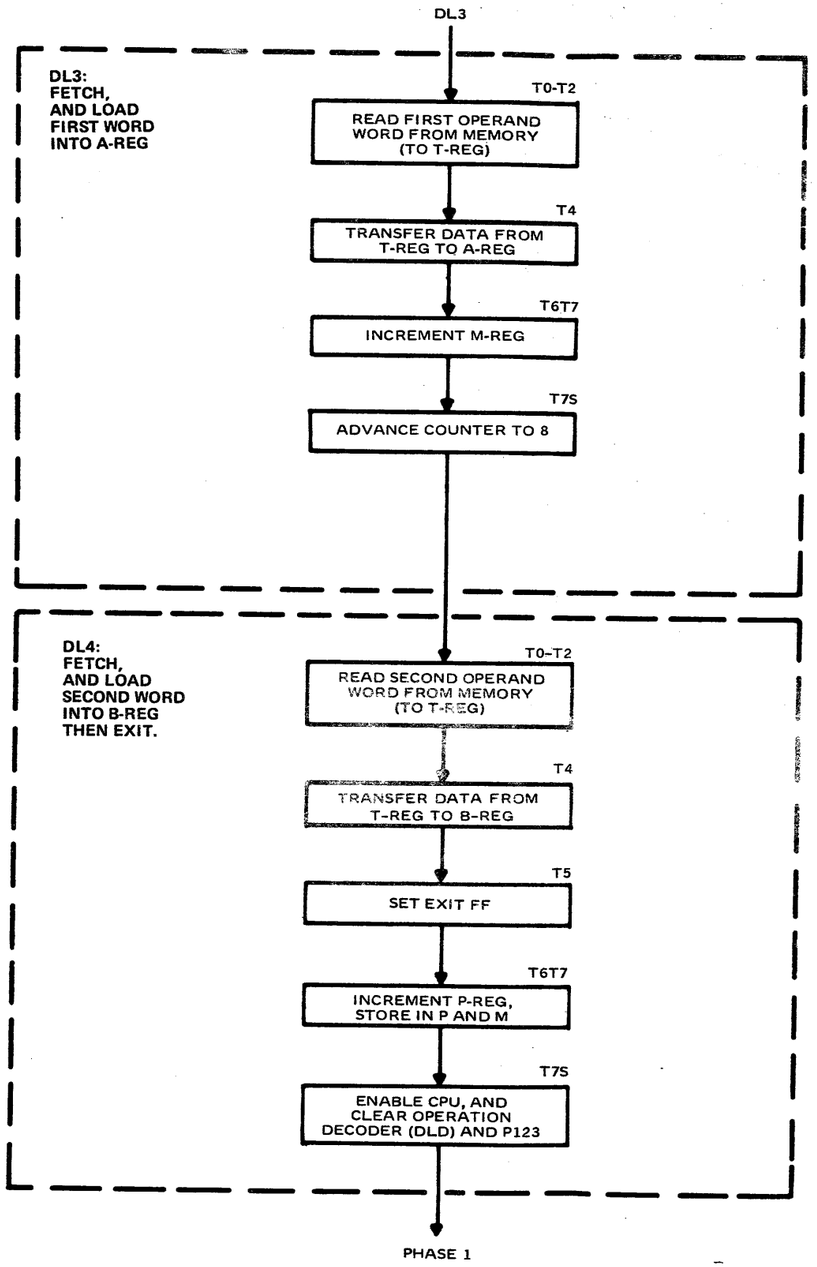


Figure 5-35. Double Load Cycles 3 and 4

Table 5-20. DL3/4 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T4 T to A		
RTSB = DL3 · T4	L	U35A
EOFB = DL3 · T4	L	U46B
SWSA = DL3 · T4	L	U107C
T6T7 Increment M		
RMSB = DL3 · T6T7	L	U64D
RBØ = DL3 · T6T7	L	U66B
ADF = DL3 · T6T7	L	U25B
SWSM = DL3 · T7	L	U55A
T7S Advance Counter		
Clock = DLD·T7·TS·($\overline{TB15 \cdot MD2}$)	T	U96B
T4 T to B		
RTSB = DL4 · T4	L	U36D
EOFB = DL4 · T4	L	U46C
SWSB = DL4 · T4	L	U44A
T5 Set Exit		
Exit = DL4 · T5	T	U13F, U14D
T6T7 Increment P, M		
RPRE = Exit · T6T7	L	U15C
SBØ = Exit · T6T7	L	U33A
ADF = Exit · T6T7	L	U25C
SWSP = Exit · T7	L	U16C
SWSM = Exit · T7	L	U55D
T7S Enable CPU		
$\overline{IIR}$ = Exit · T7 · TS	T	U76B
$\overline{DLD}$ = Exit · T7	T	U107C
$\overline{P123}$ = $\overline{DLD}$	T	U74F, U93D

*T = EAU Timing Card
L = EAU Logic Card

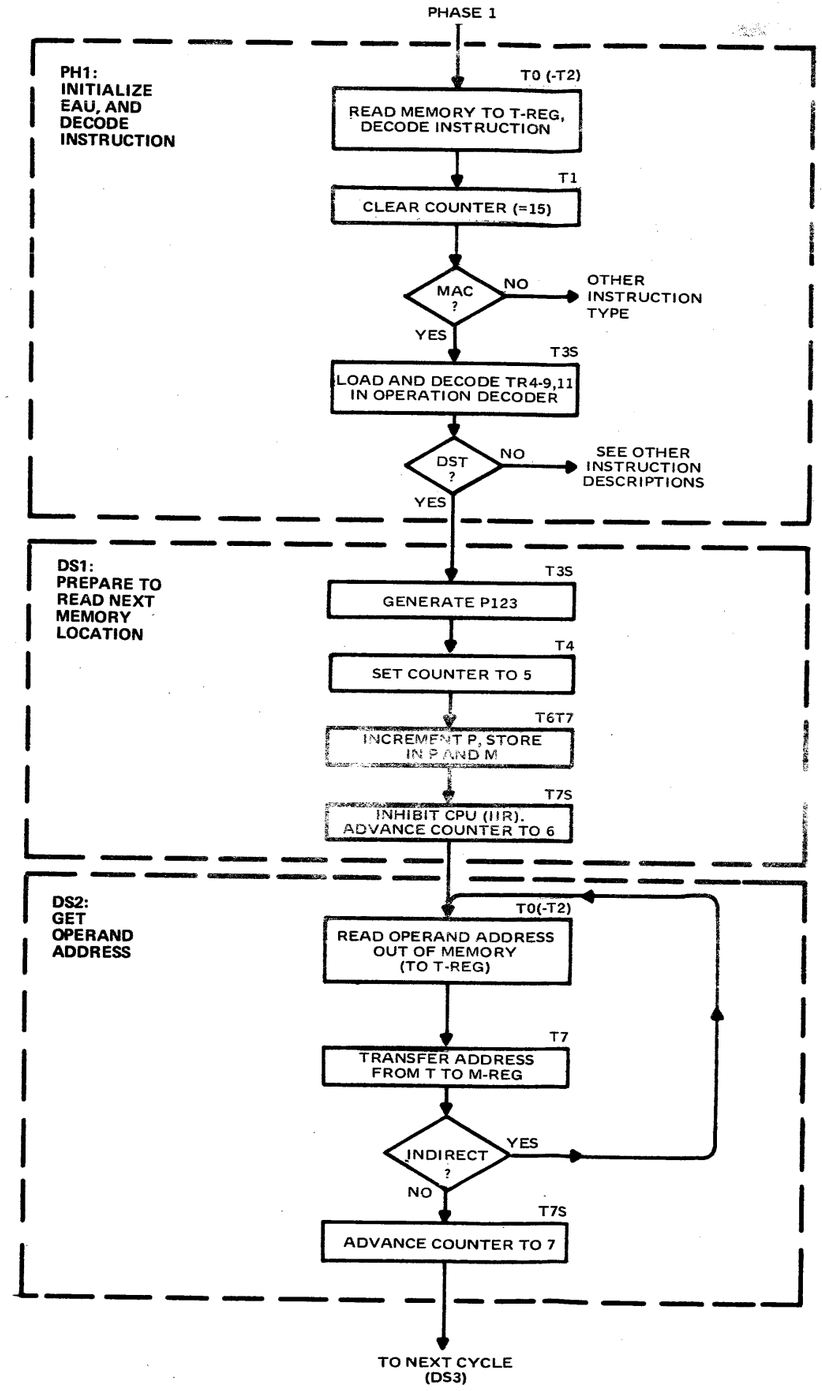


Figure 5-36. Double Store Cycles 1 and 2

Table 5-21. DS1/2 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T1 Clear Counter		
$CTR(15) = PH1 \cdot T1$	T	U42C
T3S Decode DST		
$DST = MAC \cdot TR11 \cdot TR8$	T	U106
$Clock = PH1 \cdot T3 \cdot TS$	T	U85C
T3S Generate P123		
$P123 = DST$	T	U74F, U93B
T4 Set Counter		
$CTR(5) = PH1 \cdot T4 \cdot DST$	T	U97C
T7S Inhibit CPU		
$IIR = MAC \cdot T7 \cdot TS$	T	U77, U76D
T7S Advance Counter		
$Clock = DST \cdot T7 \cdot TS \cdot (\overline{TB15 \cdot MD2})$	T	U96B
T7 T to M		
$RTSB = MD2 \cdot T6T7$	L	U35C
$ADF = MD2 \cdot T6T7$	L	U25D
$SWSM = MD2 \cdot T7$	L	U55C
$(MD2 = DST \cdot CTR(6))$	T	U27B
T7S Advance Counter		
$Clock = DST \cdot T7 \cdot TS \cdot (\overline{TB15 \cdot MD2})$	T	U96B

* T = EAU Timing Card
L = EAU Logic Card

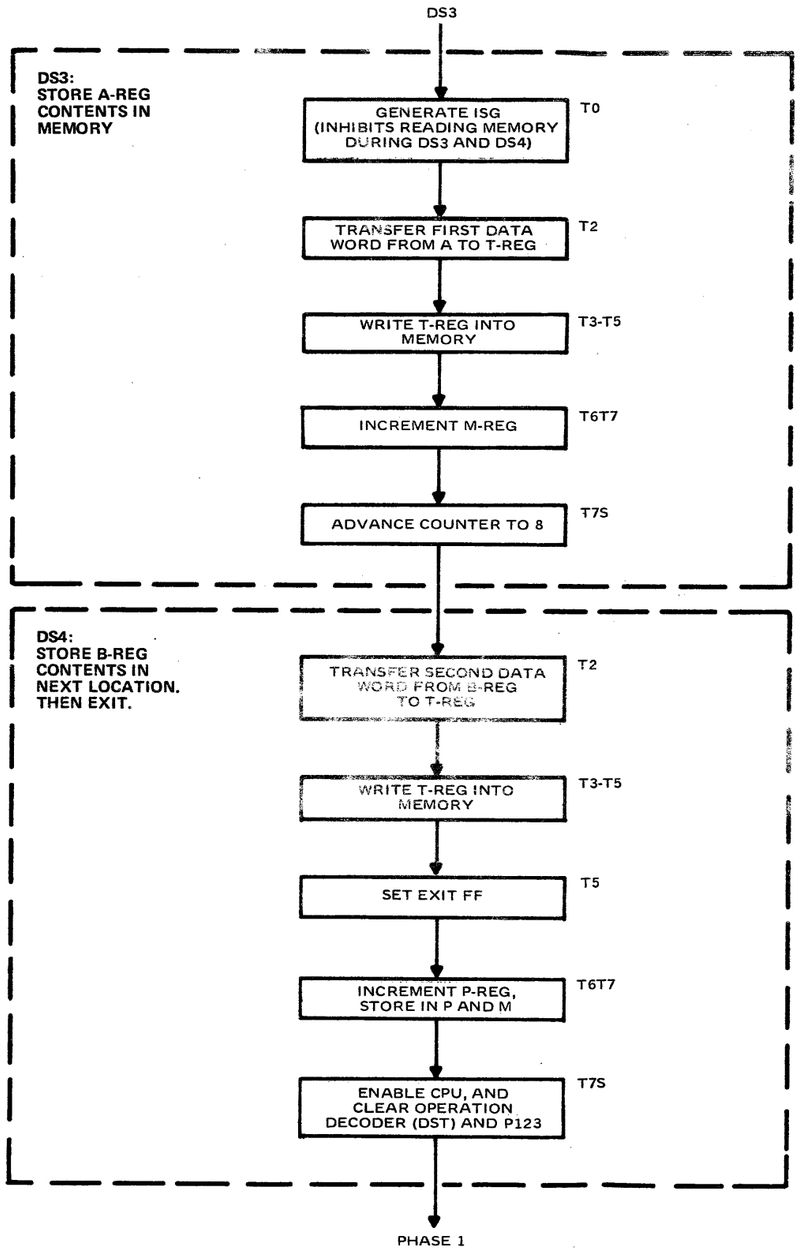


Figure 5-37. Double Store Cycles 3 and 4

Table 5-22. DS3/4 Logic Equations

EQUATION	CARD*	REFERENCE DESIGNATION
T0 Generate ISG		
$ISG = \overline{EPHX} \cdot DS34$	T	U13C
$(DS34 = DS3 + DS4)$	T	U13D, U13E
T2 A to T		
$RARB = DS3 \cdot T2$	L	U96D
$EOFB = DS34 \cdot T2$	L	U46D
$SWST = DS34 \cdot T2$	L	U36A
T6T7 Increment M		
$RMSB = DS3 \cdot T6T7$	L	U64C
$RBI = DS3 \cdot T6T7$	L	U66A
$ADF = DS3 \cdot T6T7$	L	U25A
$SWSM = DS3 \cdot T7$	L	U56C
T7S Advance Counter		
$Clock = DST \cdot T7 \cdot TS \cdot (\overline{TB15 \cdot MD2})$	T	U96B
T2 B to T		
$RBRB = DS4 \cdot T2$	L	U87C
$EOFB = DS34 \cdot T2$	L	U46D
$SWST = DS34 \cdot T2$	L	U36A
T5 Set Exit		
$Exit = DS4 \cdot T5$	T	U13B, U14D
T6T7 Increment P, M		
$RPRB = Exit \cdot T6T7$	L	U15C
$SBI = Exit \cdot T6T7$	L	U33A
$ADF = Exit \cdot T6T7$	L	U25C
$SWSP = Exit \cdot T7$	L	U16C
$SWSM = Exit \cdot T7$	L	U55D
T7S Enable CPU		
$\overline{IIR} = Exit \cdot T7 \cdot TS$	T	U76B
$\overline{DST} = Exit \cdot T7$	T	U107C
$\overline{P123} = \overline{DST}$	T	U74F, U93B

* T = EAU Timing Card
L = EAU Logic Card

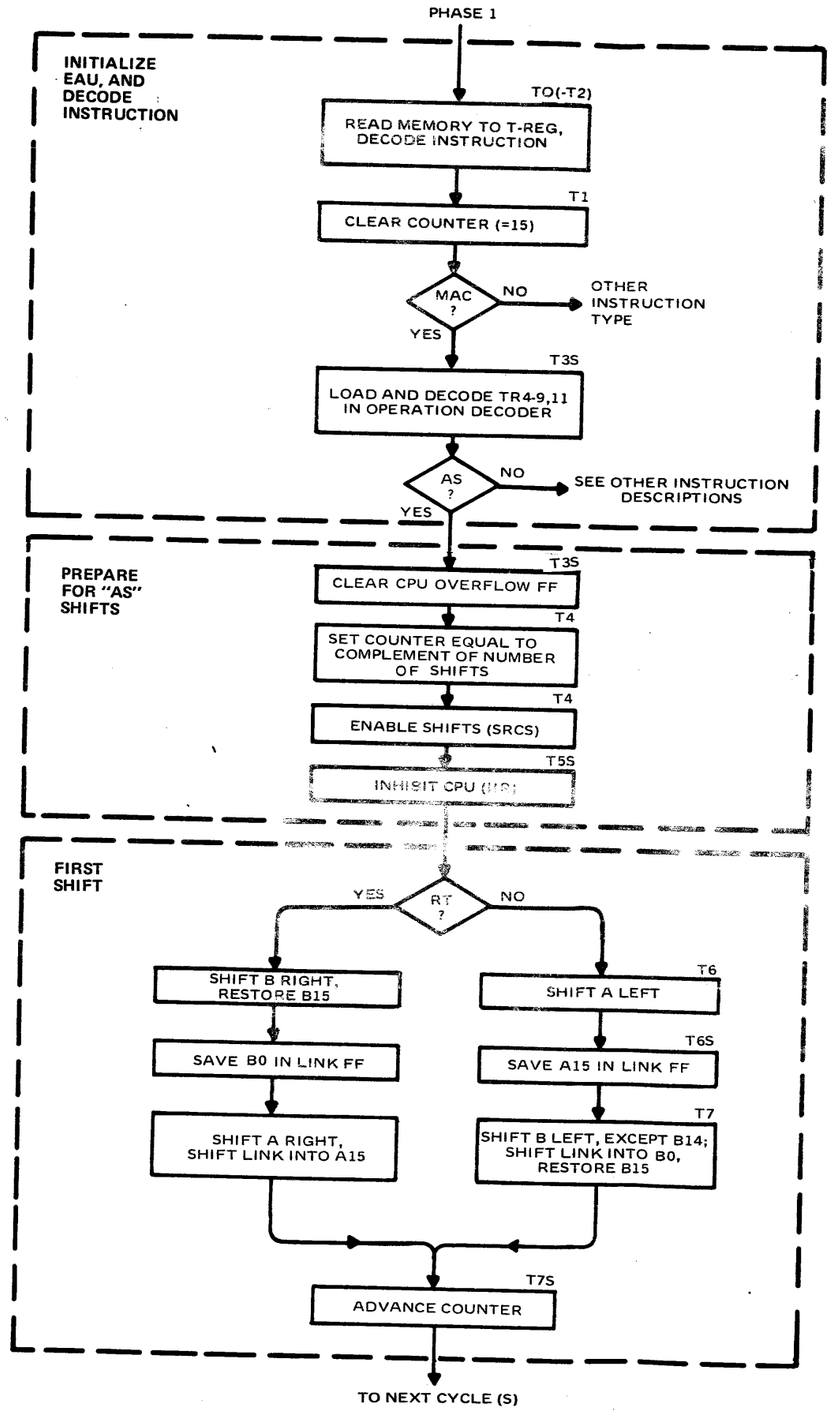


Figure 5-38. Arithmetic Shifts, Phase 1

Table 5-23. Arithmetic Shift Equations (1)

EQUATION	CARD*	REFERENCE DESIGNATION
T1 Clear Counter		
CTR(15) = PH1 · T1	T	U42C
T3S Decode AS		
AS = MAC · TR4	T	U105
Clock = PH1 · T3 · TS	T	U85C
T3S Clear Overflow		
CLF = AS · EPHX	T	U64B, U67A
IOSB = AS · EP4X	T	U62F, U67A
T4 Set Counter		
CTR(#) = AS · PH1 · T4	T	U97A
T4 Enable Shifts		
SRCS = AS · CTO0	T	U86A
T5S Inhibit CPU		
IIR = AS · T5 · TS	T	U76C, U77
RIGHT SHIFT EQUATIONS (see next page for left shift equations)		
TEV Shift B		
RBRB = SRCS · RT · TEV	L	U74A, U125B, U125A
SRMB = SRCS · RT	T	U86B
TBIST = SRCS · RT · TEV · AS · RBI5	L	U126A, U125B, U125A
SWSB = SRCS · RT · TEV	L	U76B, U125B, U125A
LINK = SRCS · RT · TEV · TS · RB0	L	U124A, U123C, U103B
TOD Shift A		
RARB = SRCS · RT · TOD	L	U91D, U122A
SRMB = SRCS · RT	T	U86B
TBIST = SRCS · RT · TOD · LINK	L	U127B
SWSA = SRCS · RT · TOD	L	U94B, U122A
T7S Advance Counter		
Clock = AS · TOD · TS · CTO0	T	U96A

* T = EAU Timing Card
L = EAU Logic Card

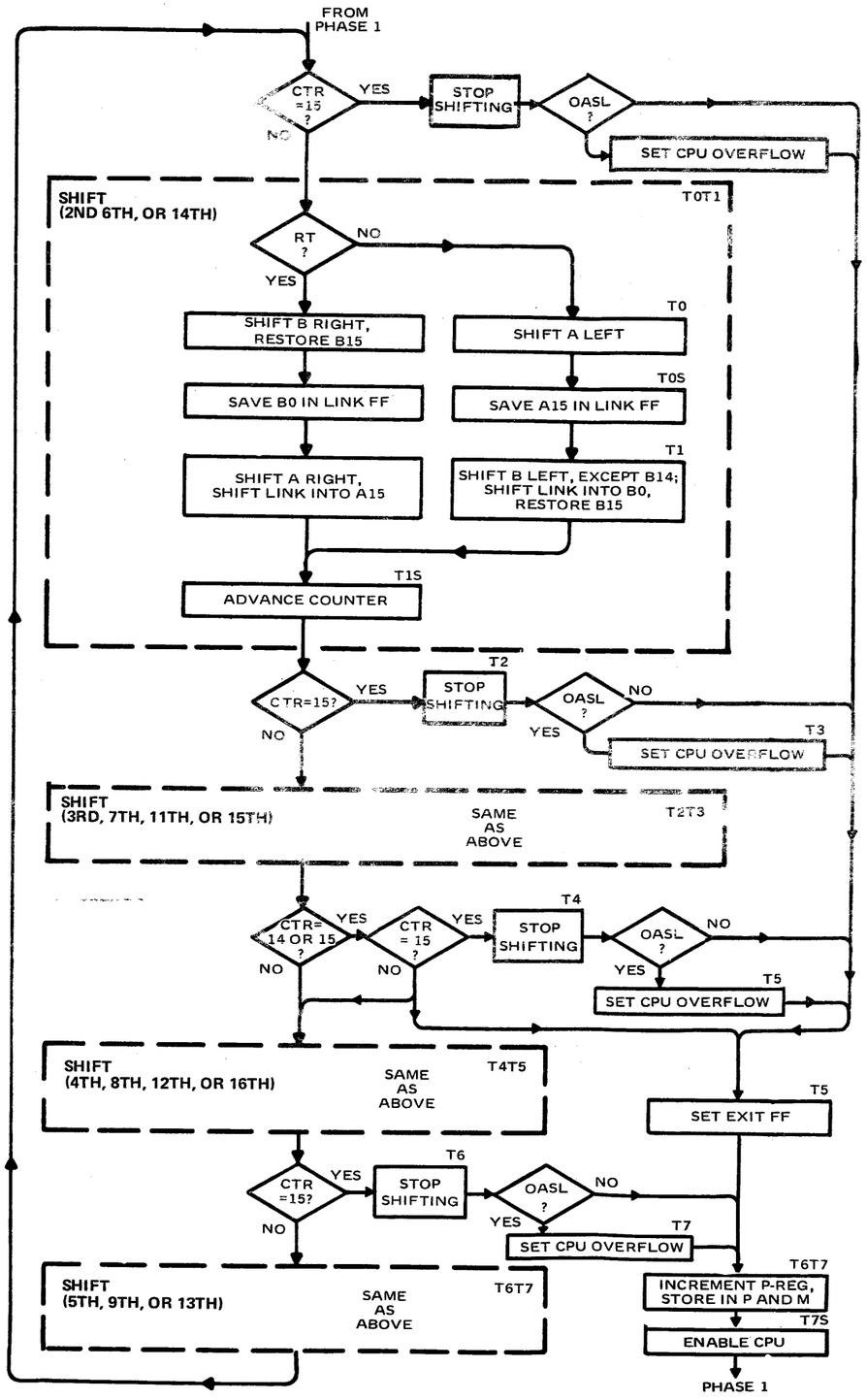


Figure 5-39. Arithmetic Shift Loops

Table 5-24. Arithmetic Shift Equations (2)

EQUATION	CARD*	REFERENCE DESIGNATION
LEFT SHIFT EQUATIONS (see preceding page for right shift equations)		
TEV Shift A		
RARB = SRCS · TEV · $\overline{RT}$	L	U91D, U114C
SLMB = SRCS · $\overline{RT}$	T	U86C
SL14B = AS · TEV · $\overline{RT}$	T	U65A
SWSA = SRCS · TEV · $\overline{RT}$	L	U94B, U114C
LINK = SRCS · TEV · T5 · $\overline{RT}$ · RB15	L	U124A, U123C, U103A
TOD Shift B		
RBRB = SRCS · TOD · $\overline{RT}$	L	U74A, U125C
SLMB = SRCS · $\overline{RT}$	T	U86C
TB0 = SRCS · TOD · $\overline{RT}$ · LINK	L	U115D, U114B, U115A
TB15 = SRCS · TOD · $\overline{RT}$ · AS · RB15	L	U126A, U125C
OASL = RB14 ⊕ RB15	L	U82D, U82C
SWSB = SRCS · TOD · $\overline{RT}$	L	U76B, U125C
Stop Shift		
$\overline{SRCS}$ = AS · $\overline{CTOO}$	T	U86A
Set Overflow		
STP = AS · TOD · CTOO · $\overline{RT}$ · OASL	T	U63D, U63A/B, U62C
IOSB = AS · TOD · CTOO · $\overline{RT}$ · OASL	T	U63D, U63A/B, U62C
T5 Set Exit		
Exit = AS · CTR(14+15) · IIRX · T5	T	U31C, U14D
T6T7 Increment P, M		
RPRB = Exit · T6T7	L	U15C
SB0 = Exit · T6T7	L	U33A
ADF = Exit · T6T7	L	U25C
SWSP = Exit · T7	L	U16C
SWSM = Exit · T7	L	U55D
T7S Enable CPU		
$\overline{IIR}$ = Exit · T7 · T5	T	U83A, U73
$\overline{AS}$ = Exit · T7	T	U83B

*T= EAU Timing Card
L= EAU Logic Card

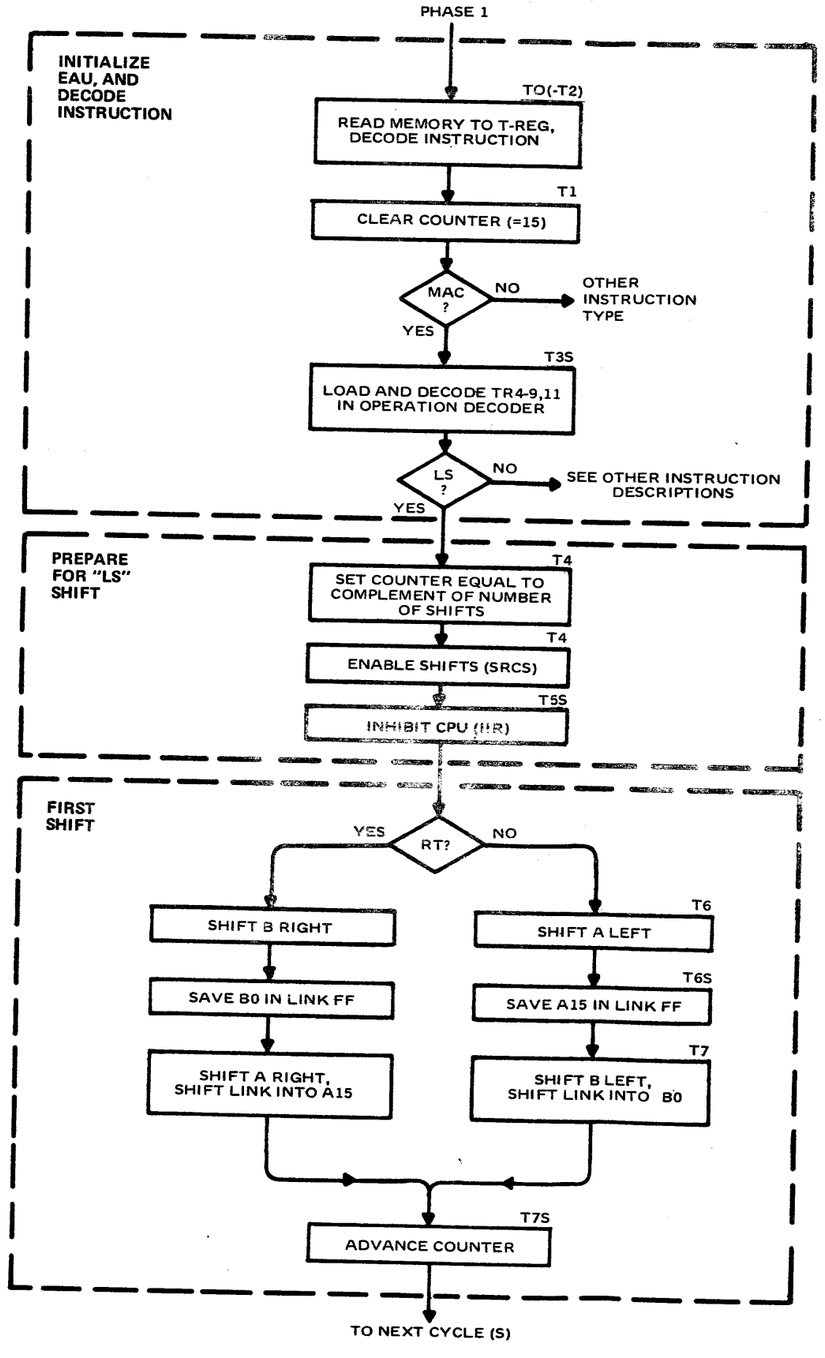


Figure 5-40. Logical Shifts, Phase 1

Table 5-25. Logical Shift Equations (1)

EQUATION	CARD*	REFERENCE DESIGNATION
T1 Clear Counter		
CTR(15) = PH1 · T1	T	U42C
T3S Decode LS		
LS = MAC · TR5	T	U105
Clock = PH1 · T3 · TS	T	U85C
T4 Set Counter		
CTR(#) = LS · PH1 · T4	T	U97A
T4 Enable Shifts		
SRCS = LS · CTO0	T	U86A
T5S Inhibit CPU		
IIR = LS · T5 · TS	T	U76C, U77
RIGHT SHIFT EQUATIONS (see next page for left shift equations)		
TEV Shift B		
RBRB = SRCS · RT · TEV	L	U74A, U125B, U125A
SRMB = SRCS · RT	T	U86B
SWSB = SRCS · RT · TEV	L	U126A, U125B, U125A
LINK = SRCS · RT · TEV · TS · RB0	L	U124A, U123C, U103B
TOD Shift A		
RARB = SRCS · RT · TOD	L	U91D, U122A
SRMB = SRCS · RT	T	U86B
TB15 = SRCS · RT · TOD · LINK	L	U127B
SWSA = SRCS · RT · TOD	L	U94B, U122A
T7S Advance Counter		
Clock = LS · TOD · TS · CTO0	T	U96A

* T = EAU Timing Card
L = EAU Logic Card

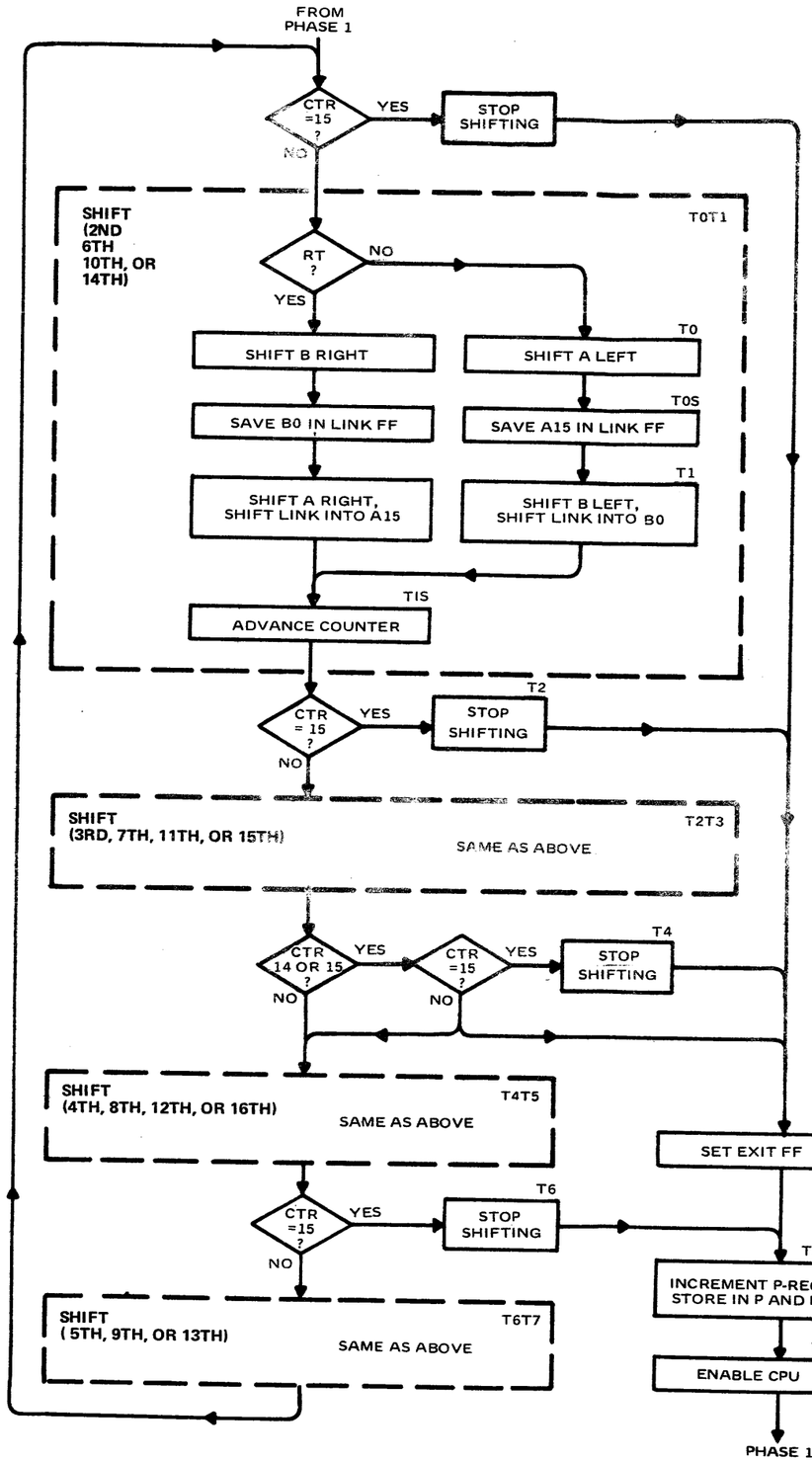


Figure 5-41. Logical Shift Loops

Table 5-26. Logical Shift Equations (2)

EQUATION	CARD*	REFERENCE DESIGNATION
LEFT SHIFT EQUATIONS (see preceding page for right shift equations)		
TEV Shift A		
RARB = SRCS · TEV · $\overline{RT}$	L	U91D, U114C
SLMB = SRCS · $\overline{RT}$	T	U86C
SL14B = LS · CT00 · $\overline{RT}$	T	U65C
SWSA = SRCS · TEV · $\overline{RT}$	L	U94B, U114C
LINK = SRCS · TEV · TS · $\overline{RT}$ · RB15	L	U124A, U123C, U103A
TOD Shift B		
RBRB = SRCS · TOD · $\overline{RT}$	L	U74A, U125C
SLMB = SRCS · $\overline{RT}$	T	U86C
SL14B = LS · CT00 · $\overline{RT}$	T	U65C
TB0 = SRCS · TOD · $\overline{RT}$ · LINK	L	U115D, U114B, U115A
SWSB = SRCS · TOD · $\overline{RT}$	L	U76B, U125C
Stop Shift		
$\overline{SRCS}$ = LS · $\overline{CT00}$	T	U86A
T5 Set Exit		
Exit = LS · CTR(14:15) · IIRX · T5	T	U31C, U14D
T6T7 Increment P, M		
RPRB = Exit · T6T7	L	U15C
SB1 = Exit · T6T7	L	U33A
ADF = Exit · T6T7	L	U25C
SWSP = Exit · T7	L	U16C
SWSM = Exit · T7	L	U55D
T7S Enable CPU		
$\overline{IIR}$ = Exit · T7 · T5	T	U83A, U73
$\overline{LS}$ = Exit · T7	T	U83B

* T - LAU Timing Card
L - LAJ Logic Card

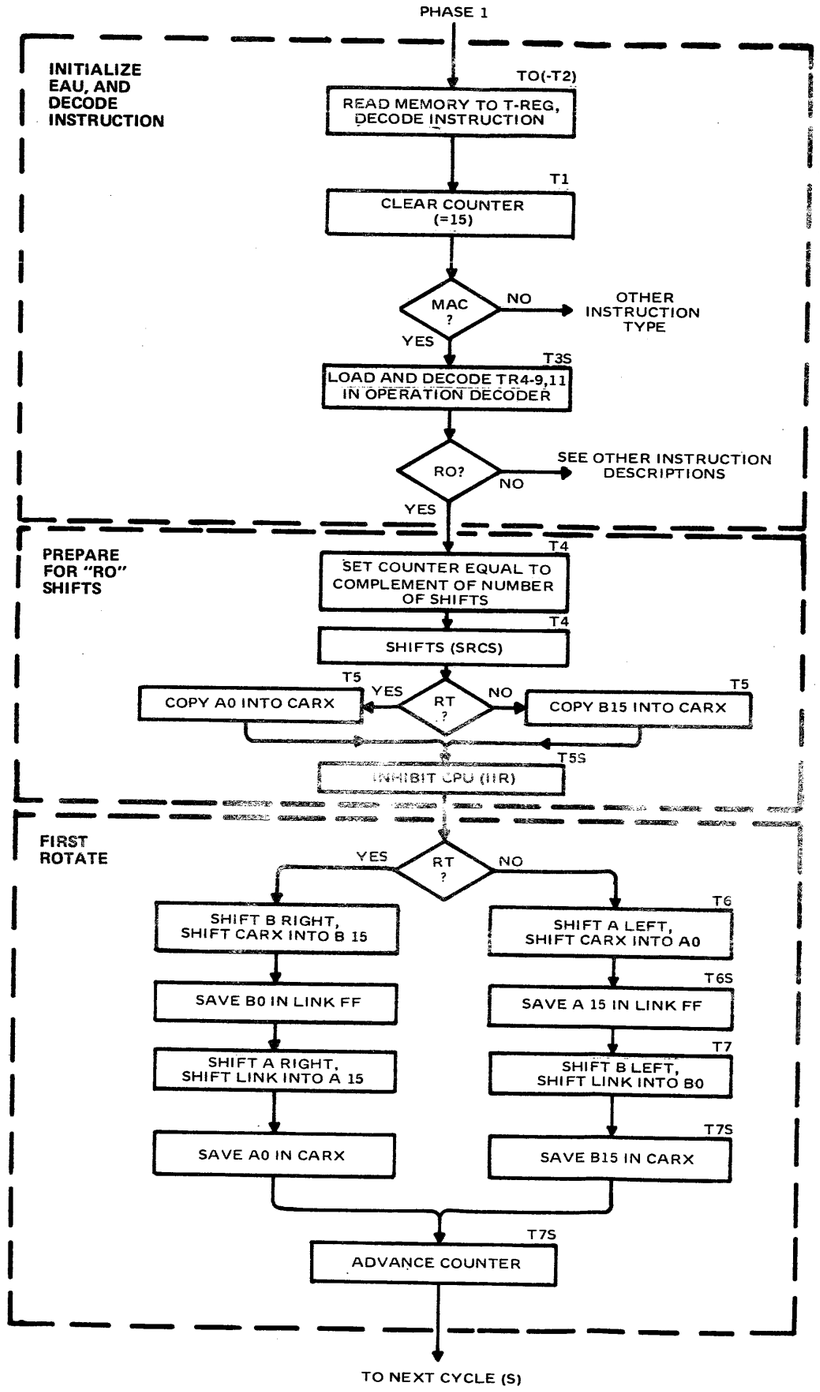


Figure 5-42. Rotates, Phase 1

Table 5-27. Rotate Equations (1)

EQUATION	CARD*	REFERENCE DESIGNATION
T1 Clear Counter		
CTR(15) = PH1 · T1	T	U42C
T3S Decode RO		
RO = MAC · TR6	T	U105
Clock = PH1 · T3 · TS	T	U85C
T4 Set Counter		
CTR(#) = RO · PH1 · T4	T	U97A
T4 Enable Shifts		
SRCS = RO · CTOO	T	U86A
T5 Set Carry		
Either RARB = ROT5 · RT	L	U96A
Either CARX = ROT5 · RT · TS · RB0	L	U103C/B, U113A, U122B
Either RBRB = ROT5 · $\overline{RT}$	L	U85B
Either CARX = ROT5 · $\overline{RT}$ · TS · RB15	L	U103C/A, U113A, U122B
T5S Inhibit CPU		
IIR = RO · TS · TS	T	U76C, U77
RIGHT ROTATE EQUATIONS (see next page for left rotate equations)		
TEV Shift B		
RBRB = SRCS · RT · TEV	L	U74A, U125B, U125A
SRMB = SRCS · RT	T	U86B
TB15 = SRCS · RO · RT · TEV · CARX	L	U127A, U125A
SWSB = SRCS · RT · TEV	L	U126A, U125B, U125A
LINK = SRCS · RT · TEV · TS · RB0	L	U124A, U123C, U103B
TOD Shift A		
RARB = SRCS · RT · TOD	L	U91D, U122A
SAMB = SRCS · RT	T	U86B
TB15 = SRCS · RT · TOD · LINK	L	U127B
SWSA = SRCS · RT · TOD	L	U94B, U122A
CARX = RO · TOD · TS · RT · TB0	L	U113A/C, U124B/C

* T = EAU Timing Card
L = EAU Logic Card

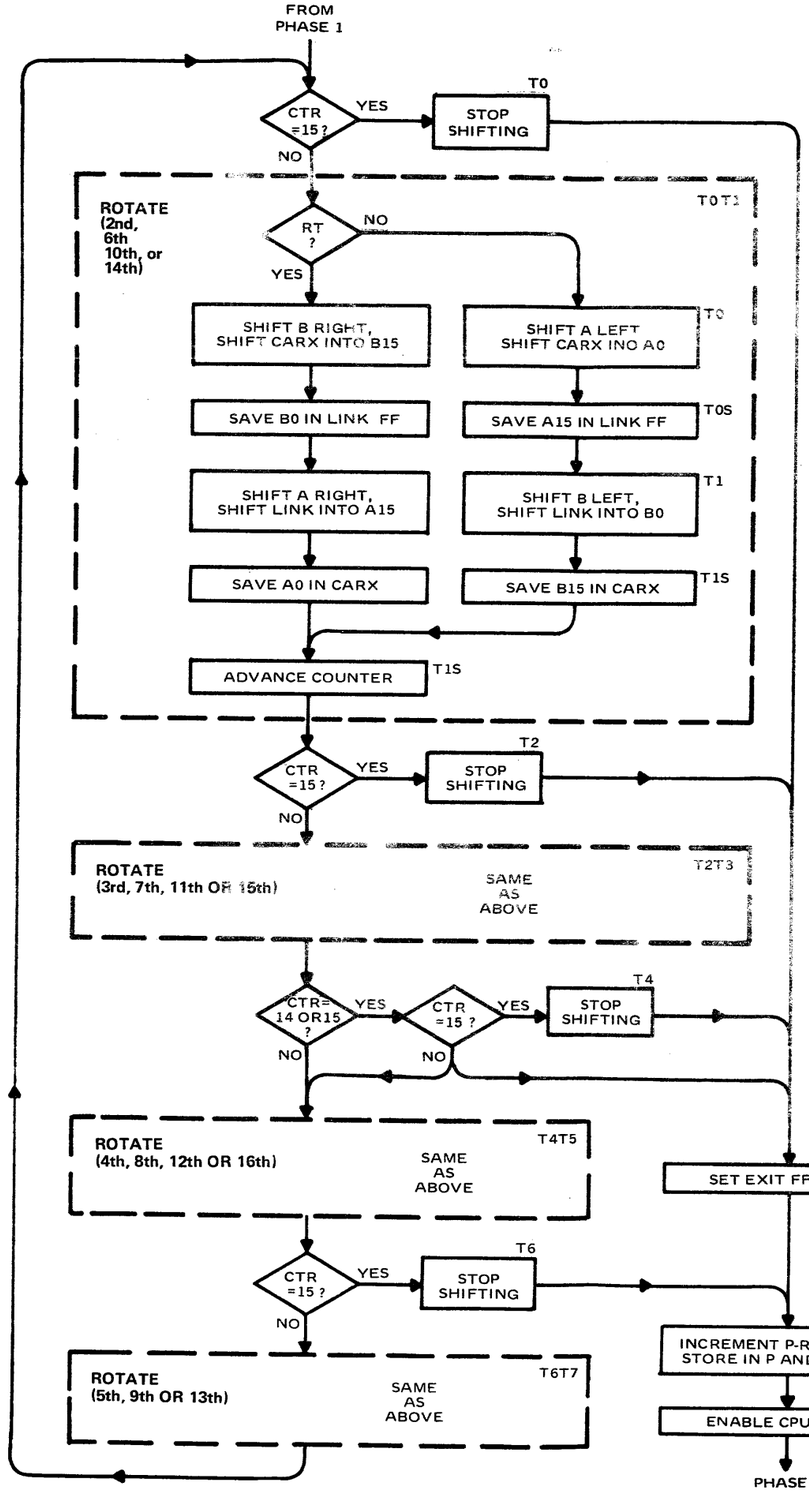


Figure 5-43. Rotate Loops

Table 5-28. Rotate Equations (2)

EQUATION	CARD*	REFERENCE DESIGNATION
LEFT ROTATE EQUATIONS (see preceding page for right rotate equations)		
TEV Shift A		
RARB = SRCS · TEV · $\overline{RT}$	L	U91D, U114C
SLMB = SRCS · $\overline{RT}$	T	U86C
SL14B = RO · CTO0 · $\overline{RT}$	T	U65C
TB0 = SRCS · TEV · $\overline{RT}$ · CARX	L	U115D, U114A, U114B
SWSA = SRCS · TEV · $\overline{RT}$	L	U94B, U114C
LINK = SRCS · TEV · TS · $\overline{RT}$ · RB15	L	U124A, U123C, U103A
TOD Shift B		
RBRB = SRCS · TOD · $\overline{RT}$	L	U74A, U125C
SLMB = SRCS · $\overline{RT}$	T	U86C
SL14B = RO · CTO0 · $\overline{RT}$	T	U65C
TB0 = SRCS · TOD · $\overline{RT}$ · LINK	L	U115D, U114B, U115A
SWSB = SRCS · TOD · $\overline{RT}$	L	U76B, U125C
CARX = RO · TOD · TS · $\overline{RT}$ · TB15	L	U113A/C, U124B/D
Stop Shift		
$\overline{SRCS}$ = RO · $\overline{CTO0}$	T	U86A
T5 Set Exit		
Exit = RO · CTR(14+15) · IIRX · T5	T	U31C, U14D
T6T7 Increment P, M		
RPRB = Exit · T6T7	L	U15C
SB0 = Exit · T6T7	L	U33A
ADF = Exit · T6T7	L	U25C
SWSP = Exit · T7	L	U16C
SWSM = Exit · T7	L	U55D
T7S Enable CPU		
$\overline{IIR}$ = Exit · T7 · TS	T	U83A, U73
$\overline{RO}$ = Exit · T7	T	U83B

* T = EAU Timing Card
L = EAU Logic Card

SECTION VI

MAINTENANCE

6-1. INTRODUCTION.

6-2. This section provides general maintenance and adjustment procedures for the HP 2152A Floating Point Processor.

6-3. ACCESS TO ASSEMBLIES.

6-4. Access to nearly all internal assemblies in the FPP unit is accomplished by removing the top cover of of the unit. The only exception is the capacitor assembly of the power supply. Paragraph 6-25 includes a removal procedure for these three capacitors.

6-5. Removal of the top cover exposes the logic cards (front half of the cabinet) and the power supply cover (rear half) which must also be removed for access to the power supply. The eleven logic cards are of the same size and construction as the cards in the computer. The cards are installed and removed (by extractor handles) in the same way as the computer cards. Note, however, that there are three printed circuit jumper plugs, joining pairs of cards via a top edge connector. These plugs must be removed before extracting the attached cards. If any of the jumpered cards is to be tested using the supplied extender, a short jumper cable (also supplied) is to be used to complete the connection between the card pairs.

6-6. Except for ac circuitry and some large components, most of the power supply circuits are located on the face-up printed circuit card in the power supply section. All of the FPP unit adjustments are located on this card. (See figure 6-1.)

6-7. PREVENTIVE MAINTENANCE.

6-8. The FPP unit requires a minimum of preventive maintenance. The following routine maintenance should be performed every 90 days:

a. Clean air filters. If environment is unusually dusty, this may be necessary more frequently than 90 days. Also clean interior of cabinet with low-pressure, dry, compressed air.

b. Perform the voltage checks and adjustments described in paragraph 6-9.

c. Perform the diagnostic tests as described in the Manual of Diagnostics.

6-9. VOLTAGE CHECKS AND ADJUSTMENTS.

6-10. There are seven adjustments in the FPP unit, all in the power supply. Three of these are voltage-setting adjustments; the remaining four set threshold levels for power fail detection.

6-11. The level of the three dc voltages may be checked at the test points on the rear panel of the FPP unit without removing the cover. (The ac voltage of the transformer secondary is available at one of the test points for troubleshooting purposes; it is not used for adjustments.) If the dc voltages are out of the tolerance limits specified in table 6-1, perform the appropriate adjustment (paragraphs 6-12 through 6-17). These three adjustments may be made independently, except for the +10V supply; if the +10V level is adjusted, the +4.75V and -2V levels must also be adjusted. Table 6-1 also provides specifications on current and ac ripple for information purposes only; there are no adjustments involved.

6-12. +10V ADJUSTMENT.

6-13. Variable resistor R54 on regulator card A3 (see figure 6-1) allows adjustment of the +10V supply. Using a voltmeter of at least 0.1 percent accuracy, measure the voltage at the 10V test point. Using a suitable nonmetallic tool, adjust R54 (+10V ADJ) for +10.0 volts on the voltmeter. This adjustment alters the +4.75V and -2V levels; adjust these two supplies as described in the following paragraphs.

6-14. +4.75V ADJUSTMENT.

Table 6-1. DC Voltage and Current Specifications

SUPPLY	MAXIMUM POSITIVE	MAXIMUM NEGATIVE	AC RIPPLE (P-P, TYPICAL)	MAXIMUM CURRENT
-2V	-1.9V	-2.1V	20 mV	25A
+4.75V	+4.9V	+4.6V	50 mV	25A
+10V	+10.2V	+9.8V	10 mV	0.1A

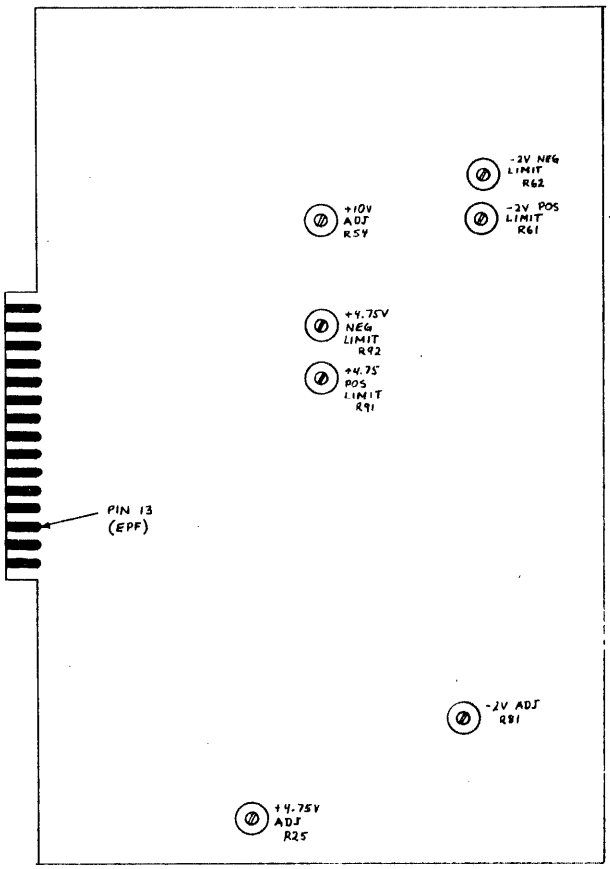


Figure 6-1. Location of Adjustments

6-15. Measure the voltage at the +4.75V test point and adjust R25 (+4.75V ADJ) for +4.75 volts on the voltmeter.

6-16. -2V ADJUSTMENT.

6-17. Measure the voltage at the -2V test point and adjust R81 (-2V ADJ) for -2.0 volts on the voltmeter.

6-18. POWER FAIL THRESHOLD ADJUSTMENTS.

6-19. The power fail threshold adjustments are set at the factory prior to shipment of the FPP unit and should require no further adjustment. However, if adjustment should become necessary, such as due to replacement of components, the following procedures may be used. All four adjustments may be made independently.

6-20. Use of voltmeter of at least 0.1 percent accuracy for setting the dc supply levels. This accuracy is necessary for the final settings only, so that the supply voltages are left at the correct values at the end of the procedure. During the procedure, however, some imprecision of the power fail trigger points should be expected. A second voltmeter is necessary in order to monitor the power fail signal; accuracy is unimportant since only a change of state is to be read (between about +4.5 volts and 0 volts). Use a suitable nonmetallic tool for making the adjustments.

6-21. +4.75V LIMIT ADJUSTMENTS.

6-22. To adjust the maximum positive and negative limits of the +4.75V supply, measure both the voltage at the +4.75V test point and the voltage at pin 13 of regulator card A3 (EPF signal). Then proceed as follows:

a. Note the level of the signal at pin 13 of A3; the reading should be 0 volts.

b. Adjust R25 (+4.75V ADJ) to increase the voltage at the +4.75V test point to +5.25 volts.

c. Adjust R91 (+4.75V POS LIMIT) just past the point where the voltage at pin 13 of A3 (noted in step "a") abruptly jumps to about +4.5 volts.

d. Change the setting of R25 (+4.75V ADJ) to lower the voltage at the +4.75V test point to +4.35 volts.

e. Adjust R92 (+4.75V NEG LIMIT) just past the point where the voltage at pin 13 of A3 (noted in step "a" abruptly jumps to about +4.5 volts.

f. Recheck adjustments by varying the +4.75V level at the test point, both high and low. The voltage at pin 13 of A3 jumps to +4.5 volts when the supply voltage reaches +5.25 volts on the high side and +4.35 volts on the low side.

g. Return the level at the +4.75V test point to exactly +4.75 volts.

h. Disconnect the voltmeters.

6-23. -2V LIMIT ADJUSTMENTS.

6-24. To adjust the maximum positive and negative limits of the -2V supply, measure both the voltage at the -2V test point and the voltage at pin 13 of regulator card A3 (EPF signal). Then proceed as follows:

a. Note the level of the signal at pin 13 of A3; the reading should be 0 volts.

b. Adjust R81 (-2V ADJ) to change the voltage at the -2V test point to -1.8 volts.

c. Adjust R61 (-2V POS LIMIT) just past the point where the voltage at pin 13 of A3 (noted in step "a") abruptly jumps to about +4.5 volts.

d. Change the setting of R81 (-2V ADJ) to change the voltage at the -2V test point to -2.2 volts.

e. Adjust R62 (-2V NEG LIMIT) just past the point where the voltage at pin 13 of A3 (noted in step "a") abruptly jumps to about +4.5 volts.

f. Recheck adjustments by varying the -2V level at the test point, both high and low. The voltage at pin 13 of A3 should jump to +4.5 volts when the supply voltage reaches -1.8 volts on the more positive side and -2.2 volts on the more negative side.

g. Return the level at the -2V test point to exactly -2.0 volts.

h. Disconnect the voltmeters.

6-25. REPLACEMENT OF CAPACITOR ASSEMBLY A5.

6-26. If it becomes necessary to check or replace any of the three large filter capacitors (C11, C12, C13, each 0.16 farad), use the following procedure:

a. Remove the top cover and the plate covering the power supply.

b. Remove regulator board A3 by removing the two hold-down screws and unplugging from its receptacle.

c. Remove the eight screws attaching the rear panel. This panel can now swing down, connected only by two cables.

d. Remove the four screws attaching the cable lugs to the capacitor terminals.

e. Remove the six screws holding the capacitor assembly onto the chassis.

f. Slide the capacitor assembly out toward the rear. (Note the orientation of the cutout in the top insulators and conductor straps. This cutout provides clearance for one of the terminal block screws on the chassis, and must be preserved when reassembling.)

g. Remove the remaining two screws which attach the insulators and conductor straps on the top of the assembly. (Note the orientation of the + terminals of the capacitors; retain this orientation before and after test or replacement.) The capacitors may now be individually removed by sliding up out of the assembly.

6-27. To reassemble and install the assembly after test or replacement, use the following procedure:

a. On top of the assembly, place (first) one insulator, positioned so that the clearance cutout (noted in step "f" above) is toward the negative side of the capacitors.

b. Then place the two conductor straps and (last) the other insulator on top of the assembly, carefully positioning the clearance cutouts directly above the cutout of the insulator installed in step "a".

c. Position the assembly so the side having the cutout (negative side) is to the left, and the positive side is to the right. Attach the insulators and conductor straps with two screws onto the terminals of the nearest of the three capacitors.

d. Slide the assembly into the chassis guides, with the negative side (having the cutout) still to the left.

e. Attach the assembly to the chassis with the six screws removed in step "e" of paragraph 6-26.

f. Attach the cable lugs to the four capacitor terminals, using the four screws removed in step d of paragraph 6-26.

g. Replace the rear panel (eight screws).

h. Plug the regulator board into its receptacle and secure with the two hold-down screws.

i. Replace the power supply cover and the top cover.

SECTION VII

MAINTENANCE DATA

7-1. INTRODUCTION.

7-2. This section consists of reference tables and diagrams to be used in support of maintenance for the HP2152A Floating Point Processor. Refer to the preceding section for maintenance procedures.

7-3. Specifically, this section includes the logic diagram for each printed circuit card, a table of replaceable parts for each card, and a component location diagram for each card. In addition, the following are included: backplane wiring diagram, complete power supply schematic diagram, and listings in both numeric and mnemonic form of the contents of the read-only memory.

Table 7-1. EAU Timing Card, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A110	02152-60012		EAU TIMING CARD	28480	02152-60012
A110C1					
A110C6	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
A110U12	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A110U13	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A110U14	1820-0966	28	INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U15	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U16	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U17	1820-0956	9	IC:CTL DUAL 2-INPUT AND BUFFER	07263	SL3459
A110U22	1820-0965	15	IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A110U23	1820-0964	14	INTEGRATED CIRCUIT: CTL	07263	SL3461
A110U24	1820-0971	17	INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U25	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U26	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U27	1820-0956		IC:CTL DUAL 2-INPUT AND BUFFER	07263	SL3459
A110U32	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A110U33	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U34	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A110U35	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U36	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U37	1820-0956		IC:CTL DUAL 2-INPUT AND BUFFER	07263	SL3459
A110U42	1820-0952	8	IC:CTL DUAL 2-INPUT NOR GATE	07263	SL3455
A110U43	1820-0968	6	INTEGRATED CIRCUIT: CTL	07263	SL3466
A110U44	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A110U45	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A110U46	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U47	1820-0952		IC:CTL DUAL 2-INPUT NOR GATE	07263	SL3455
A110U51	1820-0954	7	INTEGRATED CIRCUIT: CTL	07263	SL3457
A110U52	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U53	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U54	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U55	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U56	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U57	1820-0956		IC:CTL DUAL 2-INPUT AND BUFFER	07263	SL3459
A110U61	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A110U62	1820-0952		IC:CTL DUAL 2-INPUT NOR GATE	07263	SL3455
A110U63	1820-0952		IC:CTL DUAL 2-INPUT NOR GATE	07263	SL3455
A110U64	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U65	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A110U66	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A110U67	1820-0956		IC:CTL DUAL 2-INPUT AND BUFFER	07263	SL3459
A110U72	1820-0952		IC:CTL DUAL 2-INPUT NOR GATE	07263	SL3455
A110U73	1820-0968		INTEGRATED CIRCUIT: CTL	07263	SL3466
A110U74	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A110U75	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U76	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A110U77	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U82	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U83	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U84	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U85	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U86	1820-0956		IC:CTL DUAL 2-INPUT AND BUFFER	07263	SL3459
A110U87	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U91	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A110U92	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A110U93	1820-0954		INTEGRATED CIRCUIT: CTL	07263	SL3457
A110U94	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A110U95	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U96	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U97	1820-0956		IC:CTL DUAL 2-INPUT AND BUFFER	07263	SL3459
A110U101	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A110U102	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A110U103	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U104	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U105	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U106	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U107	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U112	1820-0968		INTEGRATED CIRCUIT: CTL	07263	SL3466
A110U113	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U114	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A110U115	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U116	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A110U117	1820-0956		IC:CTL DUAL 2-INPUT AND BUFFER	07263	SL3459
A110U122	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A110U124	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A110U125	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A110U126	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A110U126	1820-0971		INTEGRATED CIRCUIT: CTL	07263	SL3467
A110U127	1820-0954		INTEGRATED CIRCUIT: CTL	07263	SL3457

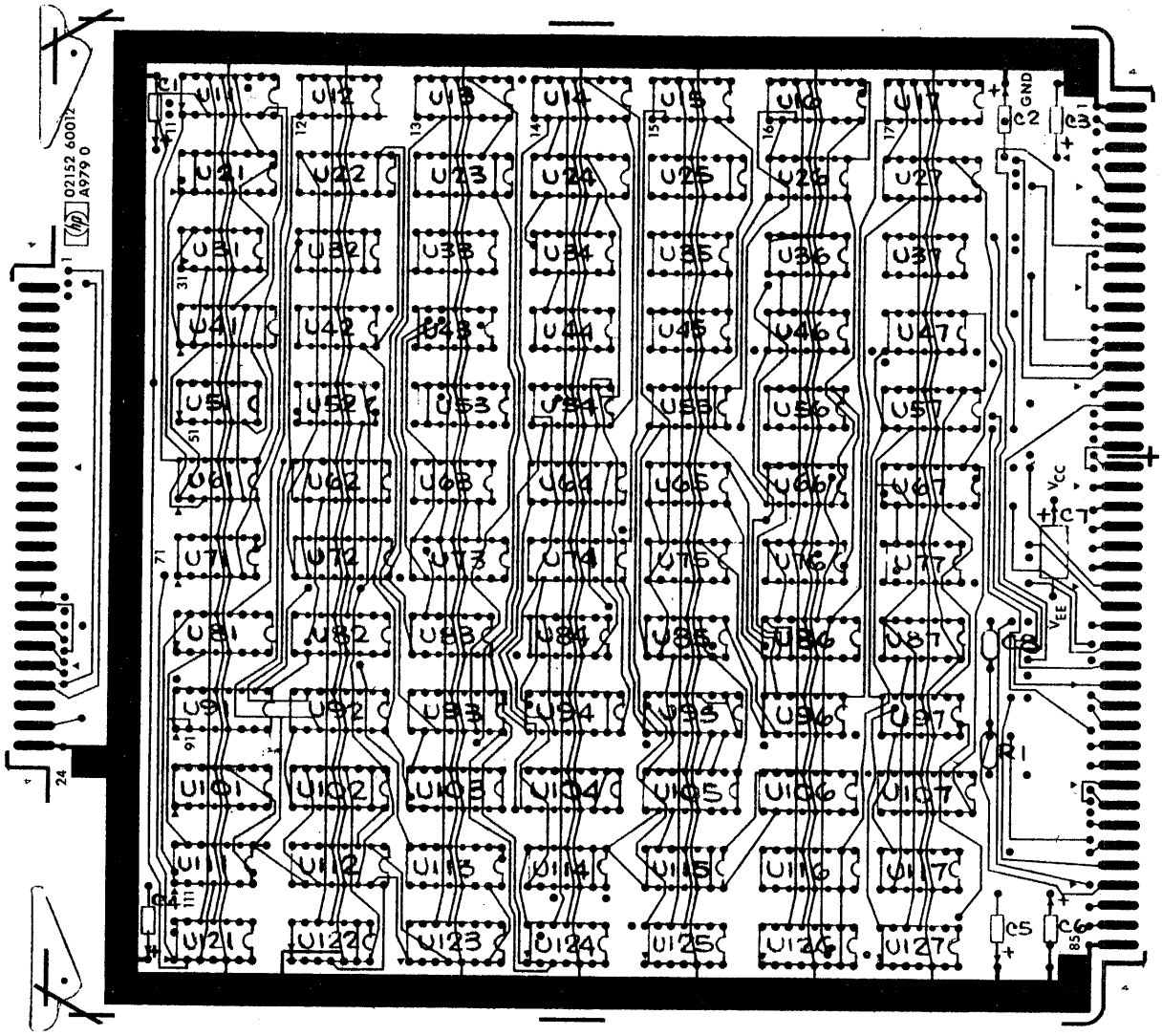
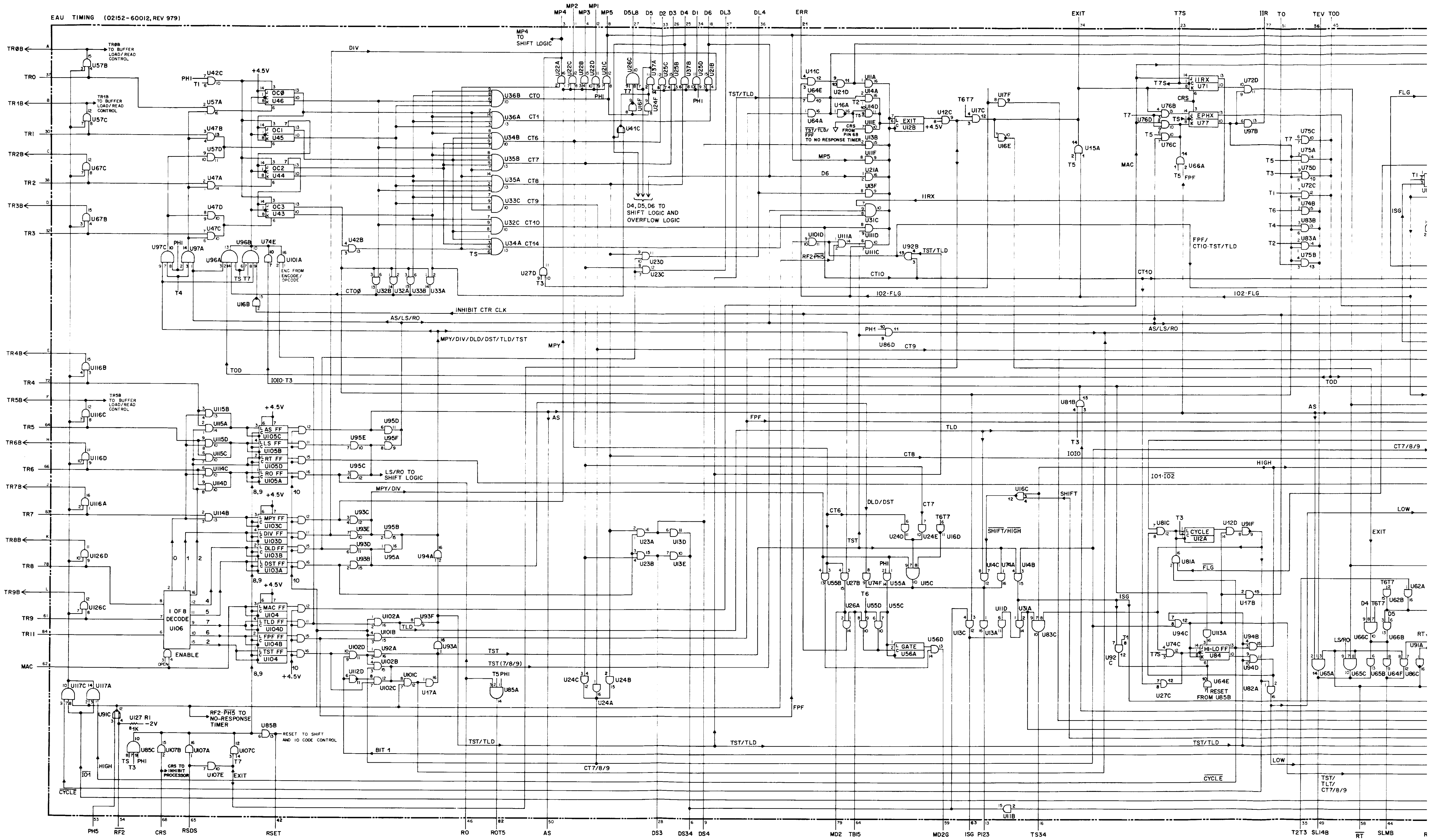


Figure 7-1. EAU Timing Card, Parts Location View


EAU TIMING (02152-60012, REV 979)
TR0B TO BUFFER LOAD/READ CONTROL
TR1B TO BUFFER LOAD/READ CONTROL
TR5B TO BUFFER LOAD/READ CONTROL
MP4 TO SHIFT LOGIC
D4, D5, D6 TO SHIFT LOGIC AND OVERFLOW LOGIC
TO NO RESPONSE TIMER
LS/RO TO SHIFT LOGIC
RF2·PH5 TO NO-RESPONSE TIMER
RESET TO SHIFT AND IO CODE CONTROL
CRS TO INHIBIT PROCESSOR
ENC FROM ENCODE/OPCODE
INHIBIT CTR CLK
AS/LS/RO
MPY/DIV/DLD/DST/TLD/TST
1 OF 8 DECODE U106
ENABLE
RESET FROM U85B
PH5 RF2 CRS RSDS RSET RO ROT5 AS DS3 DS34 DS4 MD2 TBI5 MD2G ISG PI23 TS34 T2T3 SLI4B RT SLMB

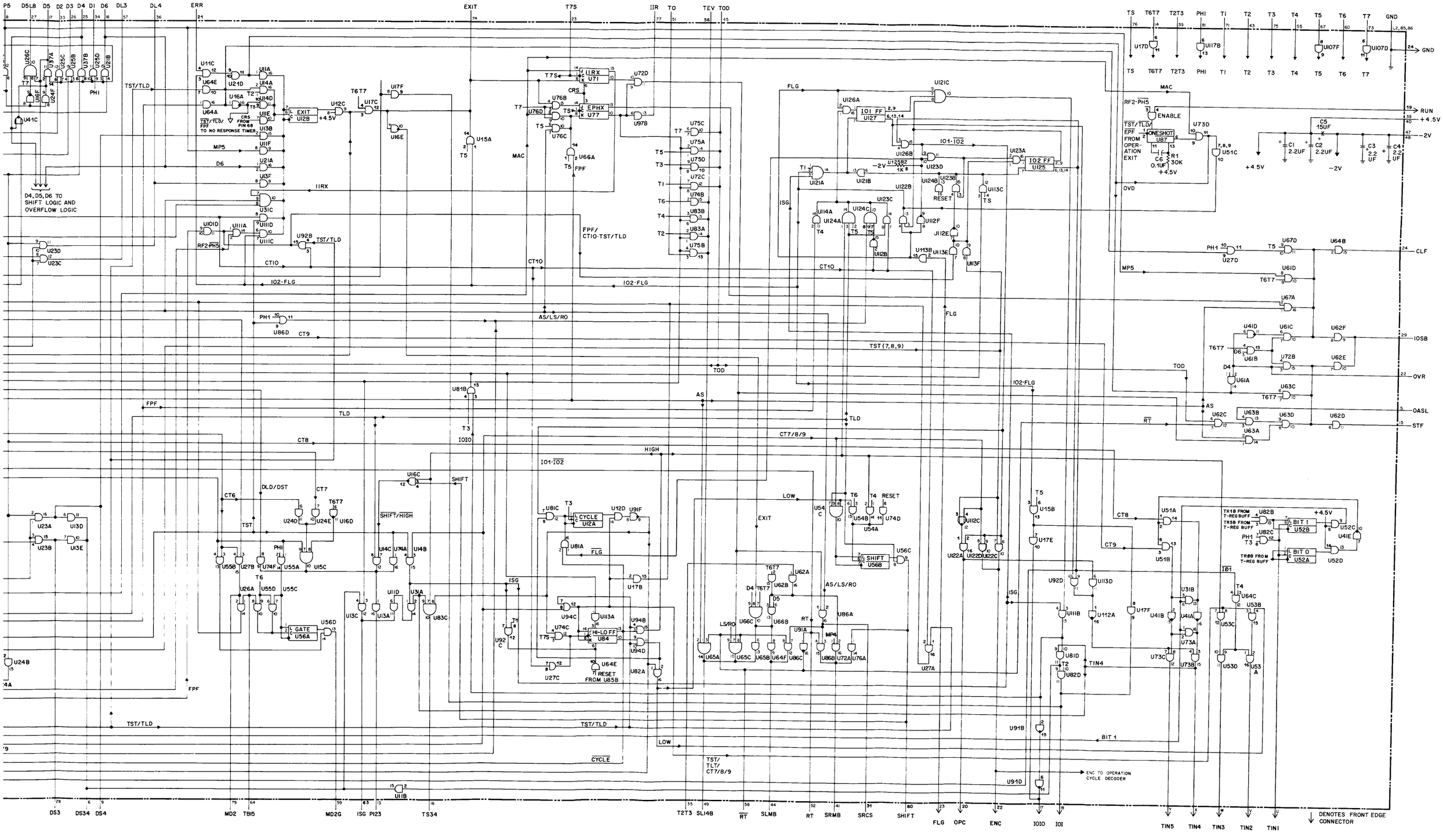


Figure 7-2. EAU Timing Card, Logic Diagram

Table 7-2. EAU Logic Card, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A109	02152-60011		EAU LOGIC CARD	28480	02152-60011
A109C1-A109C6	0180-0197		C:FXD ELECT 2.2UF 10% 20VDCW	56289	150D225X9020A2-DYS
A109C7	0180-1746	1	C:FXD ELECT 15 UF 10% 20VDCW	28480	0180-1746
A109C8	0150-0121		C:FXD CER 0.1 UF +80-20% 50VDCW	56289	5C50B1S-CML
A109R1	0683-3335	2	R:FXD COMP 33K OHM 5% 1/4W	01121	CB 3335
A109U11	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A109U12	1820-0968		INTEGRATED CIRCUIT: CTL	07263	SL3466
A109U13	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A109U14	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U15	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A109U16	1820-0520	4	INTEGRATED CIRCUIT: CTL	07263	SL15960
A109U17	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A109U21	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U22	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U23	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U24	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A109U25	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U26	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A109U27	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U31	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A109U32	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A109U33	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A109U34	1820-0954		INTEGRATED CIRCUIT: CTL	07263	SL3457
A109U35	1820-0954		INTEGRATED CIRCUIT: CTL	07263	SL3457
A109U36	1820-0954		INTEGRATED CIRCUIT: CTL	07263	SL3457
A109U37	1820-0956		IC:CTL DUAL 2-INPUT AND BUFFER	07263	SL3459
A109U41	1820-0520		INTEGRATED CIRCUIT: CTL	07263	SL15960
A109U42	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A109U43					
A109U46	1820-0967		IC:CTL DUAL RANK J-K FLIP-FLOP	07263	SL3464
A109U47	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A109U51	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A109U52	1820-0968		INTEGRATED CIRCUIT: CTL	07263	SL3466
A109U53	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U54	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A109U55	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A109U56	1820-0968		INTEGRATED CIRCUIT: CTL	07263	SL3466
A109U57	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U61	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A109U62	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A109U63	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A109U64	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A109U65	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A109U66	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A109U67	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U71	1820-0967		IC:CTL DUAL RANK J-K FLIP-FLOP	07263	SL3464
A109U72	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U73	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U74	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A109U75	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A109U76	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A109U77	1820-0967		IC:CTL DUAL RANK J-K FLIP-FLOP	07263	SL3464
A109U81	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U82	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U83	1820-0953		IC:CTL TRIPLE 2-2-3INPUT AND GATE	07263	SL3456
A109U84	1820-0967		INTEGRATED CIRCUIT: CTL	07263	SL3464
A109U85	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A109U86	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U87	1820-0207	1	IC:TTL MONOSTABLE MULTIVIBRATOR	07263	SL12895
A109U91	1820-0520		INTEGRATED CIRCUIT: CTL	07263	SL15960
A109U92	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U93	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A109U94	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U95	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A109U96	1820-0954		INTEGRATED CIRCUIT: CTL	07263	SL3457
A109U97	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A109U101	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U102	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U103					
A109U105	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A109U106	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
A109U107	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A109U111	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A109U112	1820-0520		INTEGRATED CIRCUIT: CTL	07263	SL15960
A109U113	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A109U114	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A109U115	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A109U116	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U117	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A109U121	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A109U122	1820-0966		INTEGRATED CIRCUIT: CTL	07263	SL3463
A109U123	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U124	1820-0964		INTEGRATED CIRCUIT: CTL	07263	SL3461
A109U125	1820-0952		IC:CTL DUAL 2-INPUT NOR GATE	07263	SL3455
A109U125	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A109U127	1820-0952		IC:CTL DUAL 2-INPUT NOR GATE	07263	SL3455

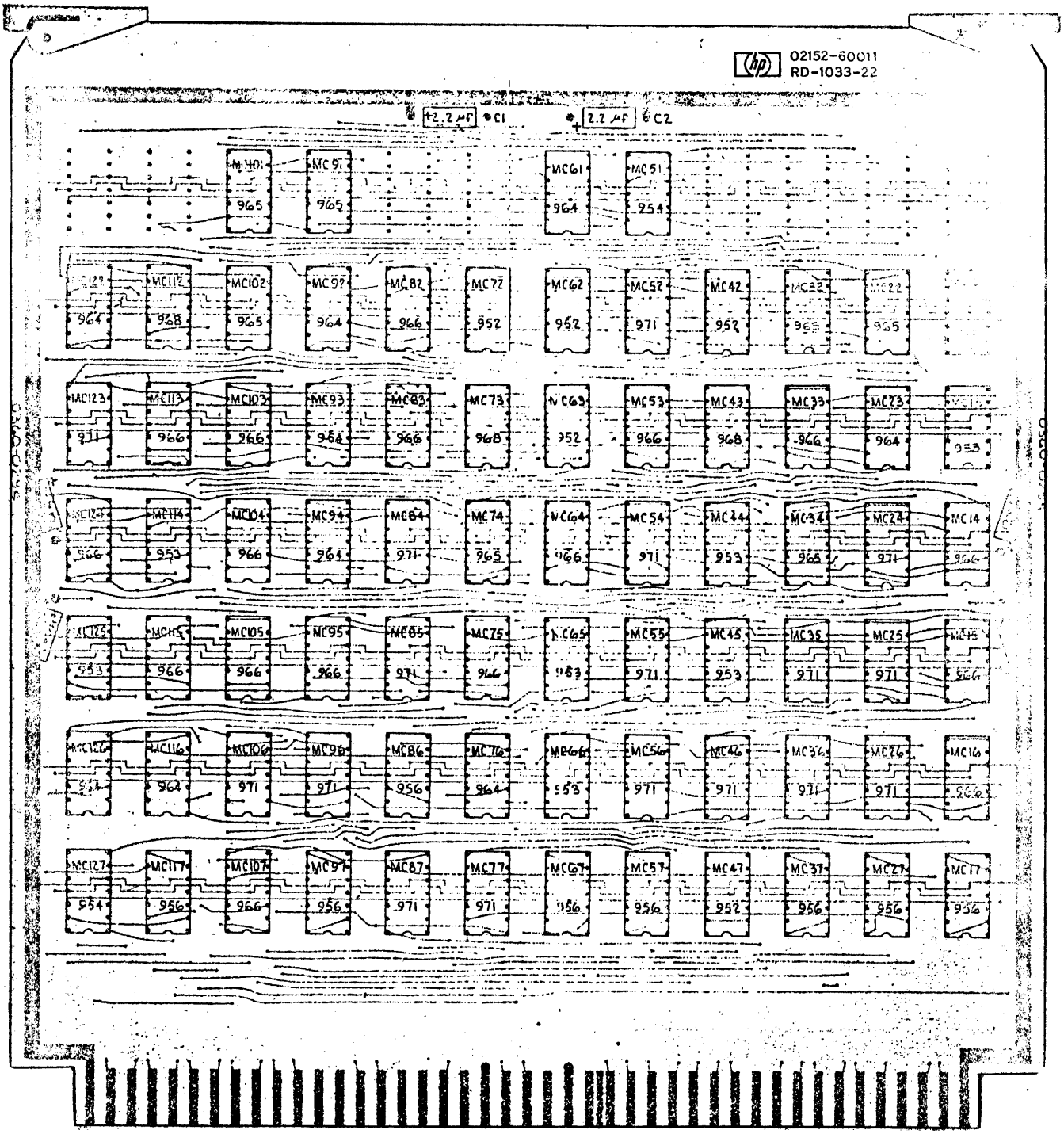


Figure 7-3. EAU Logic Card, Parts Location View

EAU LOGIC CARD (02152-60011, REV 1033)

EAU TIMING SIGNALS FROM TIMING BOARD (02116-6196)

MP1 59 → MP1
MP2 5 → MP2
MP3 69 → MP3
MP4 46 → MP4
MP5 9 → MP5
DI 62 → DI
D2 60 → D2
D3 27 → D3
D4 6 → D4
D5 73 → D5
D6 8 → D6
DS3 67 → DS3
DS4 52 → DS4
DS34 14 → DS34
DL3 70 → DL3
DL4 18 → DL4
AS 84 → AS
RO 75 → RO
$\overline{RT}$ 78 → $\overline{RT}$
RT 57 → RT
ROT5 68 → ROT5
TEV 81 → TEV
TOD 82 → TOD
SRCS 83 → SRCS
D5L8 71 → D5L8
GATE 76 → GATE
EXIT 17 → EXIT
MD2 16 → MD2
TS 29 → TS FROM COMPUTER
RBI5 77 → RBI5
CI6 50 → CI6
TO 15 → TO
TOTI 36 → TOTI
TI 13 → TI
T5 54 → T5
T3 32 → T3
T7 28 → T7
TIT2 51 → TIT2
T3T4 74 → T3T4
T4T5 23 → T4T5
T6 30 → T6
T6T7 1 → T6T7
T2 22 → T2
T4 24 → T4
T2T3 21 → T2T3
+4.5V 39, 40 → +4.5V
-2V 47, 48 → -2V
GRD 2, 85, 86 → GRD

SBO 44
TBI5 80
TBO 79
SWSA 66
SWSB 10
RBO 43
RSET 63
OVR 55
OVD 56
$\overline{OVR}$ 64
RB 4

MC65A, MC33A, MC65B, MC67A, MC23B, MC33D, MC33C, MC65C, MC51B
MC126C, MC127B, MC117A, MC127A, MC126A
MC114B, MC114A, MC115A, MC115D, MC115B, MC116C, MC116A, MC115C, MC117B
MC33B, MC23A, MAF FF, MC73A
MC22A, MC34A
MC93A, MC94A, MC94B, MC94C, MC107D, MC107C, MC97A
MC77D, MC77C, MC75A, MC75B, MC75C, MC75D, MC58
MC14C, MC76B, MC74D, MC76A, MC34C, MC34D
MC27A, MC51A
MC104A, MC91B
MC123O, MC123C, MC123A, MC123B, MC124A, MC112G2, LINK FF, MC112B
MC125A, MC125B, MC125C
MC105D, MC105C, MC105B, MC105A, MC107B, MC114C, MC122A
MC103A, MC103B
MC67B, MC64A, MC66B, MC66A, MC66C, TS34, T6T7
MC47B, MC47A, MC47R2 1K, -2V
MC47R1 1K
C1 2.2μF, C2 2.2μF, C3 2.2μF, C4 2.2μF, C5 2.2μF, C6 2.2μF
MC101D, MC101A, MC22C, MC22B
MC103C, MC124D, MC124C, MC124B, MC122B, MC113C, MC122C, MC113B, MC113D, MC102A, MC102D, MC102B, MC103D, CARX FF, MC112A, MC113A
MC52A, MC52B, MC52C, MC52D, MC53B, MC53A, SMD/SDD FF, MC43A, SMR/SDV FF, MC43B, MC42B, MC42R2 1K, MC53C, MC53D, MC42A, MC42R1 1K, MC107A
MC62A, MC62R1 1K, MC62B, MC62R2 1K, MC63A, MC63R1 1K, MC83B, MC83C, MC83D, MC82A, MC92A, MC92C, MC93B, MC83A, OVR, MC73B, MC63B, MC63R2 1K, MC82B, MC104B, MC72A, MC82D

MC103B: T∅ (7), T2 (8), (6), (9) → MC101C: T∅ + T2

(LINK)
(SMD)
(SIGN)
(CARRY)
(MP5+D6)
(MAF)
(SMD/SDD)
(SIGN)
(SMR-MP3)
(SMR/SDV)
(SMR/SDV)
(OVS)
SMR

DWG REV B

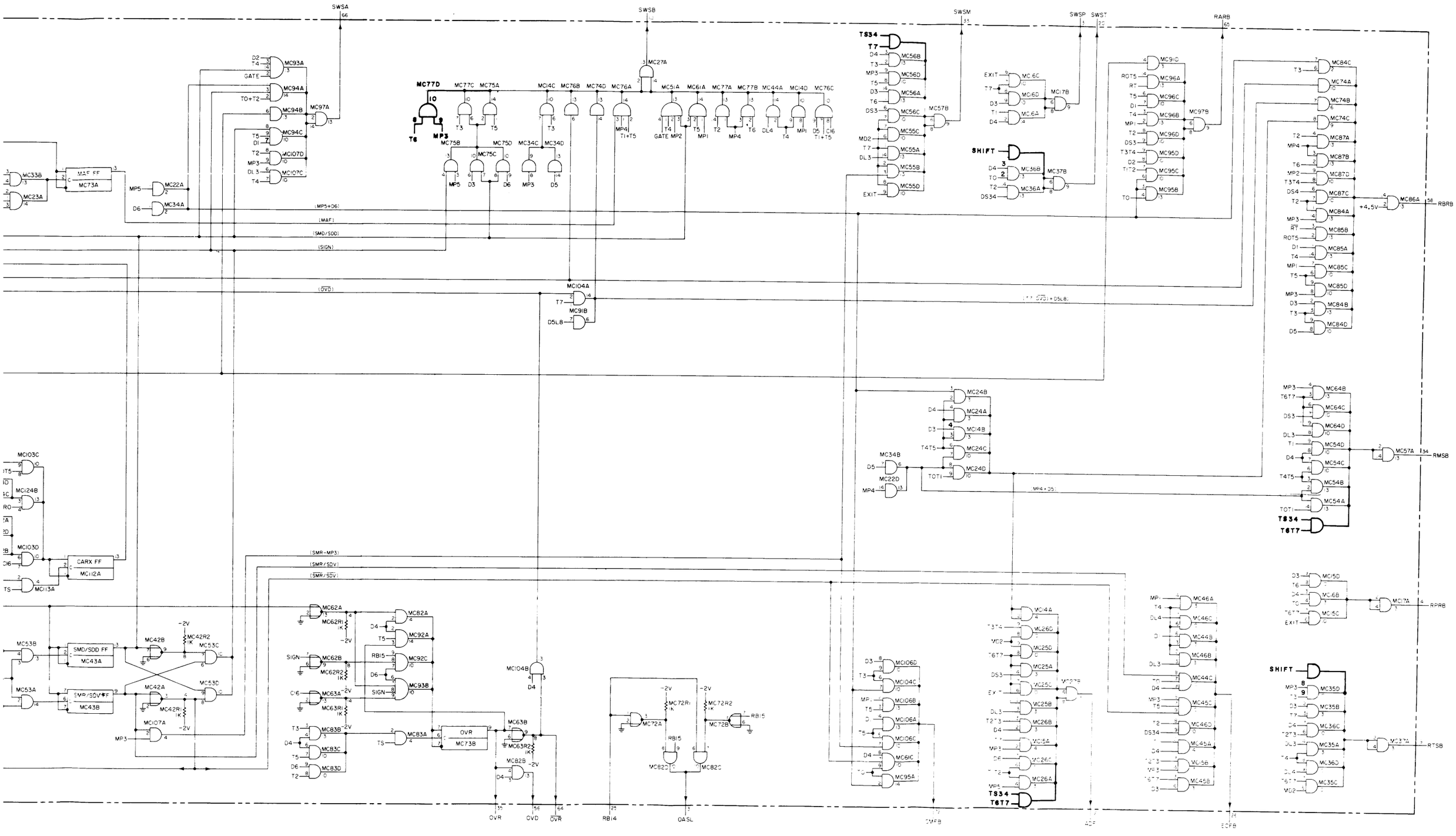


Figure 7-4. EAU Logic Card, Logic Diagram

Table 7-3. EAU Interface Card, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A209	02152-60013		EAU I/O CARD	28480	02152-60013
A209C1 - A209C5	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
A209C6	0150-0050		C:FXD CER DI:C 1000 PF +80-20% 1000VDCW	56289	C0678102E102ZE19-CDH
A209R1 - A209R39	0683-3915	39	R:FXD COMP 390 OHM 5% 1/4W	01121	CB 3915
A209R41	0683-1015		R:FXD COMP 100 OHM 5% 1/4W	01121	CB 1015
A209U22	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A209U23	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A209U32	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A209U33	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A209U34	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A209U42	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A209U43	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A209U44	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A209U52	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A209U53	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A209U54	1820-0952		IC:CTL DUAL 4-INPUT NOR GATE	07263	SL3455
A209U62	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A209U63	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A209U64	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961

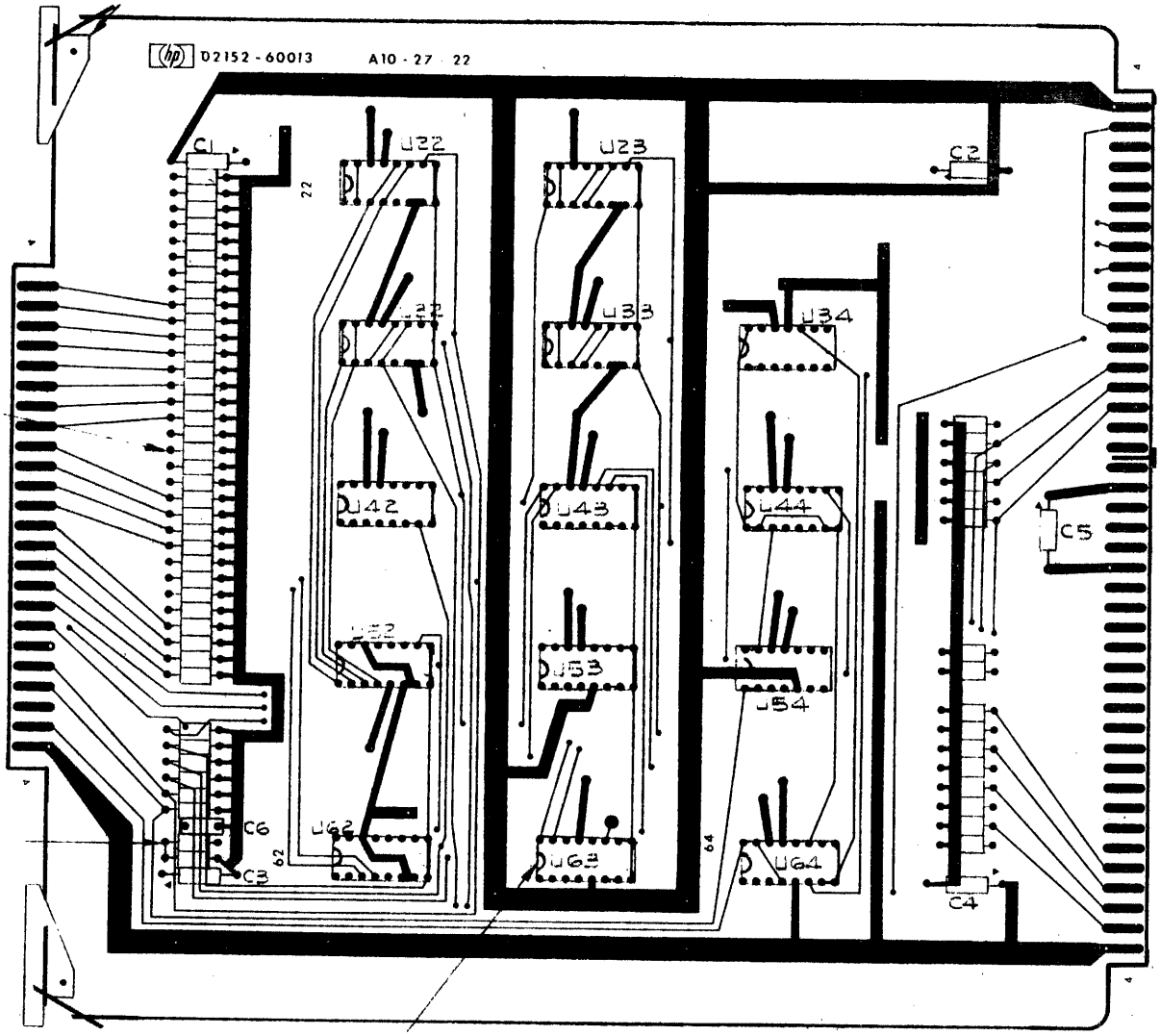
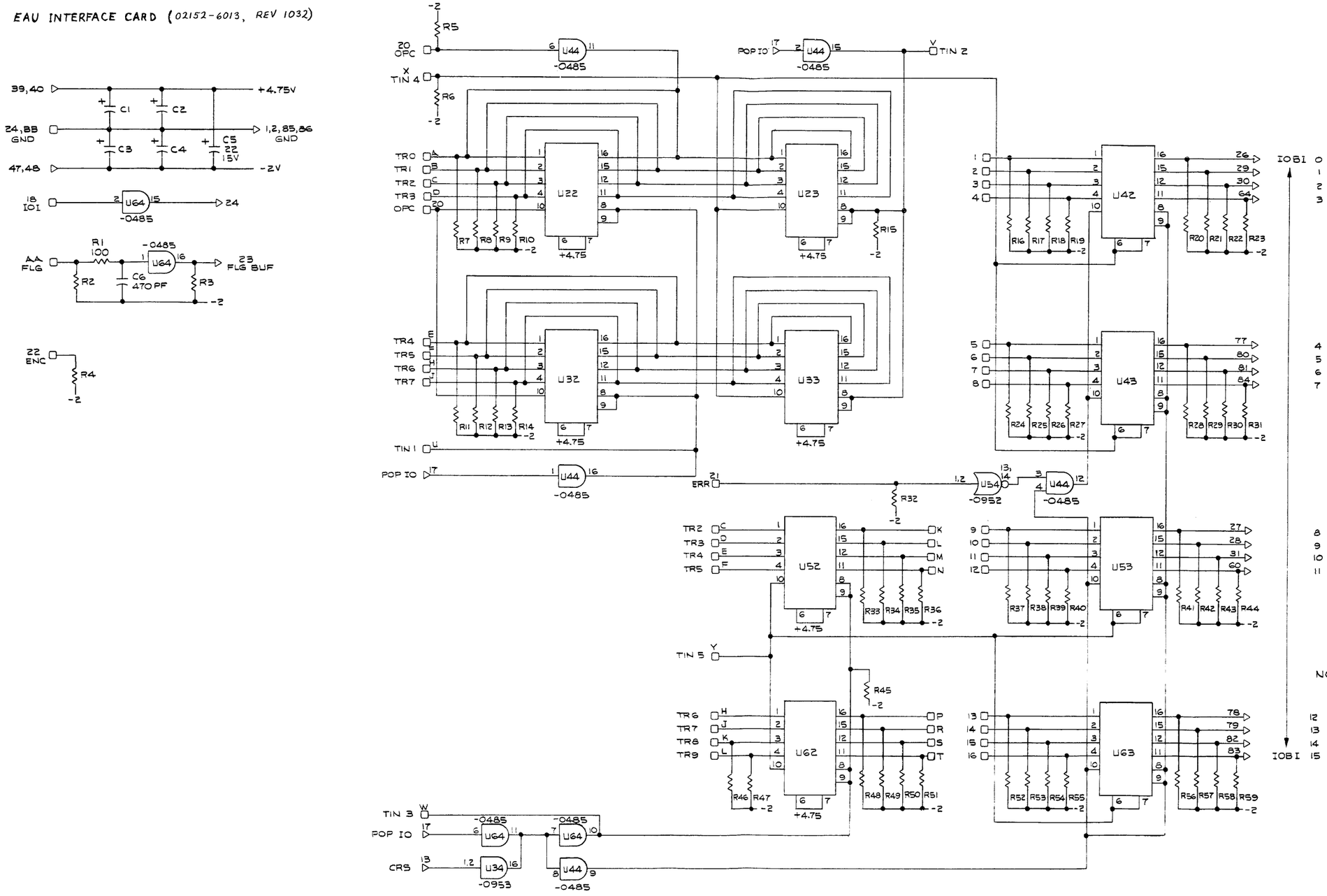


Figure 7-5. EAU Interface Card, Parts Location View

EAU INTERFACE CARD (02152-6013, REV 1032)



NOTE: UNLESS OTHERWISE SPECIFIED,

1. RESISTANCE IN OHMS, 1/4W, 5%.
2. CAPACITANCE IN MICROFARADS.
3. C1-4 ARE 2.2 UF, 20V.
4. R2 THRU R59 ARE 390Ω.
5. □ INDICATES 48 PIN CONNECTOR.
6. ▷ INDICATES 86 PIN CONNECTOR.
7. INTEGRATED CKTS ARE (1820-0486).

Figure 7-6. EAU Interface Card, Logic Diagram

Table 7-4. Test Card A7, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A7	02152-60008		TEST CARD	28480	02152-60008
A7C1-A7C4	0180-0197	46	C:FXD ELECT 2.2 UF 10% 20VDCW	56289	1500225X9020A2-DYS
A7C5-A7C11	0150-0050	10	C:FXD CER DISC 1000 PF +80-20% 1000VDCW	56289	C0678102E102ZE19-CDH
A7R13	0683-1015	9	R:FXD COMP 100 OHM 5% 1/4W	01121	CB 1015
A7R14	0683-1015		R:FXD COMP 100 OHM 5% 1/4W	01121	CB 1015
A7R16	0683-1015		R:FXD COMP 100 OHM 5% 1/4W	01121	CB 1015
A7R18	0683-1015		R:FXD COMP 100 OHM 5% 1/4W	01121	CB 1015
A7R21	0683-1015		R:FXD COMP 100 OHM 5% 1/4W	01121	CB 1015
A7R23	0683-1015		R:FXD COMP 100 OHM 5% 1/4W	01121	CB 1015
A7R25-A7R36	1810-0047	88	RESISTOR NETWORK	28480	1810-0047
A7U11	1820-0485	31	INTEGRATED CIRCUIT: CTL	07263	SL15961
A7U13	1820-0486	99	INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U14	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U15	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U21	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A7U23	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U24	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U25	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U31	1820-0482	11	INTEGRATED CIRCUIT: CTL	07263	SL15964
A7U32	1820-0488	114	INTEGRATED CIRCUIT: CTL	07263	SL15965
A7U33	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U34	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U35	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U41	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A7U42	1820-0967	10	IC:CTL DUAL RANK J-K FLIP-FLOP	07263	SL3464
A7U43	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U44	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U45	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A7U51	1820-0186	22	INTERGRATED CIRCUIT:	28480	1820-0186
A7U52	1820-0967		IC:CTL DUAL RANK J-K FLIP-FLOP	07263	SL3464
A7U53	1820-0250	2	INTEGRATED CIRCUIT:TTL 6 BIT COMP	28480	1820-0250
A7U54	1820-0250		INTEGRATED CIRCUIT:TTL 6 BIT COMP	28480	1820-0250
A7U55	1820-0141	4	IC:TTL QUAD	28480	1820-0141
A7U61	1820-0187	24	INTEGRATED CIRCUIT: CTL	07263	SL15966
A7U62	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A7U63	1820-0141		IC:TTL QUAD	28480	1820-0141
A7U64	1820-0141		IC:TTL QUAD	28480	1820-0141
A7U65	1820-0141		IC:TTL QUAD	28480	1820-0141

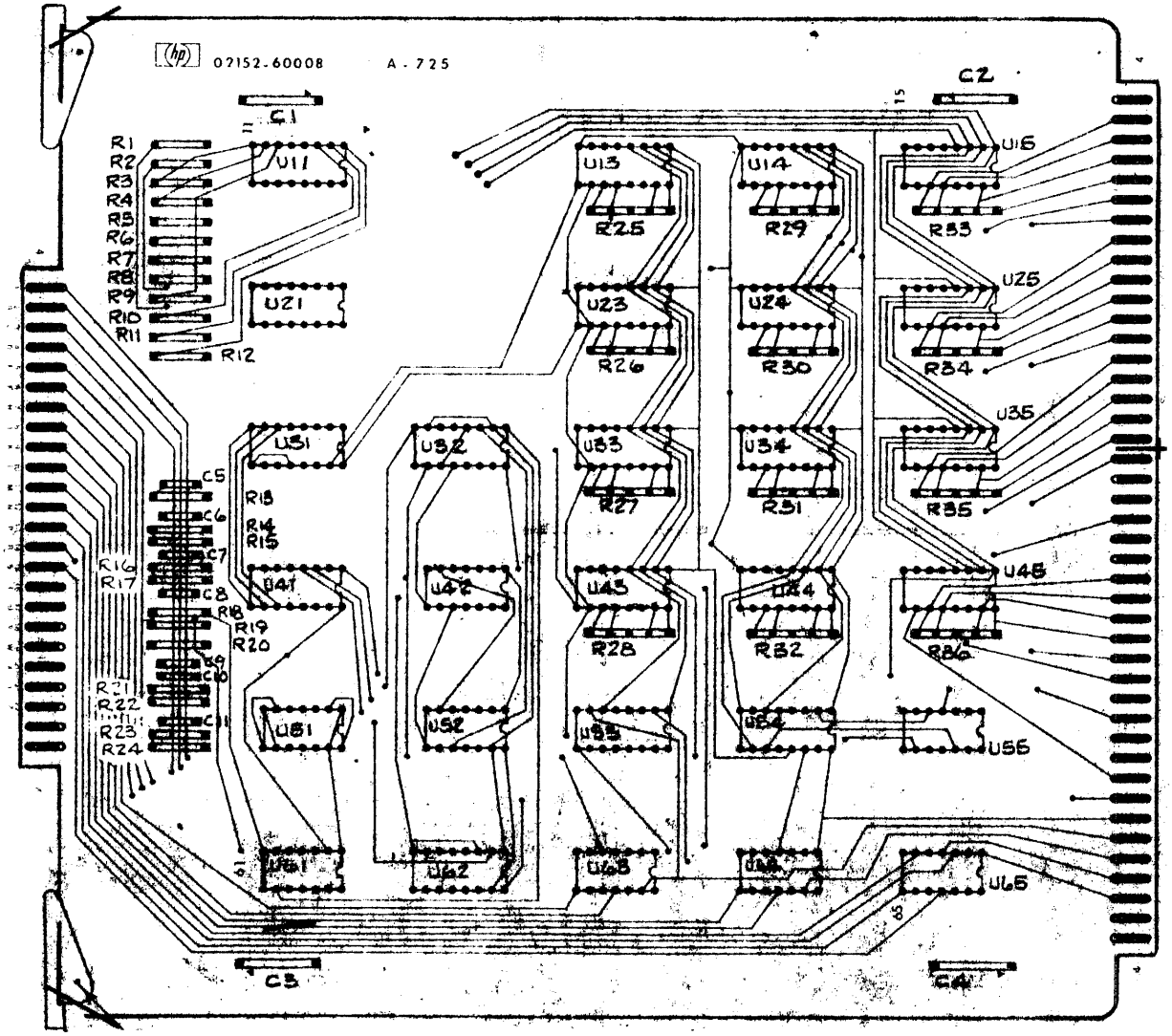
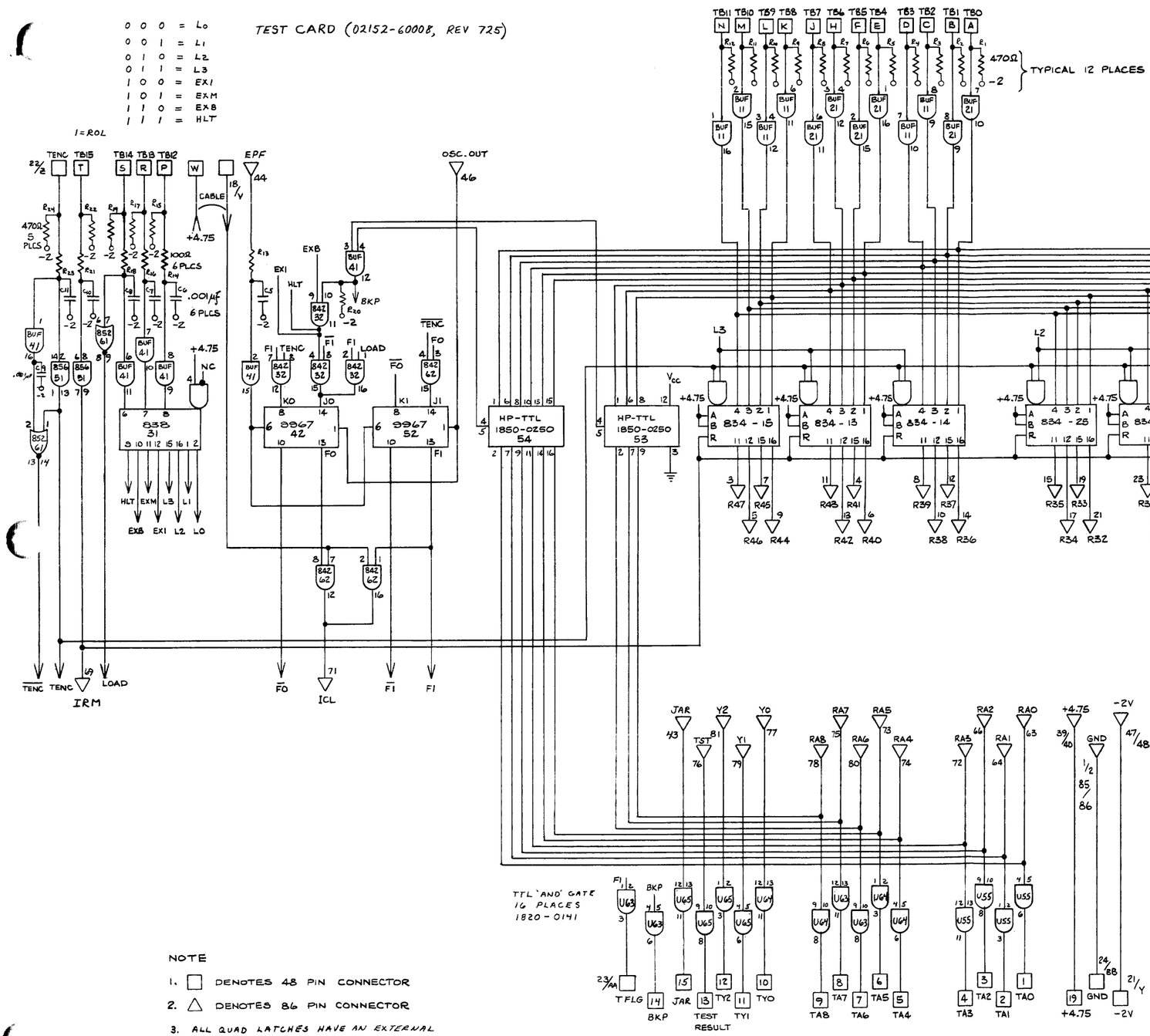


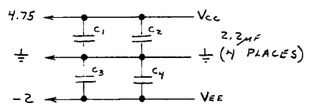
Figure 7-7. Test Card A7, Parts Location View

TEST CARD (02152-60008, REV 725)

0	0	0 = L0
0	0	1 = L1
0	1	0 = L2
0	1	1 = L3
1	0	0 = EX1
1	0	1 = EXM
1	1	0 = EX8
1	1	1 = HLT

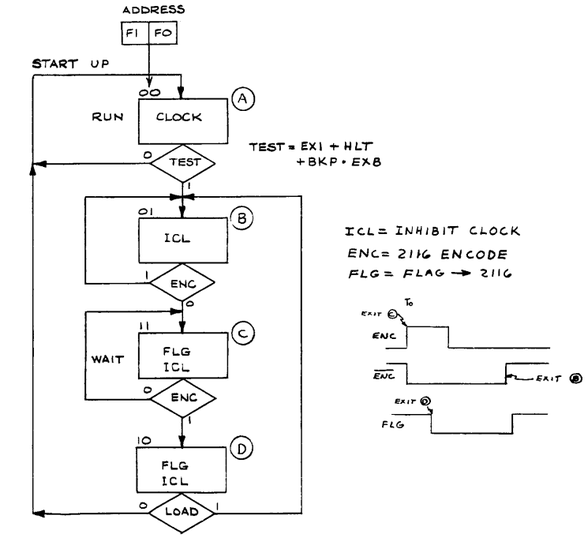
1 = ROL





NOTE

1. □ DENOTES 48 PIN CONNECTOR
2. △ DENOTES 86 PIN CONNECTOR
3. ALL QUAD LATCHES HAVE AN EXTERNAL PULL DOWN RESISTOR ADJACENT TO OUTPUT



FLOW CHART FOR TEST BOARD CONTROLLER

Figure 7-8. Test Card A7, Logic Diagram

Table 7-5. ROM Address Card A8, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A8	02152-60005		ROM ADDRESS CARD	28480	02152-60005
A8C1-					
A8C4	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	1500225X9020A2-DYS
A8Q1	1853-0015	10	TSTR:SI PNP	80131	2N3640
A8Q2	1853-0015		TSTR:SI PNP	80131	2N3640
A8R1					
A8R9	0683-3315	42	R:FXD COMP 330 OHM 5% 1/4W	01121	CB 3315
A8R10-					
A8R38	1810-0047		RESISTOR NETWORK	28480	1810-0047
A8U11	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A8U12	1820-0953	28	IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A8U14	1820-0305	3	INTEGRATED CIRCUIT:BINARY FULL ADDER	01295	SN7483N
A8U15	1820-0305		INTEGRATED CIRCUIT:BINARY FULL ADDER	01295	SN7483N
A8U16	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A8U17	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U21	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U22	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U23	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U25	1820-0305		INTEGRATED CIRCUIT:BINARY FULL ADDER	01295	SN7483N
A8U26	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U27	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U32	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U33	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U34	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A8U35	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A8U36	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A8U36	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U37	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U41	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U42	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
A8U43	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U44	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U45	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U46	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U47	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U51	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U52	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A8U53	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A8U54	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A8U55	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
A8U56	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U57	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U61	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A8U62	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U63	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U64	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U65	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U66	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
A8U67	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A8U72	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A8U73	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A8U74	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U75	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U76	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U77	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U81	1820-0483		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U82	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A8U83	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A8U84	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U85	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U86	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U87	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U91	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A8U92	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U93	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U94	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U95	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A8U96	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A8U97	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963

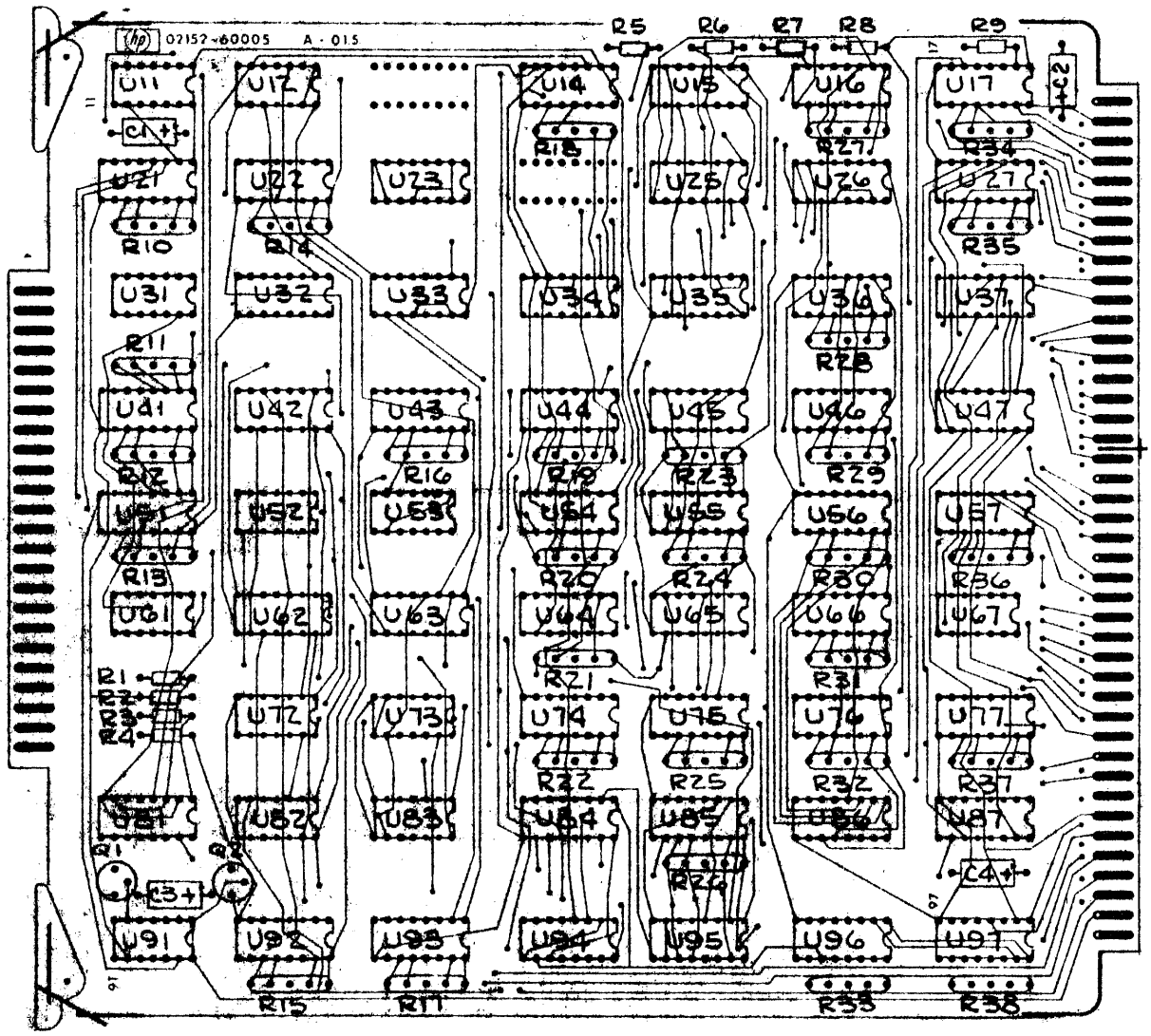


Figure 7-9. ROM Address Card A8, Parts Location View

ROM ADDRESS CARD (02152-60005, REV. 015)

P/O INSTRUCTION REGISTER (BC)

P/O CONSTANT CALL

CURRENT PAGE STORAGE

JSB CONTROL

CONDITIONAL BRANCHING

P/O INSTRUCTION REGISTER (TS)

TEST DECODER

P/O INSTRUCTION REGISTER (BP)

P/O INSTRUCTION REGISTER (BL)

Figure 7-10. ROM Address Card A8, Logic Diagram

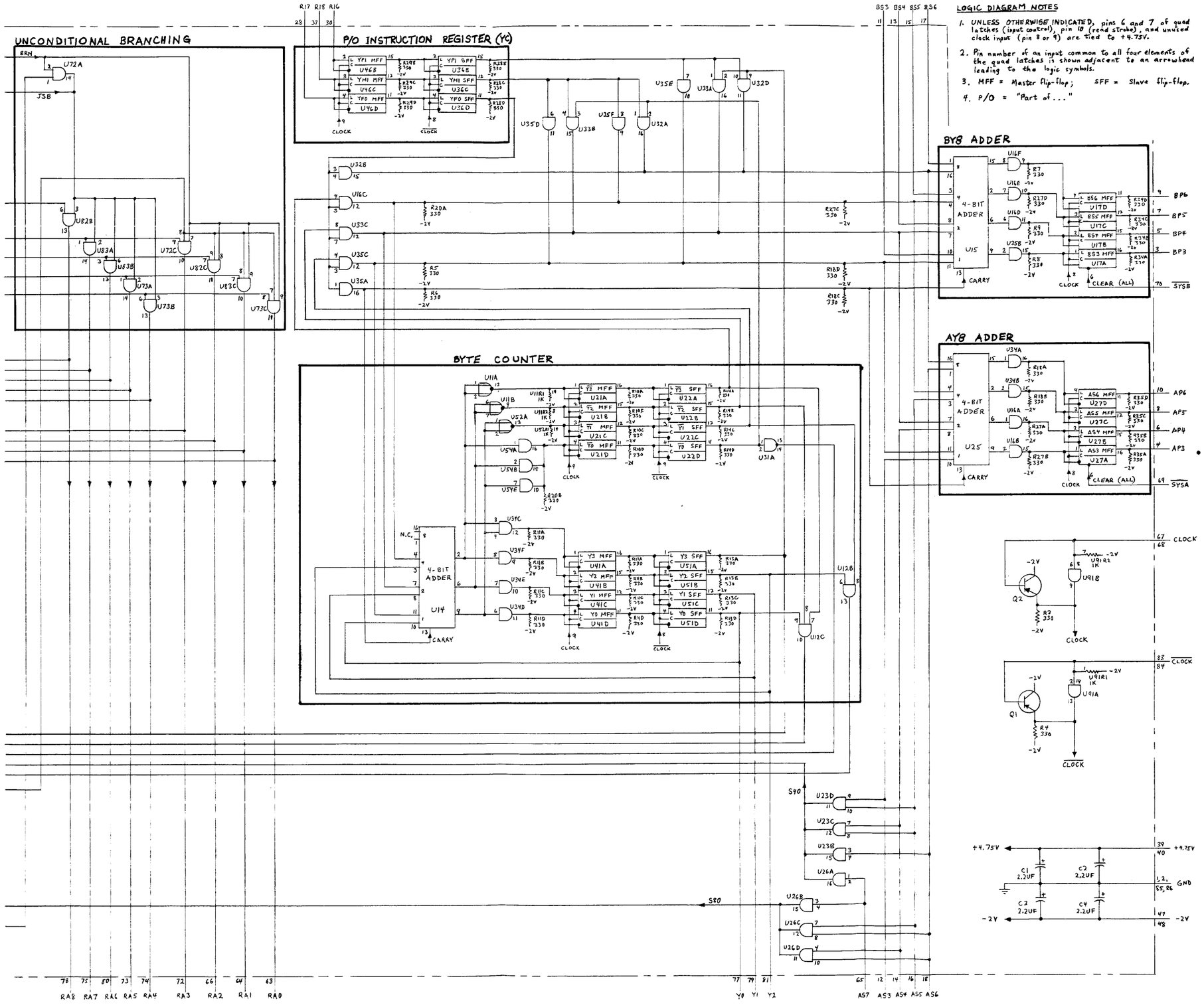


Figure 7-10. ROM Address Card A8, Logic Diagram (Cont'd)

Table 7-6. Read-Only Memory Card A9, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A9	02152-60007		ROM CARD	28480	02152-60007
A9C1 -					
A9C4	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
A9R1	0698-5615	2	R:FXD FLM 3.32K 0.5% 1/4W	28480	0698-5615
A9R10	0698-5615		R:FXD FLM 3.32K 0.5% 1/4W	28480	0698-5615
A9U13	94-19-144	1	INTEGRATED CIRCUIT	28480	94-19-144
A9U15	82-19-144	1	INTEGRATED CIRCUIT	28480	82-19-144
A9U23	93-19-144	1	INTEGRATED CIRCUIT	28480	93-19-144
A9U25	81-19-144	1	INTEGRATED CIRCUIT	28480	81-19-144
A9U35	136-22-144	1	INTEGRATED CIRCUIT	28480	136-22-144
A9U43	139-22-142	1	INTEGRATED CIRCUIT	28480	139-22-142
A9U45	135-22-144	1	INTEGRATED CIRCUIT	28480	135-22-144
A9U53	90-19-144	1	INTEGRATED CIRCUIT	28480	90-19-144
A9U55	78-19-144	1	INTEGRATED CIRCUIT	28480	78-19-144
A9U63	138-22-144	1	INTEGRATED CIRCUIT	28480	138-22-144
A9U65	77-19-144	1	INTEGRATED CIRCUIT	28480	77-19-144
A9U73	88-09-144	1	INTEGRATED CIRCUIT	28480	88-09-144
A9U75	76-19-144	1	INTEGRATED CIRCUIT	28480	76-19-144
A9U83	87-19-144	1	INTEGRATED CIRCUIT	28480	87-19-144
A9U85	75-19-144	1	INTEGRATED CIRCUIT	28480	75-19-144
A9U95	134-22-144	1	INTEGRATED CIRCUIT	28480	134-22-144
A9U95	137-22-143	1	INTEGRATED CIRCUIT	28480	137-22-143
A9U97	1820-0144	3	INTEGRATED CIRCUIT:LEVEL TRANSLATOR	04713	MC1018P
A9U103	85-19-144	1	INTEGRATED CIRCUIT	28480	85-19-144
A9U105	133-22-144	1	INTEGRATED CIRCUIT	28480	133-22-144
A9U107	1820-0144		INTEGRATED CIRCUIT:LEVEL TRANSLATOR	04713	MC1018P
A9U113	84-19-144	1	INTEGRATED CIRCUIT	28480	84-19-144
A9U115	72-19-144	1	INTEGRATED CIRCUIT	28480	72-19-144
A9U125	71-19-144	1	INTEGRATED CIRCUIT	28480	71-19-144
A9U126	1820-0144		INTEGRATED CIRCUIT:LEVEL TRANSLATOR	04713	MC1018P

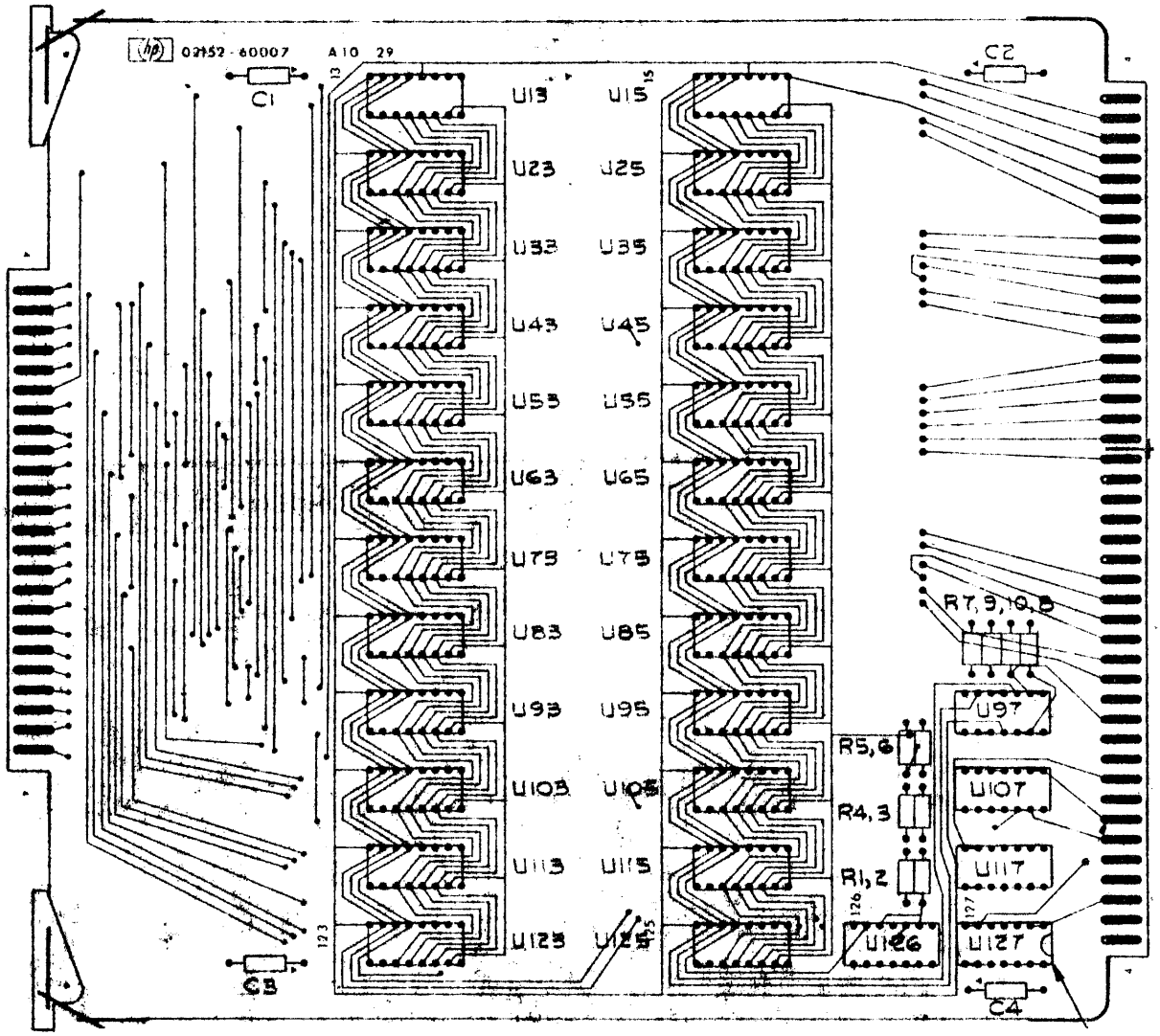
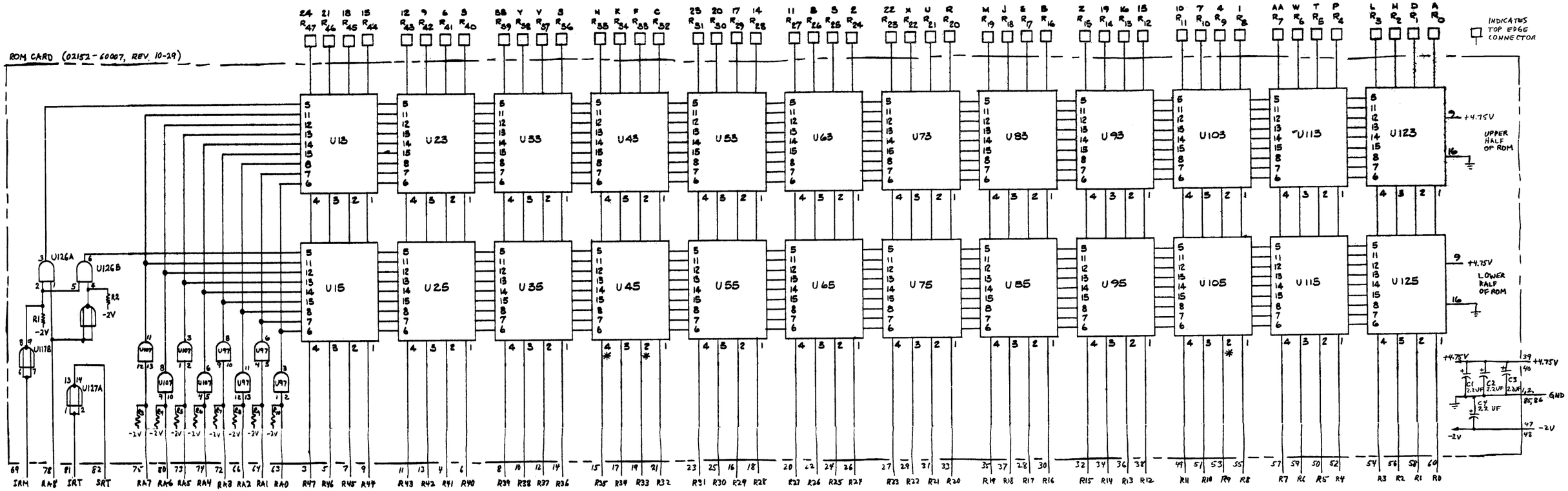


Figure 7-11. Read-Only Memory Card A9, Parts Location View



NOTES

1. * INDICATES PIN NOT CONNECTED ON THIS MICROCIRCUIT PACKAGE.
2. ALL RESISTORS ARE 560 OHMS.
3. ROM PC WIRING SHOWN IN SIMPLIFIED FORM ABOVE. ACTUAL WIRING OF EACH PACKAGE IS AS SHOWN BELOW.

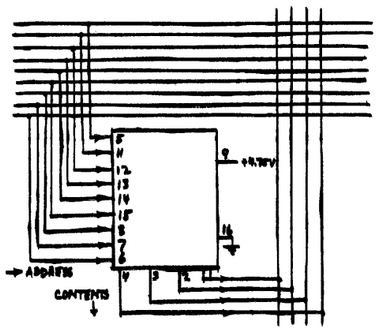


Figure 7-12. Read-Only Memory Card A9, Logic Diagram

Table 7-7. D-Register Card A10, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A10	02152-60004		D REGISTER CARD	28480	02152-60004
A10C1	0140-0192	3	C:FXD MICA 63 PF 5%	28480	0140-0192
A10C4	0160-2588	1	C:FXD CER 1000PF 5% 50VDCW	28480	0160-2588
A10C6 - A10C9	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
A10L1	9140-0105	1	COIL:MOLDED CHOKE 8.20 UH 10%	28480	9140-0105
A10Q1 - A10Q3	1854-0005	3	TSTR:SI NPN	80131	2N708
A10Q4 - A10Q7	1853-0015		TSTR:SI PNP	80131	2N3640
A10R5	0683-1025	1	R:FXD COMP 1000 OHM 5% 1/4W	01121	CB 1025
A10R6 - A10R9	1810-0047		RESISTOR NETWORK	28480	1810-0047
A10R10 - A10R12	0683-3315		R:FXD COMP 330 OHM 5% 1/4W	01121	CB 3315
A10R13					
A10R22	1810-0047		RESISTOR NETWORK	28480	1810-0047
A10R23 - A10R25	0683-3315		R:FXD COMP 330 OHM 5% 1/4W	01121	CB 3315
A10R26					
A10R33	1810-0047		RESISTOR NETWORK	28480	1810-0047
A10R34	0683-3315		R:FXD COMP 330 OHM 5% 1/4W	01121	CB 3315
A10R35	0683-3315		R:FXD COMP 330 OHM 5% 1/4W	01121	CB 3315
A10U11	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U14	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U15	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U21	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U24	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U25	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U34	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U35	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U44	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U45	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U51	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U53	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A10U54	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A10U55	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U61	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U62	1820-0967		IC:CTL DUAL RANK J-K FLIP-FLOP	07263	SL3464
A10U63	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U64	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U65	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U72	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A10U73	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A10U74	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U75	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U81	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U83	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U84	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U85	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A10U92	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A10U93	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A10U94	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A10U95	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A10Y1	0410-0035		CRYSTAL:QUARTZ 10MC/S 0.005%	28480	0410-0035

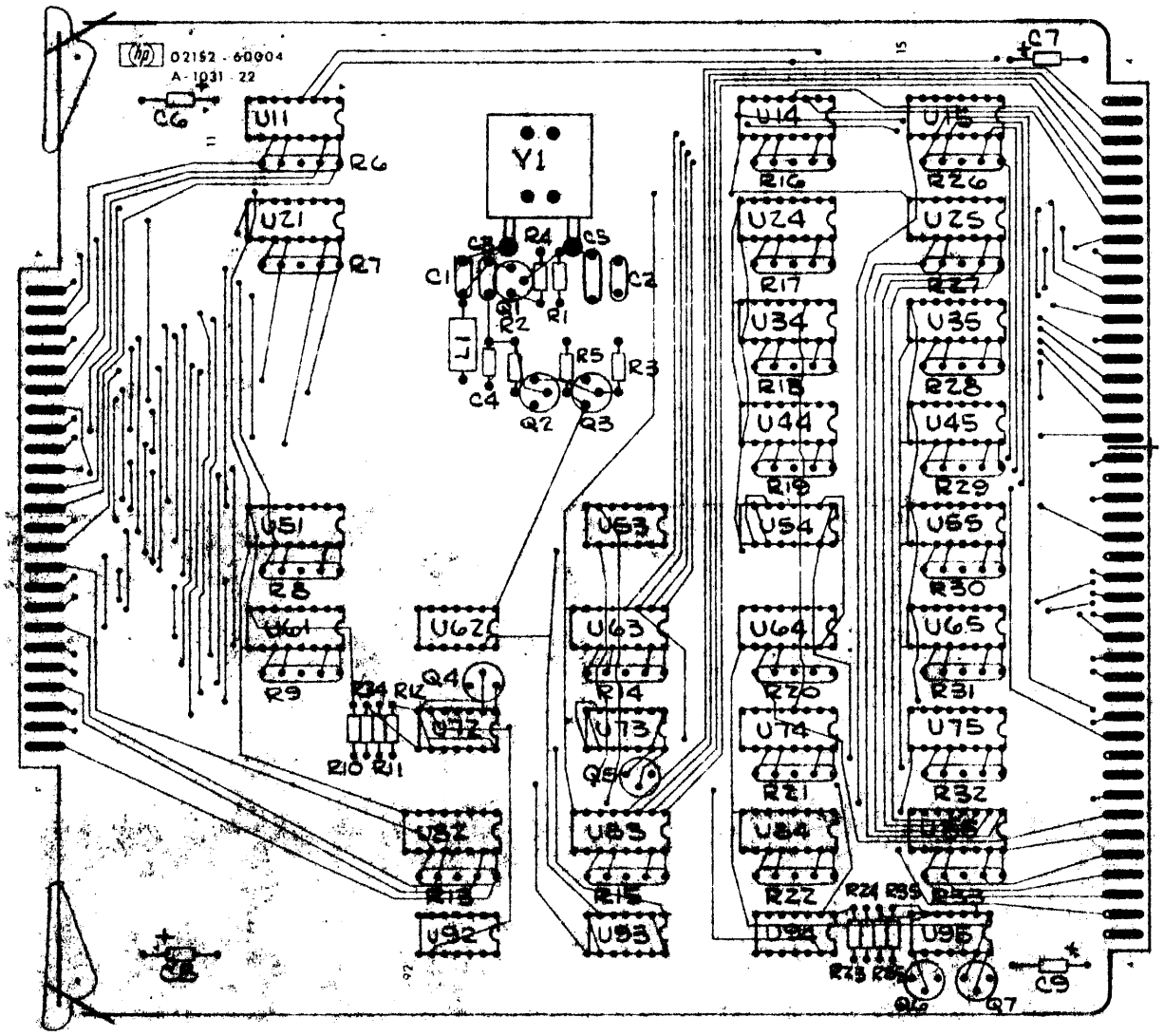


Figure 7-13. D-Register Card A10, Parts Location View

D-REGISTER CARD (02152-60004, REV 1031)

10 MHz OSCILLATOR
DESIGN TAKEN FROM 2116
TIMING GENERATOR CARD
(02116-6281)

NOTE:
1. ALL RESISTORS ARE 330Ω ¼ WATT 5% EXCEPT WHERE INDICATED
2. ALL QUAD LATCHES HAVE AN EXTERNAL PULL DOWN RESISTOR ADJACENT TO OUTPUT

Figure 7-14. D-Register Card A10, Logic Diagram

Table 7-8. D-Shifter Card A11, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A11	02152-60001	3	SHIFTER CARD	28480	02152-60001
A11C1-A11C4	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
A11R1-A11R6	0683-4715	26	R:FXD COMP 470 OHM 5% 1/4W	01121	CB 4715
A11U12	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A11U13	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A11U15	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U16	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U17	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U21					
A11U25	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U26	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U31					
A11U35	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U36	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U37	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U41					
A11U45	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U46	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U47	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U51					
A11U55	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U56	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U57	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U61					
A11U65	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U66	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U67	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U71					
A11U75	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U76	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U77	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U81					
A11U85	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U86	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U87	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U91	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U92	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U93	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
A11U94	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U95	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U96	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U97	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U103	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A11U104	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
			A13 SAME AS A11		
			A15 SAME AS A11		

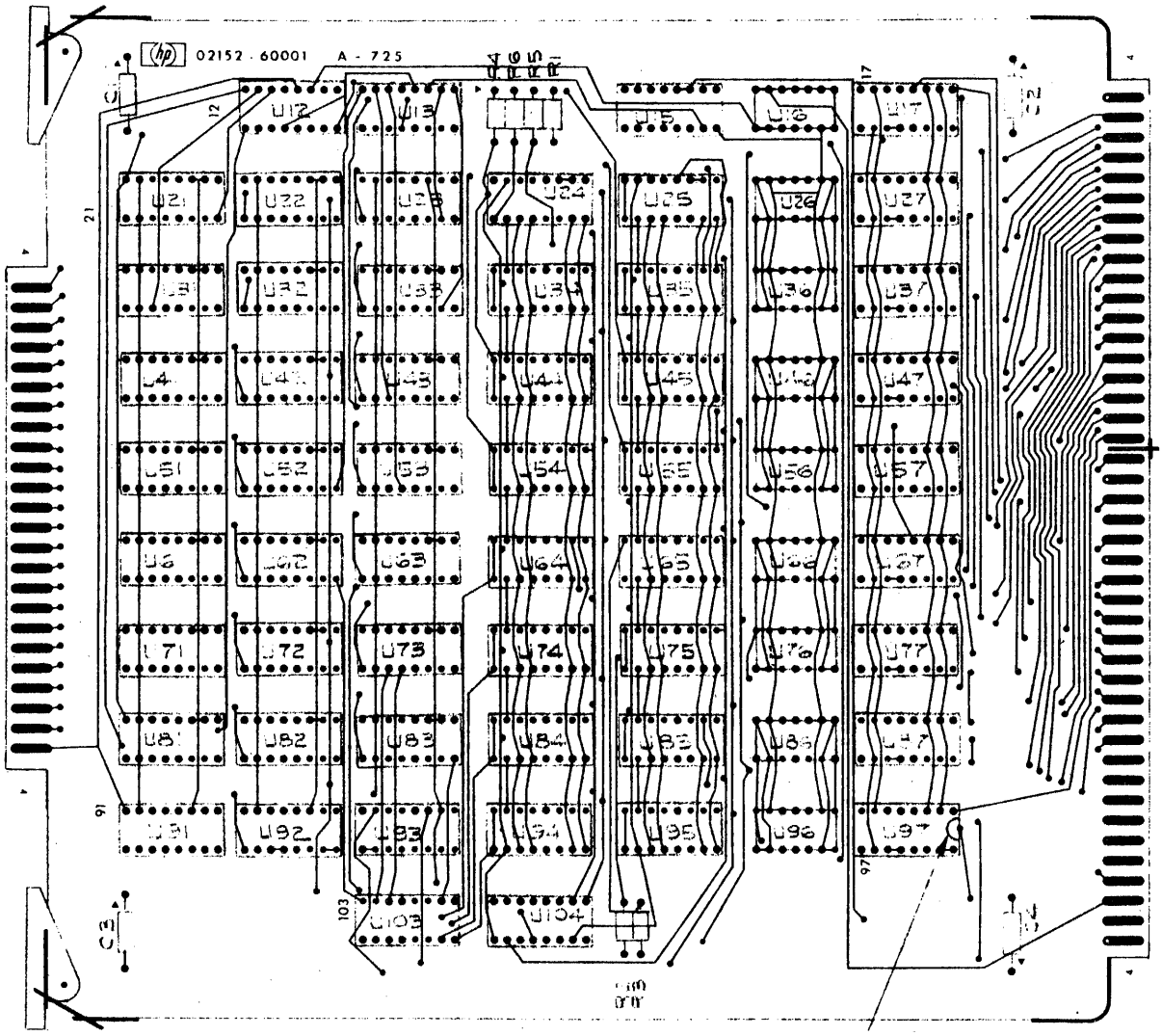


Figure 7-15. D-Shifter Card A11, Parts Location View

SHIFTER BOARD (02152-60001, REV. 725)

INDICATES TOP EDGE CONNECTOR

Figure 7-16. D-Shifter Card A11, Logic Diagram

Table 7-9. C-Adder Card A12, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A12	02152-60002	3	ADDER CARD	28480	02152-60002
A12C1 - A12C4	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
A12Q1	1853-0015		TSTR:SI PNP	80131	2N3640
A12Q2	1853-0015		TSTR:SI PNP	80131	2N3640
A12R1 - A12R12	0683-3315		R:FXD COMP 330 OHM 5% 1/4W	01121	CB 3315
A12R13 - A12R31	1810-0047		RESISTOR NETWORK	28480	1810-0047
A12U3	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U4	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U5	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U6	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U11	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U12	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U14	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U16	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U21	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U22	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U23	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U24	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U25	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U26	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U31	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U32	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U33	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U34	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U35	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U36	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U41	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U42	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U43	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U44	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U45	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U46	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U51	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U52	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U53	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U54	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U55	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U56	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U61	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U62	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U63	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U64	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U65	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U66	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U71	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U72	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U73	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U74	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U75	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U76	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U81	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U82	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U83	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U84	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U85	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U86	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U91	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U92	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U93	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U94	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U95	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U96	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U101	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U102	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U103	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U104	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U105	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A12U106	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U111	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U113	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U113	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U114	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U115	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U116	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U121	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U122	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U123	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U124	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U125	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A12U126	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965

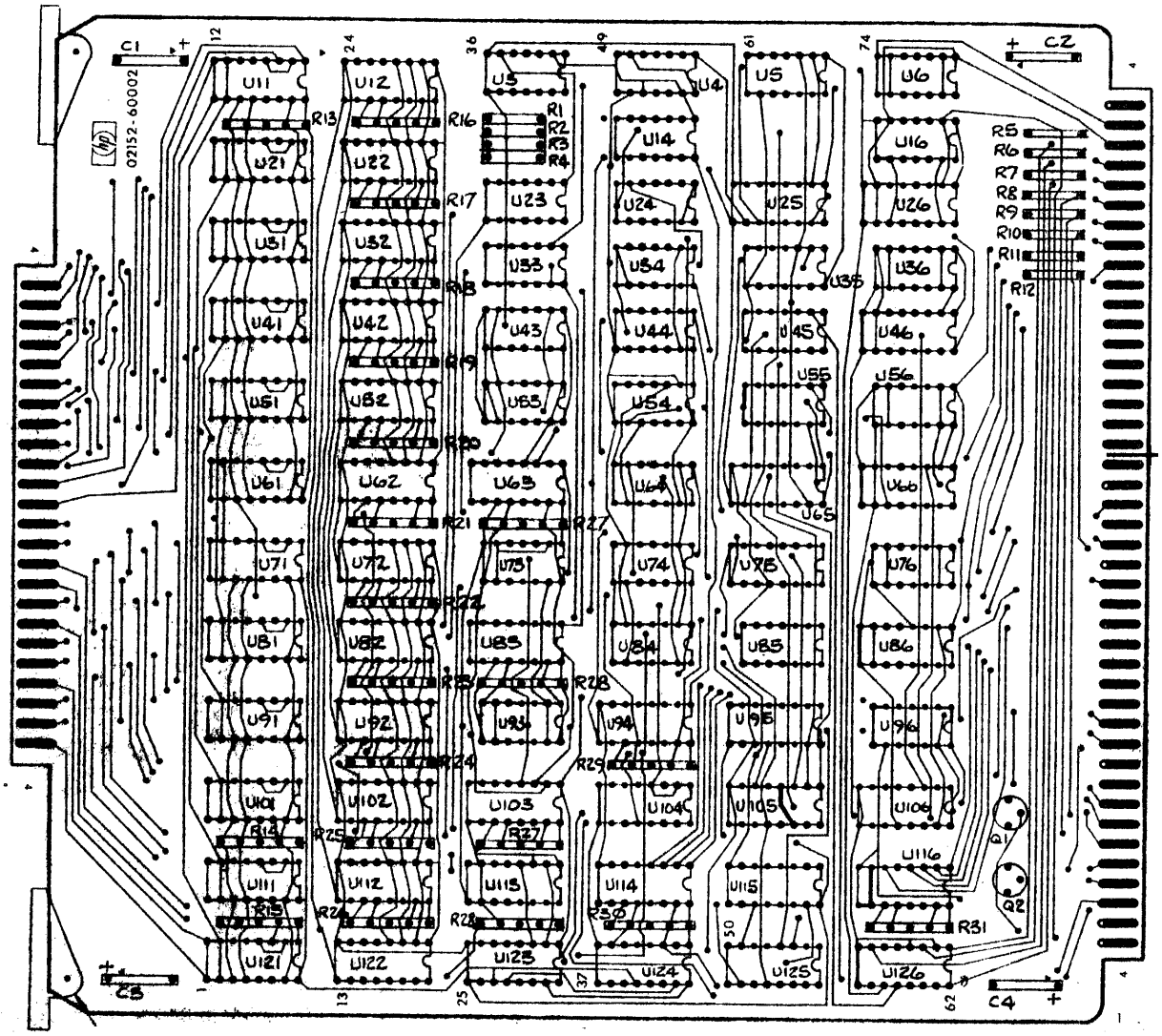


Figure 7-17. C-Adder Card A12, Parts Location View

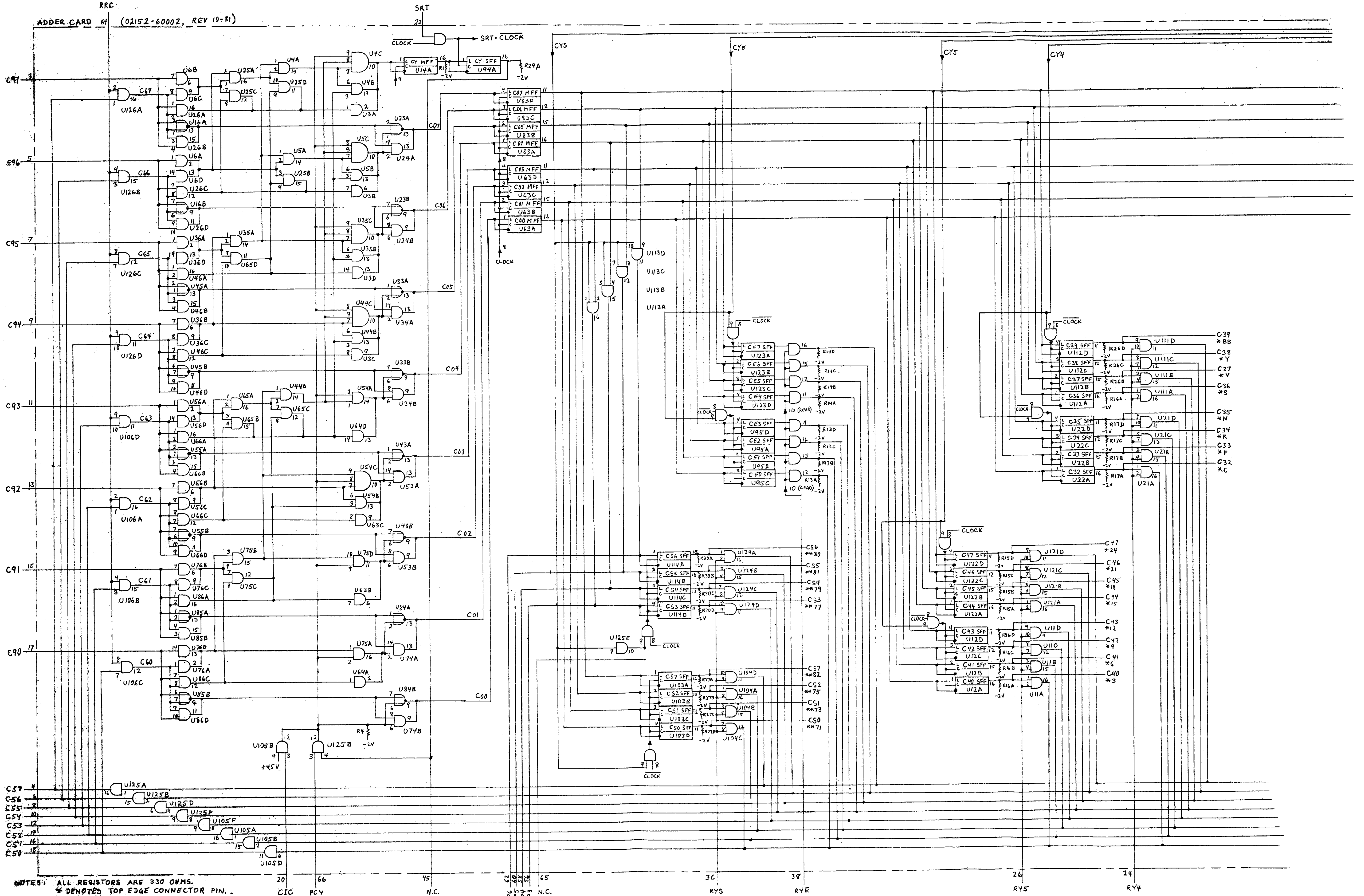


Figure 7-18. C-Adder Card A12, Logic Diagram

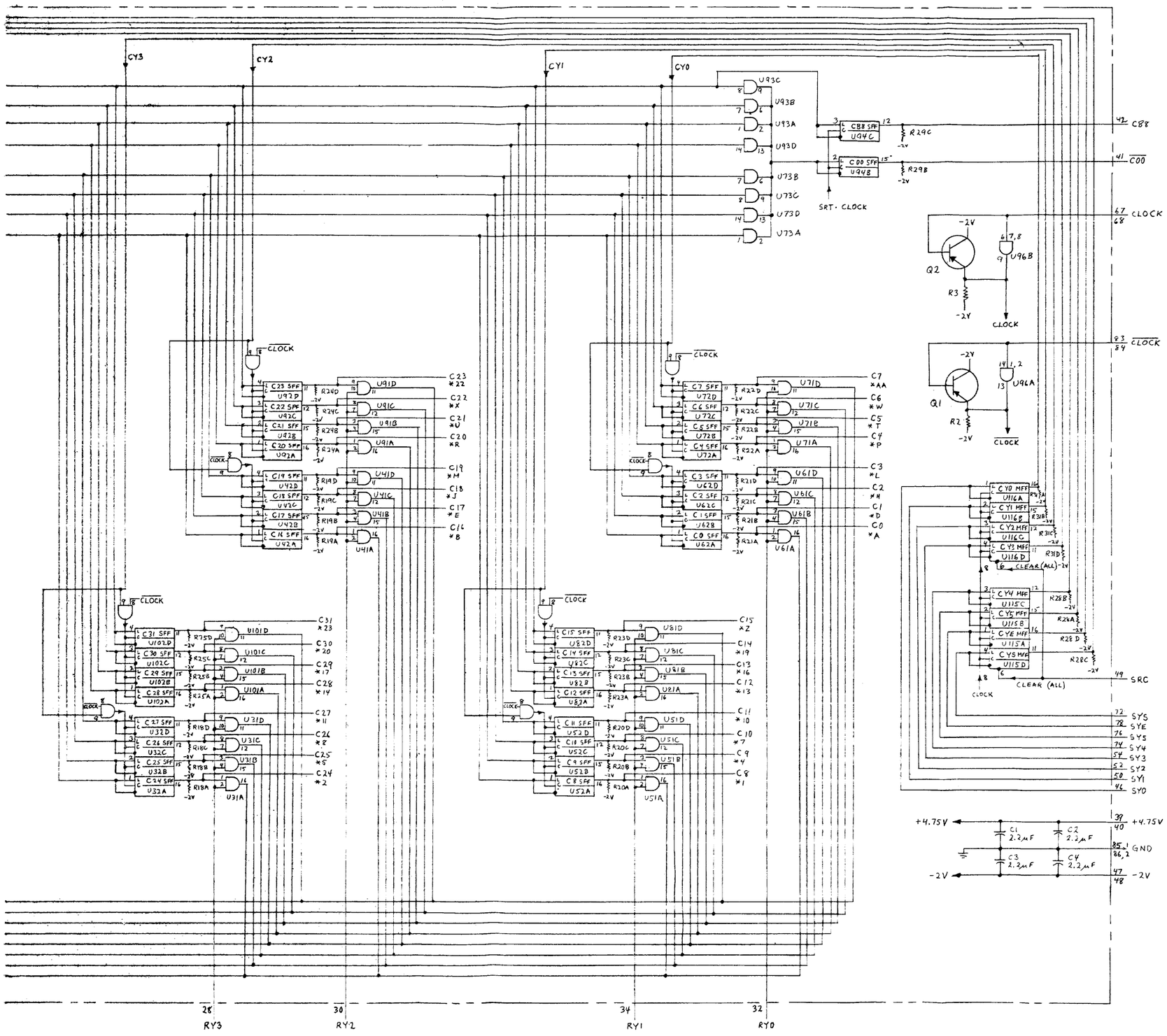


Figure 7-18. C-Adder Card A12, Logic Diagram (Cont'd)

Table 7-10. B-Shifter Card A13, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A11	02152-60001	3	SHIFTER CARD	28480	02152-60001
A11C1-					
A11C4	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
A11R1-					
A11R6	0683-4715	26	R:FXD COMP 470 OHM 5% 1/4W	01121	CB 4715
A11U12	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A11U13	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A11U15	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U16	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U17	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U21					
A11U25	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U26	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U31					
A11U35	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U36	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U37	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U41					
A11U45	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U46	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U47	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U51					
A11U55	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U56	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U57	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U61					
A11U65	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U66	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U67	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U71					
A11U75	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U76	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U77	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U81					
A11U85	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U86	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U87	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U91	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U92	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U93	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
A11U94	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U95	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U96	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U97	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U103	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A11U104	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
			A13 SAME AS A11		
			A15 SAME AS A11		

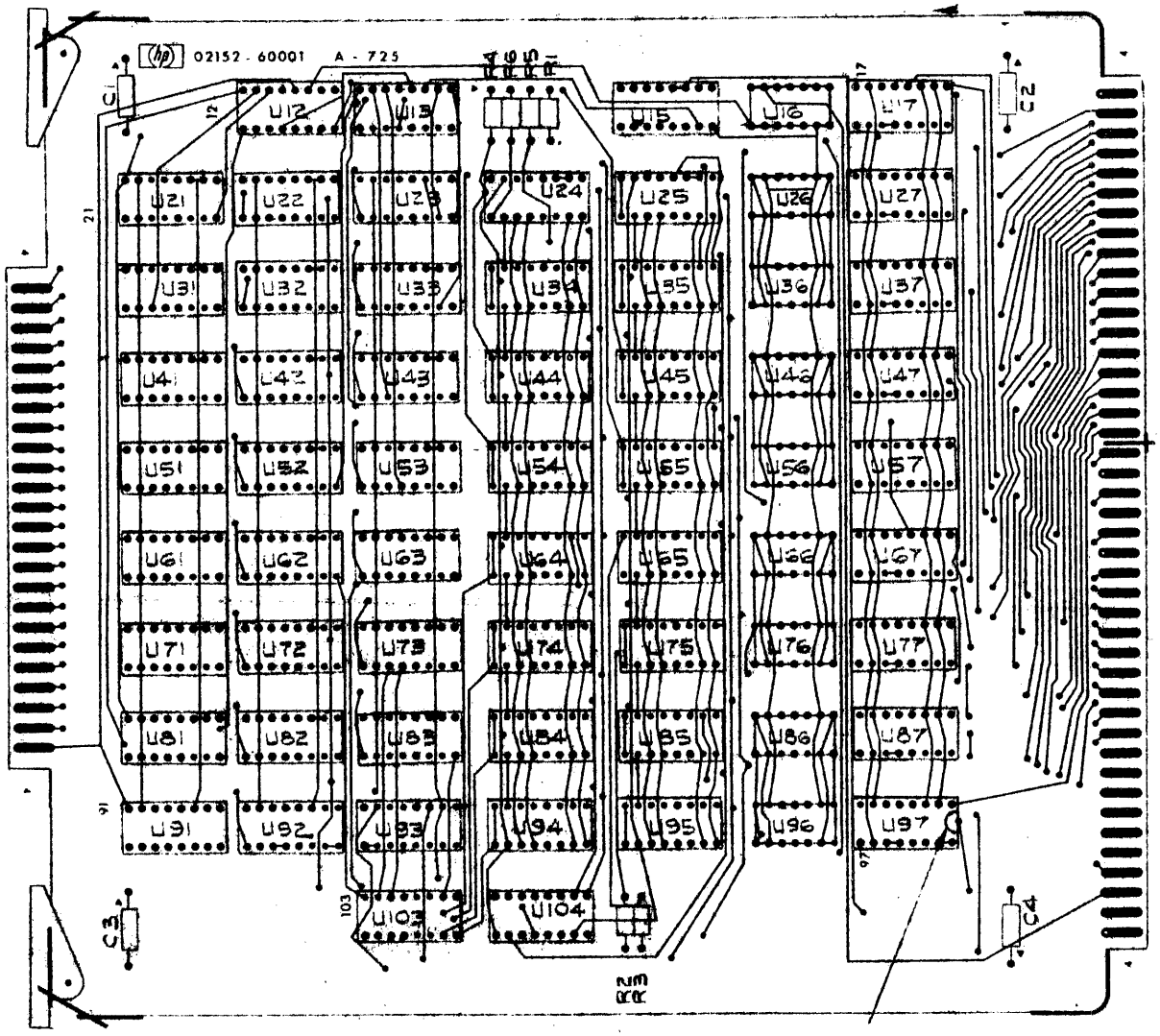


Figure 7-19. B-Shifter Card A13, Parts Location View

SHIFTER BOARD (02152-60001, REV 725)

INDICATES TOP EDGE CONNECTOR

B47 B39 B31 B23 B15 B7 B46 B38 B30 B22 B14 B6 B45 B37 B29 B21 B13 B5 B44 B36 B28 B20 B12 B4 B43 B35 B27 B19 B11 B3 B42 B34 B26 B18 B10 B2 B41 B33 B25 B17 B9 B1 B40 B32 B24 B16 B8 B0

+4.75V C1 2.2uf C2 2.2uf C3 2.2uf C4 2.2uf -2V GND

BS7 BS6 BS5 BS4 BS3 BS2 BS1 BS0 TSB

1 OF 8 DECODER U93

1 OF 8 DECODER U104

SW7 SW6 SW5 SW4 SW3 SW2 SW1 SW0

SV7 SV6 SV5 SV4 SV3 SV2 SV1 SV0

R1 470 R2 470 R3 470 R4 470 R5 470 R6 470

B70 B71 B72 B73 B74 B75 B76 B77

CPB N.C. TMI TSA TM2 N.C. TM3 TSB TM4

B97 B77 A57 A77 C57 B96 B76 A56 A76 C56 B95 B75 A55 A75 C55 B94 B74 A54 A74 C54 B93 B73 A63 A73 C53 B92 B72 A52 A72 C52 B91 B71 A51 A71 C51 B90 B70 A50 A70 C50

Figure 7-20. B-Shifter Card A13, Logic Diagram

Table 7-11. B-Adder Card A14, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A12	02152-60002	3	ADDER CARD	28480	02152-60002
A12C1 -					
A12C4	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
A12Q1	1853-0015		TSTR:SI PNP	80131	2N3640
A12Q2	1853-0015		TSTR:SI PNP	80131	2N3640
A12R1 -					
A12R12	0683-3315		R:FXD COMP 330 OHM 5% 1/4W	01121	CB 3315
A12R13 -					
A12R31	1810-0047		RESISTOR NETWORK	28480	1810-0047
A12U3	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U4	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U5	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U6	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U11	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U12	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U14	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U16	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U21	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U22	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U23	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U24	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U25	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U26	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U31	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U32	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U33	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U34	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U35	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U36	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U41	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U42	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U43	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U44	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U45	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U46	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U51	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U52	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U53	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U54	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U55	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U56	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U61	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U62	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U63	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U64	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U65	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U66	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U71	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U72	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U73	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U74	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U75	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U76	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U81	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U82	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U83	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U84	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U85	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U86	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U91	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U92	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U93	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U94	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U95	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U96	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U101	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U102	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U103	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U104	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U105	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A12U106	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U111	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U113	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U113	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U114	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U115	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U116	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U121	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U122	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U123	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U124	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U125	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A12U126	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965

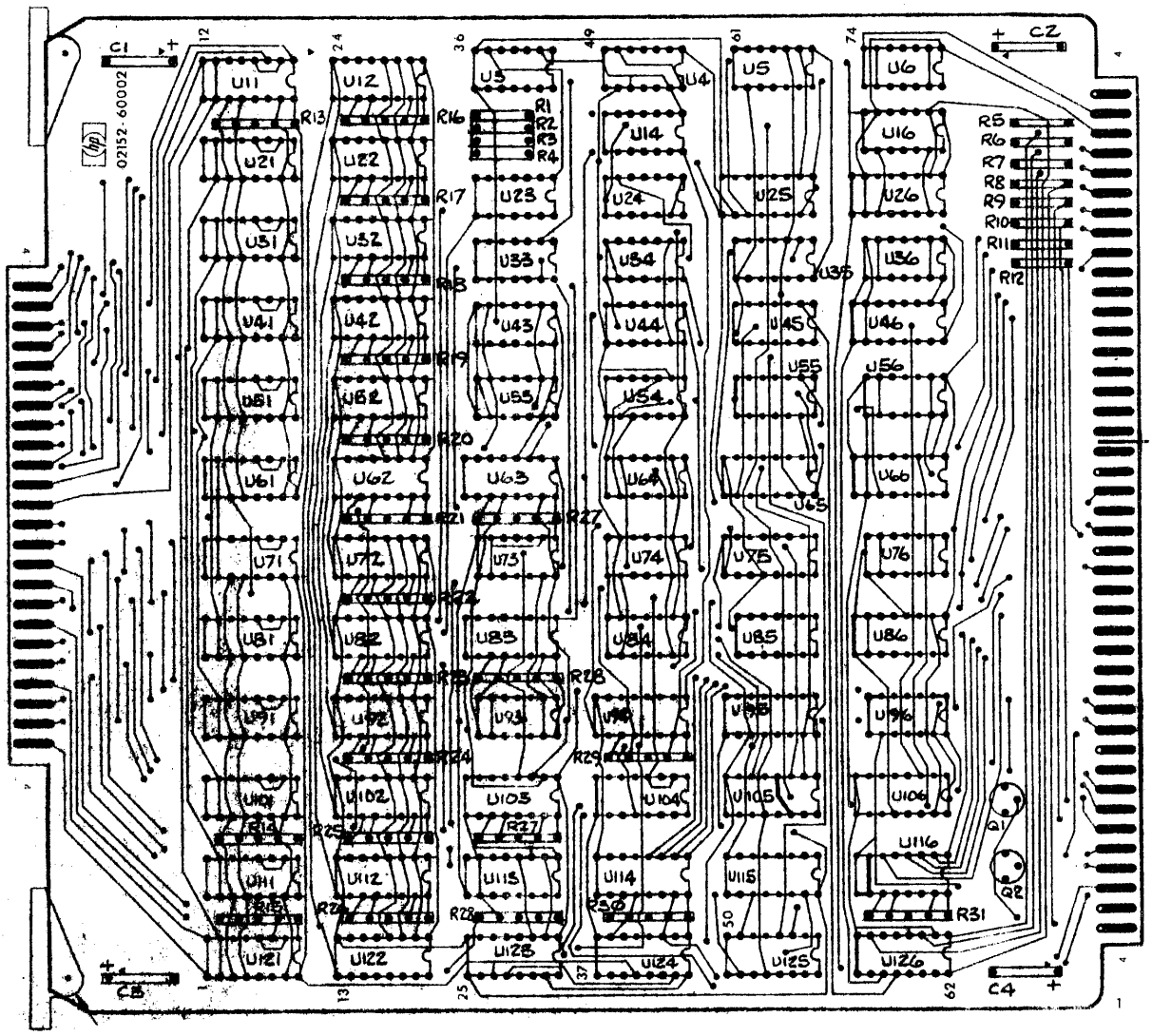


Figure 7-21. B-Adder Card A14, Parts Location View

ADDER CARD 64 (02152-60002, REV 10-31)

NOTES: ALL REGISTORS ARE 330 OHMS.
* DENOTES TOP EDGE CONNECTOR PIN.
**DENOTES BACKPLANE CONNECTOR PIN.

Figure 7-22. B-Adder Card A14, Logic Diagram

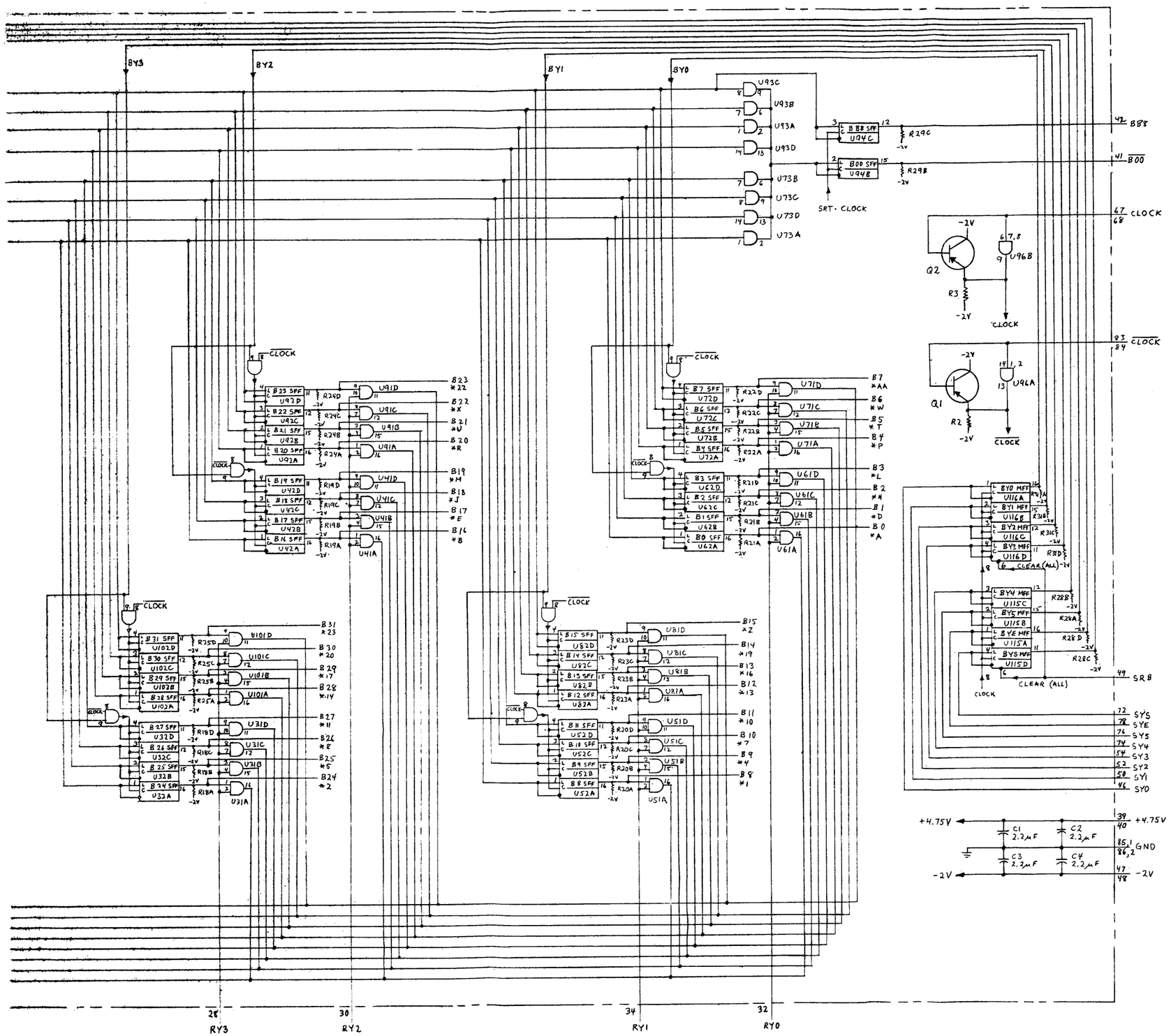


Figure 7-22. B-Adder Card A14, Logic Diagram (Cont'd)

Table 7-12. A-Shifter Card A15, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A11	02152-60001	3	SHIFTER CARD	28480	02152-60001
A11C1-					
A11C4	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
A11R1-					
A11R6	0683-4715	26	R:FXD COMP 470 OHM 5% 1/4W	01121	CB 4715
A11U12	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A11U13	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A11U15	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U16	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U17	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U21					
A11U25	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U26	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U31					
A11U35	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U36	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U37	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U41					
A11U45	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U46	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U47	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U51					
A11U55	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U56	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U57	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U61					
A11U65	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U66	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U67	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U71					
A11U75	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U76	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U77	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U81					
A11U85	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U86	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A11U87	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U91	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U92	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U93	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
A11U94	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U95	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U96	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A11U97	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A11U103	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A11U104	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
			A13 SAME AS A11		
			A15 SAME AS A11		

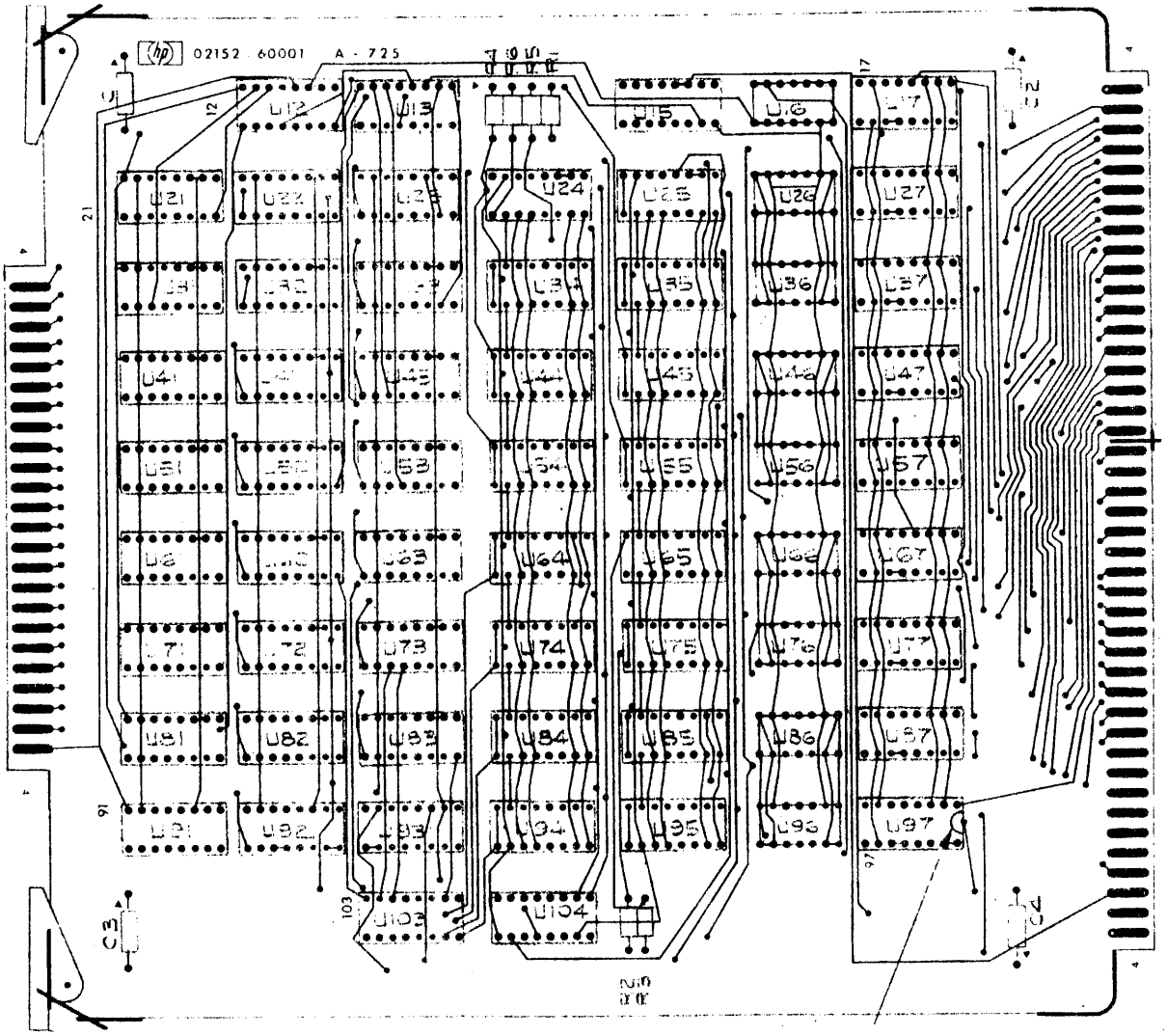


Figure 7-23. A-Shifter Card A15, Parts Location View

Figure 7-24. A-Shifter Card A15, Logic Diagram

Table 7-13. A-Adder Card A16, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A12	02152-60002	3	ADDER CARD	28480	02152-60002
A12C1 - A12C4	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
A12Q1	1853-0015		TSTR:SI PNP	80131	2N3640
A12Q2	1853-0015		TSTR:SI PNP	80131	2N3640
A12R1 - A12R12	0683-3315		R:FXD COMP 330 OHM 5% 1/4W	01121	CB 3315
A12R13 - A12R31	1810-0047		RESISTOR NETWORK	28480	1810-0047
A12U3	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U4	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U5	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U6	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U11	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U12	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U14	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U16	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U21	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U22	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U23	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U24	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U25	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U26	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U31	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U32	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U33	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U34	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U35	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U36	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U41	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U42	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U43	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U44	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U45	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U46	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U51	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U52	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U53	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U54	1820-0953		IC:CTL TRIPLE 2-2-3-INPUT AND GATE	07263	SL3456
A12U55	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U56	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U61	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U62	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U63	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U64	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U65	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U66	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U71	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U72	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U73	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U74	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U75	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U76	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U81	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U82	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U83	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U84	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U85	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A12U86	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U91	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U92	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U93	1820-0965		IC:CTL QUAD 2-INPUT AND GATE	07263	SL3462
A12U94	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U95	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U96	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A12U101	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U102	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U103	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U104	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U105	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A12U106	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U111	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U113	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U113	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U114	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U115	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U116	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U121	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U122	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U123	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A12U124	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A12U125	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A12U126	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965

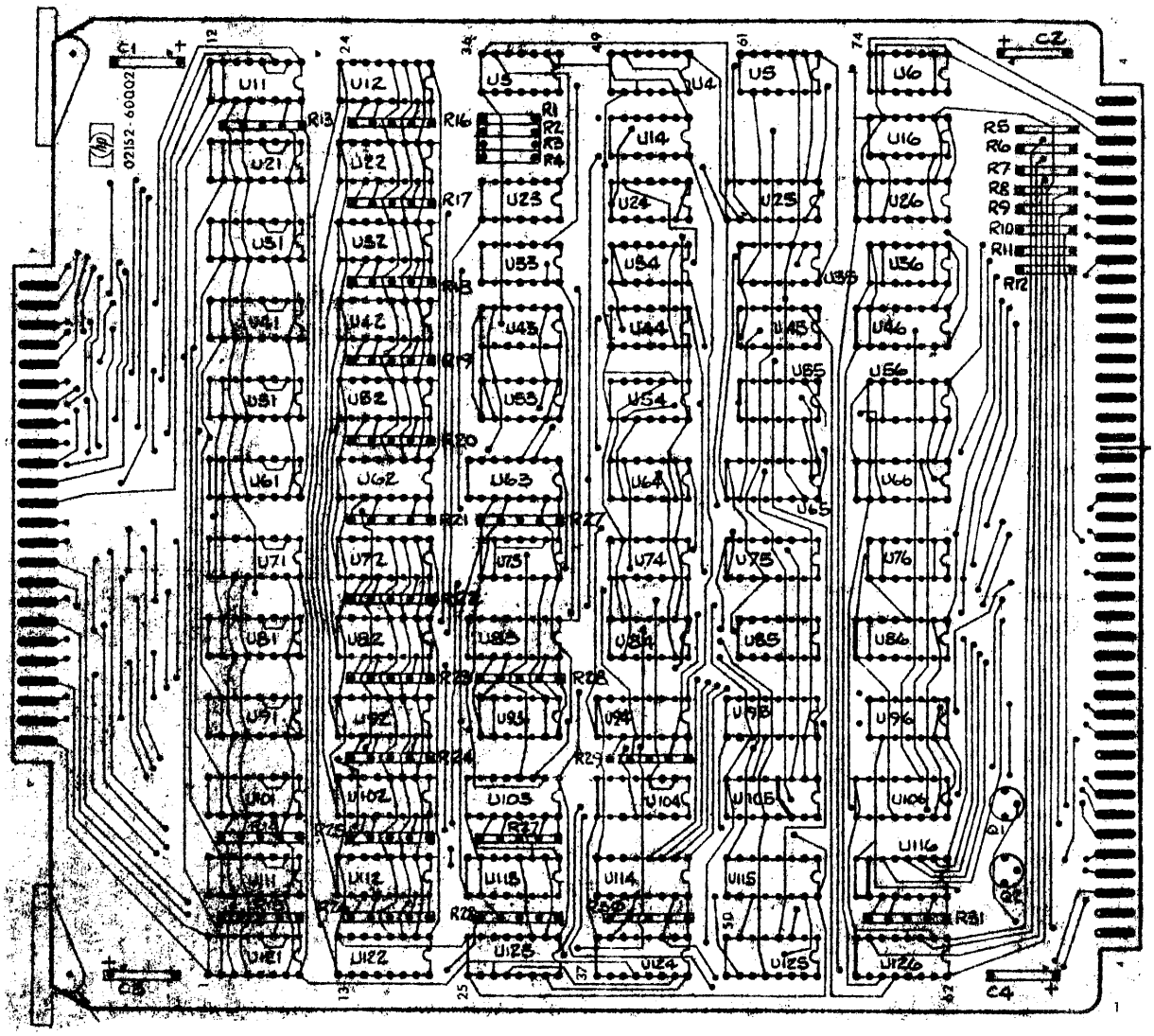


Figure 7-25. A-Adder Card A16, Parts Location View

ADDER CARD 64 (02152-60002, REV 10-31)

RRA

SAT 22

CLOCK → SRT · CLOCK

A97 3 · A96 5 · A95 7 · A94 9 · A93 11 · A92 13 · A91 15 · A90 17

A67 U126A · A66 U126B · A65 U126C · A64 U126D · A63 U106D · A62 U106A · A61 U106B · A60 U106C

U4A U4B U4C U25A U25B U25C U25D U5A U5B U5C U35A U35B U35C U35D U44A U44B U44C U65A U65B U65C U65D U54B U54C U75A U75B U75C U75D U64A U64D U63B U63C

U6A U6B U6C U6D U16A U16B U26A U26B U26C U26D U36A U36B U36C U36D U46A U46B U46C U46D U56A U56B U56C U56D U66A U66B U66C U66D U76A U76B U76C U76D U86A U86B U86C U86D U55A U55B U85A U85B

U23A U23B U24A U24B U33A U33B U34A U34B U43A U43B U53A U53B U63A U63B U74A U74B U64A U84A U84B

R4 −2V

U105B 12 4 3 · U125B 12 3 4 · +4.5V

A57 4 · A56 6 · A55 8 · A54 10 · A53 12 · A52 14 · A51 16 · A50 18

U125A 16 1 · U125B 15 2 · U125D 4 11 · U125F 9 8 · U105F 9 8 · U105A 16 1 · U105B 15 2 · U105D 11 6

AYS AYE AYS AY4

A07 MFF U83D · A06 MFF U83C · A05 MFF U83B · A04 MFF U83A

A13 MFF U63D · A02 MFF U63C · A01 MFF U63B · A00 MFF U63A

CLOCK 8

U113D 10 9 · U113C 7 8 12 · U113B 5 14 15 · U113A 1 2 16

CLOCK 9 8

AE7 SFF U123A · AE6 SFF U123B · AE5 SFF U123C · AE4 SFF U123D

10 (READ) −2V

AE3 SFF U95D · AE2 SFF U95A · AE1 SFF U95B · AE0 SFF U95C

10 (READ) −2V

A39 SFF U112D · A38 SFF U112C · A37 SFF U112B · A36 SFF U112A

A35 SFF U22D · A34 SFF U22C · A33 SFF U22B · A32 SFF U22A

A47 SFF U122D · A46 SFF U122C · A45 SFF U122B · A44 SFF U122A

A43 SFF U12D · A42 SFF U12C · A41 SFF U12B · A40 SFF U12A

AS6 SFF U114A · AS5 SFF U114B · AS4 SFF U114C · AS3 SFF U114D

AS7 SFF U103A · AS2 SFF U103B · AS1 SFF U103C · AS0 SFF U103D

U125E 9 10

R10D R10C R10B R10A R13D R13C R13B R13A R30A R30B R30C R30D R37A R37B R37C R37D −2V

A39 **BB · A38 **Y · A37 **V · A36 **S · A35 **N · A34 **K · A33 **F · A32 **C

A47 *24 · A46 *21 · A45 *18 · A44 *15 · A43 *12 · A42 *9 · A41 *6 · A40 *3

A56 **80 · A55 **81 · A54 **79 · A53 **77 · A57 **82 · A52 **75 · A51 **73 · A50 **71

NOTES:
1. ALL REGISTORS ARE 330 OHMS.
* DENOTES TOP EDGE CONNECTOR PIN.
** DENOTES BACKPLANE CONNECTOR PIN.

CIA 20 · PCY 66 · TCA 45 · ASX 62 65 · RYS 36 · RYE 38 · RY5 26 · RY4 24

Figure 7-26. A-Adder Card A16, Logic Diagram

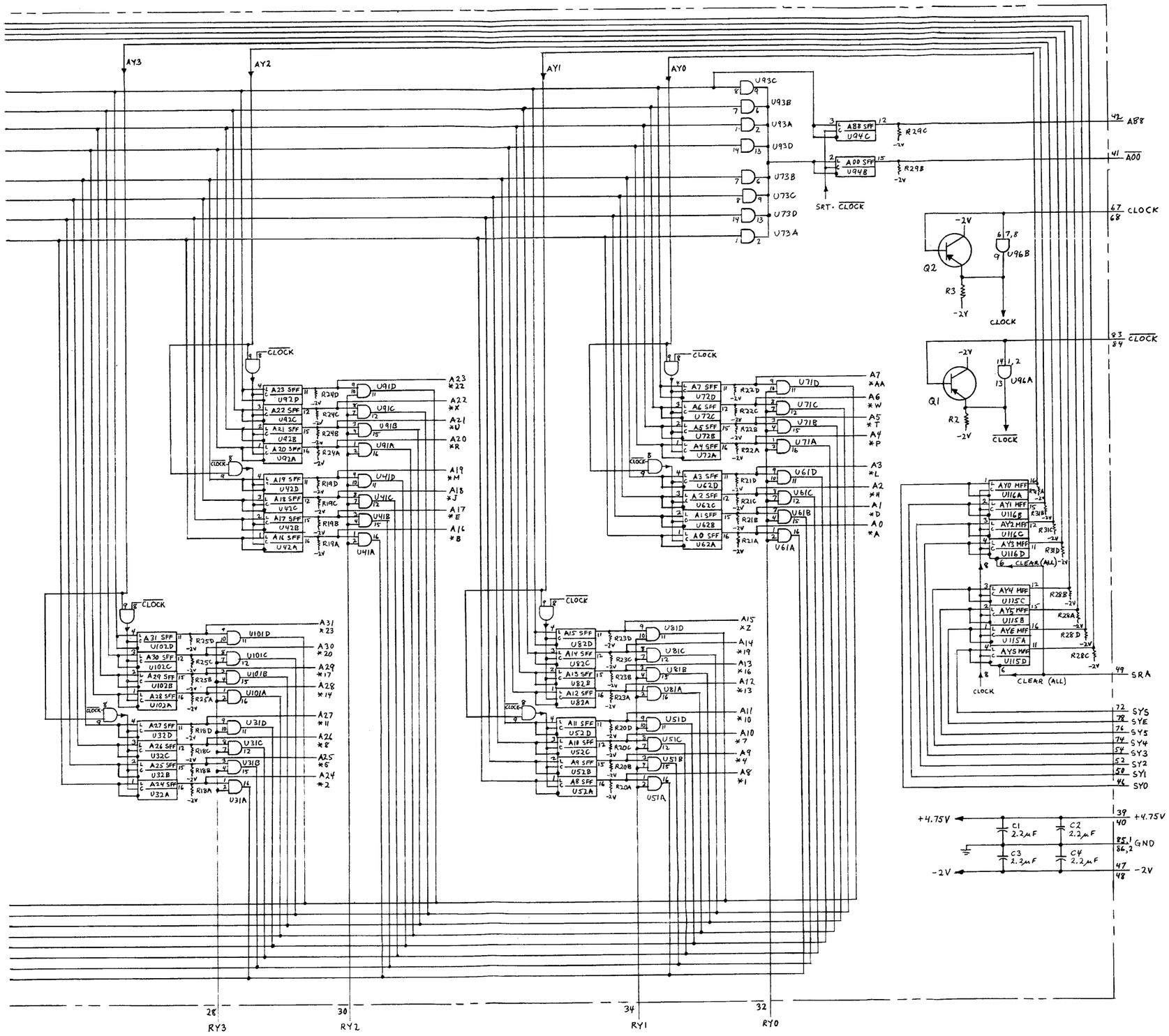


Figure 7-26. A-Adder Card A16, Logic Diagram (Cont'd)

Table 7-14. FPP Interface Card A17, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A17	02152-60017		I/O INTERFACE CARD	28480	02152-60017
A17C1	0150-0050		C:FXD CER DISC 1000 PF +80-20% 1000VDCW	56289	C0678102E102ZE19-CDH
A17C2	0150-0050		C:FXD CER DISC 1000 PF +80-20% 1000VDCW	56289	C0678102E102ZE19-CDH
A17C3 - A17C6	0180-0197		C:FXD ELECT :.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
A17Q1	1853-0015		TSTR:SI PNP	80131	2N3640
A17Q2	1853-0015		TSTR:SI PNP	80131	2N3640
A17R1 - A17R16	0683-4715		R:FXD COMP 470 OHM 5% 1/4W	01121	CB 4715
A17R17	0683-1015		R:FXD COMP 1CO OHM 5% 1/4W	01121	CB 1015
A17R18	0683-4715		R:FXD COMP 470 OHM 5% 1/4W	01121	CB 4715
A17R19	0683-1015		R:FXD COMP 1CO OHM 5% 1/4W	01121	CB 1015
A17R20 - A17R22	0683-4715		R:FXD COMP 470 OHM 5% 1/4W	01121	CB 4715
A17R23 - A17R28	1810-0047		RESISTOR NETWORK	28480	1810-0047
A17R29 - A17R40	0683-3315		R:FXD COMP 330 OHM 5% 1/4W	01121	CB 3315
A17U21	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A17U24	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A17U25	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A17U31	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A17U32	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
A17U33	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
A17U34	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A17U35	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A17U41	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A17U42	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A17U43	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186
A17U44	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A17U45	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A17U51	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A17U52	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A17U53	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A17U54	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A17U55	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A17U61	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A17U62	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A17U63	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A17U64	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
A17U65	1820-0482		INTEGRATED CIRCUIT: CTL	07263	SL15964
A17U71	1820-0485		INTEGRATED CIRCUIT: CTL	07263	SL15961
A17U72	1820-0488		INTEGRATED CIRCUIT: CTL	07263	SL15965
A17U73	1820-0187		INTEGRATED CIRCUIT: CTL	07263	SL15966
A17U74	1820-0486		INTEGRATED CIRCUIT: CTL	07263	SL15963
A17U75	1820-0186		INTERGRATED CIRCUIT:	28480	1820-0186

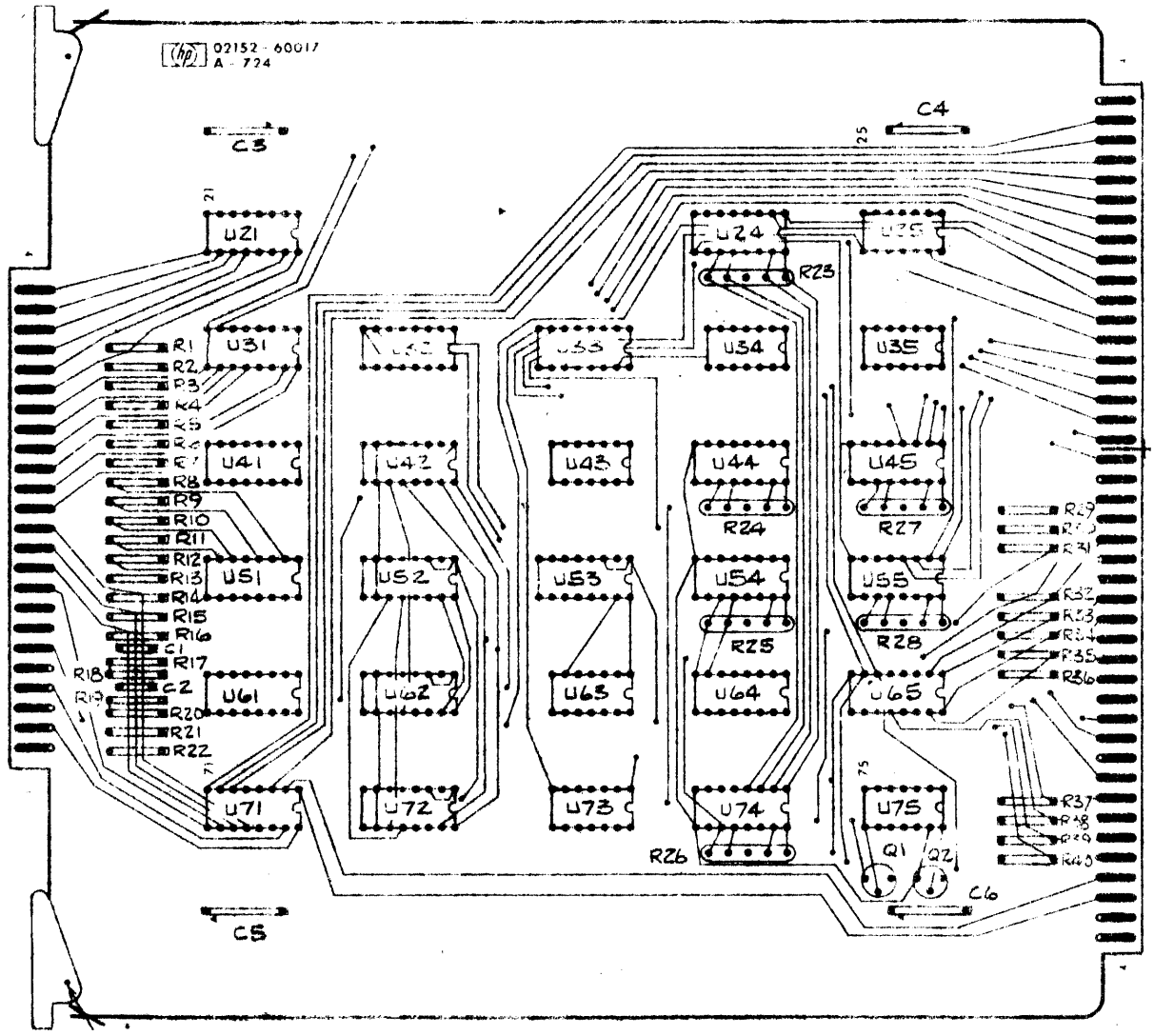
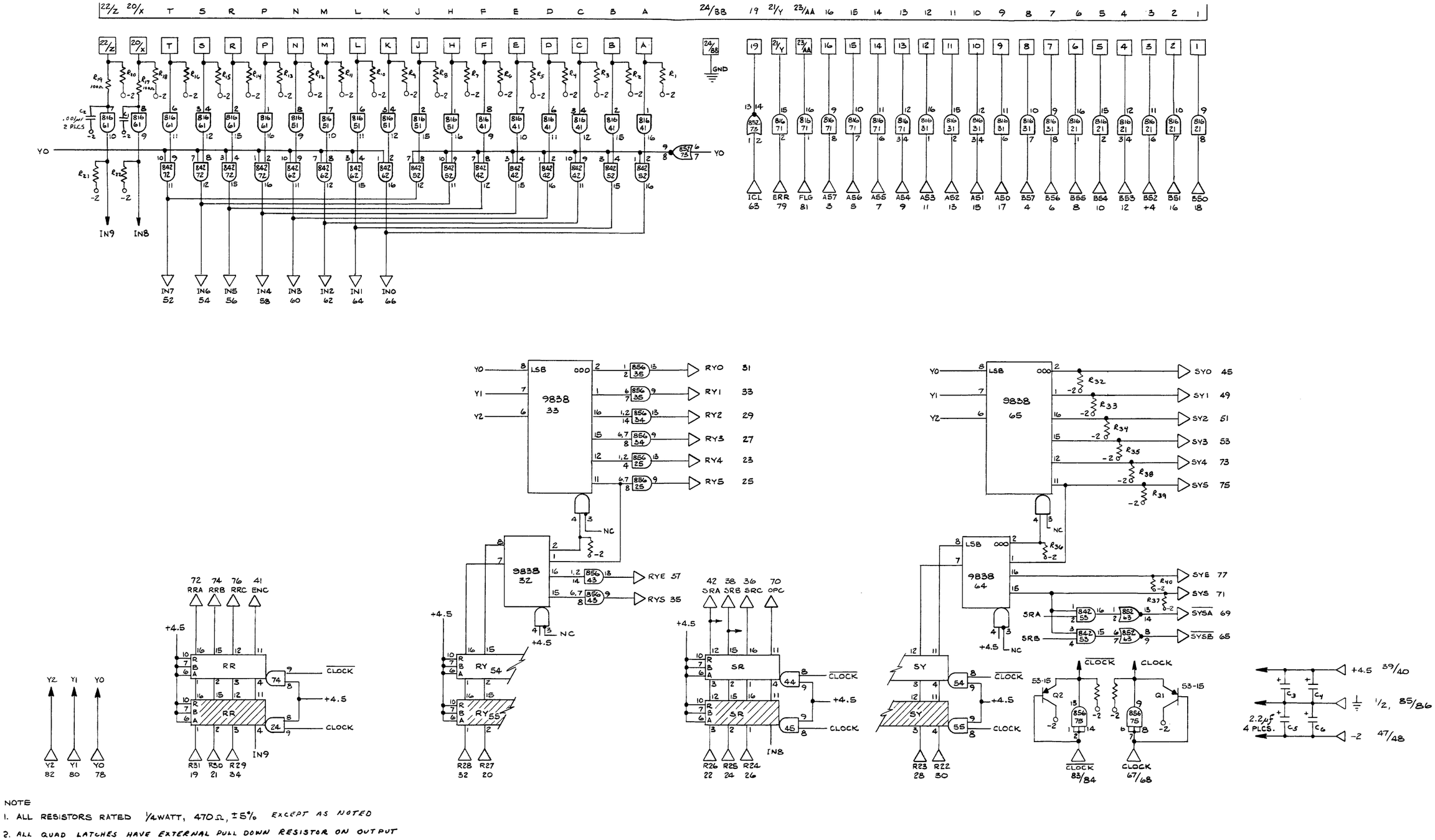


Figure 7-27. FPP Interface Card A17, Parts Location View

FPP INTERFACE CARD (02152-60017, REV 724)



NOTE

1. ALL RESISTORS RATED 1/4 WATT, 470Ω, ±5% EXCEPT AS NOTED
2. ALL QUAD LATCHES HAVE EXTERNAL PULL DOWN RESISTOR ON OUTPUT

Figure 7-28. FPP Interface Card A17, Logic Diagram

Table 7-15. Regulator Card A2A3, Replaceable Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A3	02152-60009		POWER SUPPLY REGULATOR CARD	28480	02125-60009
A3C25	0180-2206	1	C:FXD ELECT 50 UF 10% 6VDCW	56289	150D606X9006B2
A3C26	0160-0153	1	C:FXD MY 0.031 UF 10% 200VDCW	56289	192P10292-PTS
A3C30	0160-0162	1	C:FXD MY 0.022 UF 10% 200VDCW	56289	192P22392-PTS
A3C31	0160-0174	1	C:FXD CER 0.47 UF +80-20% 25VDCW	56289	5C1187S-CML
A3C32	0180-0061	3	C:FXD ELECT 100UF +100%-10% 15VDCW	56289	30107G015DD4
A3C40	0180-2144	1	C:FXD ELECT 200 UF +75-10% 25VDCW	28480	0180-2144
A3C45	0180-2217	1	C:FXD ELECT 350 UF +75-10% 50VDCW	28480	0180-2217
A3C46	0180-0061		C:FXD ELECT 100UF +100%-10% 15VDCW	56289	30107G015DD4
A3C50	0180-0291	1	C:FXD ELECT 1.0UF 10% 35VDCW	56289	150D105X9035A2-DYS
A3C51	0180-0141	1	C:FXD ELECT 50 UF +75-10% 50VDCW	56289	300506G050DD2-DSM
A3C60	0150-0096	1	C:FXD CER 0.05 UF +80-20% 100VDCW	91418	TA
A3C61	0180-0061		C:FXD ELECT 100UF +100%-10% 15VDCW	56289	30107G015DD4
A3C70	0180-0374	1	C:FXD ELECT 10.0 UF 10% 20VDCW	56289	150D106X9020B2-76
A3C71	0180-0116	1	C:FXD ELECT 6.8 UF 10% 35VDCW	56289	150D685X9035B2-DYS
A3C72	0150-0121	3	C:FXD CER 0.1 UF +80-20% 50VDCW	56289	5C5081S-CML
A3C73	0150-0121		C:FXD CER 0.1 UF +80-20% 50VDCW	56289	5C5081S-CML
A3CR20	1902-3082	2	DIODE BREAKDOWN:4.64V 5%	28480	1902-3082
A3CR21	1901-0040	10	DIODE:SILICON 30MA 30WV	07263	FDG1088
A3CR22	1901-0158	6	DIODE:SILICON 0.75A 200 PIV	28480	1901-0158
A3CR30	1902-3139	1	DIODE:BREAKDOWN 8.25V 5%	04713	SZ10939-158
A3CR35	1901-0040		DIODE:SILICON 30MA 30WV	07263	FDG1088
A3CR40	1901-0158		DIODE:SILICON 0.75A 200 PIV	28480	1901-0158
A3CR41	1901-0158		DIODE:SILICON 0.75A 200 PIV	28480	1901-0158
A3CR42	1902-3082		DIODE BREAKDOWN:4.64V 5%	28480	1902-3082
A3CR50	1901-0040		DIODE:SILICON 30MA 30WV	07263	FDG1088
A3CR60	1901-0158		DIODE:SILICON 0.75A 200 PIV	28480	1901-0158
A3CR61	1901-0158		DIODE:SILICON 0.75A 200 PIV	28480	1901-0158
A3CR62	1901-0040		DIODE:SILICON 30MA 30WV	07263	FDG1088
A3CR63	1901-0158		DIODE:SILICON 0.75A 200 PIV	28480	1901-0158
A3CR70					
A3CR73	1901-0040		DIODE:SILICON 30MA 30WV	07263	FDG1088
A3CR81	1902-0049	1	DIODE:BREAKDOWN 6.19V 5%	04713	SZ10939-122
A3CR82	1902-0126	1	DIODE:BREAKDOWN 7.61V 5%	04713	SZ10939-14
A3CR95	1901-0040		DIODE:SILICON 30MA 30WV	07263	FDG1088
A3CR96	1901-0040		DIODE:SILICON 30MA 30WV	07263	FDG1088
A3Q1	1854-0072	15	TSTR:SI NPN	80131	2N3054
A3Q2	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q3	1853-0020	10	TSTR:SI PNP(SELECTED FROM 2N3702)	28480	1853-0020
A3Q4	1855-0010	1	TSTR:SI	80131	2N2646
A3Q5	1853-0020		TSTR:SI PNP(SELECTED FROM 2N3702)	28480	1853-0020
A3Q6	1853-0039	3	TSTR:SI PNP	80131	2N3638A
A3Q7	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q8	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q9	1853-0020		TSTR:SI PNP(SELECTED FROM 2N3702)	28480	1853-0020
A3Q10	1853-0041	1	TSTR:SI PNP	02735	38640
A3Q11	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q12	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q13	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q14	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q15	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q16	1853-0020		TSTR:SI PNP(SELECTED FROM 2N3702)	28480	1853-0020
A3Q17	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q18	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q19	1853-0020		TSTR:SI PNP(SELECTED FROM 2N3702)	28480	1853-0020
A3Q20	1853-0039		TSTR:SI PNP	80131	2N3638A
A3Q23	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q24	1853-0020		TSTR:SI PNP(SELECTED FROM 2N3702)	28480	1853-0020
A3Q25	1853-0020		TSTR:SI PNP(SELECTED FROM 2N3702)	28480	1853-0020
A3Q26	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q27	1853-0020		TSTR:SI PNP(SELECTED FROM 2N3702)	28480	1853-0020
A3Q28	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q29	1853-0020		TSTR:SI PNP(SELECTED FROM 2N3702)	28480	1853-0020
A3Q30	1853-0039		TSTR:SI PNP	80131	2N3638A
A3Q31	1853-0001	1	TSTR:SI PNP(SELECTED FROM 2N1132)	28480	1853-0001
A3Q32	1854-0072		TSTR:SI NPN	80131	2N3054
A3Q33	1853-0020		TSTR:SI PNP(SELECTED FROM 2N3702)	28480	1853-0020
A3R20	0757-0416	1	R:FXD MET FLM 511 OHM 1% 1/8W	14674	C4
A3R21	0698-3153	1	R:FXD MET FLM 3.83K 1% 1/8W	91637	MFF-1/10-32
A3R22	0698-3159	1	R:FXD MET FLM 26.1K OHM 1% 1/8W	75042	CEA
A3R23	0757-0442	4	R:FXD MET FLM 10.0K 1% 1/8W	14674	C4
A3R24	0698-3441	1	R:FXD MET FLM 215 OHM 1% 1/8W	91637	MF-1/10-32
A3R25	2100-1788	7	R:VAR FLM 500 OHM 10% LIN 1/2W	28480	2100-1788
A3R26	0698-0084	1	R:FXD MET FLM 2.15K 1% 1/8W	14674	C4
A3R27	0698-0083	2	R:FXD MET FLM 1.96K OHM 1% 1/8W	14674	C4

Table 7-15. Regulator Card A2A3, Replaceable Parts (Cont'd)

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A3R30	0684-3311	3	R:FXD COMP 33C OHM 10% 1/4W	01121	CB 3311
A3R31	0684-1021	13	R:FXD COMP 100 OHM 10% 1/4W	01121	CB 1021
A3R32	0684-1021		R:FXD COMP 100 OHM 10% 1/4W	01121	CB 1021
A3R33	0684-4711	2	R:FXD COMP 47C OHM 10% 1/4W	01121	CB 4711
A3R34	0684-1001	1	R:FXD COMP 10 OHM 10% 1/4W	01121	CB 1001
A3R35	0684-2201	3	R:FXD COMP 22 OHM 10% 1/4W	01121	CB 2201
A3R38	0684-1521	3	R:FXD COMP 150 OHM 10% 1/4W	01121	CB 1521
A3R39	0684-1521		R:FXD COMP 150 OHM 10% 1/4W	01121	CB 1521
A3R40	0684-1021		R:FXD COMP 100 OHM 10% 1/4W	01121	CB 1021
A3R41	0757-1094	3	R:FXD MET FLM 1.47K OHM 1% 1/8W	14674	C4 T-0
A3R42	0698-3136	1	R:FXD MET FLM 17.8K OHM 1% 1/8W	14674	C4
A3R43	0698-5631	1	R:FXD FLM 5.11K 0.5% 1/4W	28480	0698-5631
A3R45	0684-3921	1	R:FXD COMP 390 OHM 10% 1/4W	01121	CB 3921
A3R50	0684-4721	1	R:FXD COMP 470 OHM 10% 1/4W	01121	CB 4721
A3R51	0684-3311		R:FXD COMP 33C OHM 10% 1/4W	01121	CB 3311
A3R52	0757-0419	1	R:FXD MET FLM 681 OHM 1% 1/8W	14674	C4
A3R53	0757-1094		R:FXD MET FLM 1.47K OHM 1% 1/8W	14674	C4 T-0
A3R54	2100-1788		R:VAR FLN 500 OHM 10% LIN 1/2W	28480	2100-1788
A3R55	0757-0274	1	R:FXD MET FLM 1.21K OHM 1% 1/8W	28480	0757-0274
A3R60	0684-1021		R:FXD COMP 1000 OHM 10% 1/4W	01121	CB 1021
A3R61	2100-1788		R:VAR FLN 500 OHM 10% LIN 1/2W	28480	2100-1788
A3R62	2100-1788		R:VAR FLN 500 OHM 10% LIN 1/2W	28480	2100-1788
A3R63	0757-0440	1	R:FXD MET FLM 7.50K 1% 1/8W	14674	C4
A3R64	0684-6831	1	R:FXD COMP 68K OHM 10% 1/4W	01121	CB 6831
A3R65	0757-1094		R:FXD MET FLM 1.47K OHM 1% 1/8W	14674	C4 T-0
A3R70	0757-0442		R:FXD MET FLM 10.0K 1% 1/8W	14674	C4
A3R71	0698-3156	2	R:FXD MET FLM 14.7K OHM 1% 1/8W	14674	C4
A3R72	0698-3156		R:FXD MET FLM 14.7K OHM 1% 1/8W	14674	C4
A3R73	0757-0442		R:FXD MET FLM 10.0K 1% 1/8W	14674	C4
A3R80	0684-1021		R:FXD COMP 1000 OHM 10% 1/4W	01121	CB 1021
A3R81	2100-1788		R:VAR FLN 500 OHM 10% LIN 1/2W	28480	2100-1788
A3R82	0698-0085	1	R:FXD MET FLM 2.61K OHM 1% 1/8W	14674	C4
A3R83	0684-4711		R:FXD COMP 470 OHM 10% 1/4W	01121	CB 4711
A3R84	0684-2211	3	R:FXD COMP 220 OHM 10% 1/4W	01121	CB 2211
A3R85	0684-3311		R:FXD COMP 330 OHM 10% 1/4W	01121	CB 3311
A3R86	0683-1505	1	R:FXD COMP 15 OHM 5% 1/4W	01121	CB 1505
A3R90	0757-0430	1	R:FXD MET FLM 2.21K OHM 1% 1/8W	28480	0757-0430
A3R91	2100-1788		R:VAR FLN 500 OHM 10% LIN 1/2W	28480	2100-1788
A3R92	2100-1788		R:VAR FLN 500 OHM 10% LIN 1/2W	28480	2100-1788
A3R93	0698-0083		R:FXD MET FLM 1.96K OHM 1% 1/8W	14674	C4
A3R94	0684-2231	1	R:FXD COMP 22K OHM 10% 1/4W	01121	CB 2231
A3R95	0684-1831	1	R:FXD COMP 18K OHM 10% 1/4W	01121	CB 1831
A3R96	0684-1021		R:FXD COMP 1000 OHM 10% 1/4W	01121	CB 1021
A3R100	0684-2221	2	R:FXD COMP 2200 OHM 10% 1/4W	01121	CB 2221
A3R101	0684-6821	1	R:FXD COMP 6.8K OHM 10% 1/4W	01121	CB 6821
A3R102	0684-1021		R:FXD COMP 1000 OHM 10% 1/4W	01121	CB 1021
A3R103	0684-1021		R:FXD COMP 1000 OHM 10% 1/4W	01121	CB 1021
A3R104	0684-2221		R:FXD COMP 2200 OHM 10% 1/4W	01121	CB 2221
A3R105	0757-0442		R:FXD MET FLM 10.0K 1% 1/8W	14674	C4
A3R106	0684-1521		R:FXD COMP 1500 OHM 10% 1/4W	01121	CB 1521
A3R107	0684-1011	7	R:FXD COMP 100 OHM 10% 1/4W	01121	CB 1011
A3R108	0684-5621	1	R:FXD COMP 5.6K OHM 10% 1/4W	01121	CB 5621
A3R109	0684-1021		R:FXD COMP 1000 OHM 10% 1/4W	01121	CB 1021
A3R110	0684-1021		R:FXD COMP 1000 OHM 10% 1/4W	01121	CB 1021
A3R111	0684-1021		R:FXD COMP 1000 OHM 10% 1/4W	01121	CB 1021
A3R112	0684-1021		R:FXD COMP 1000 OHM 10% 1/4W	01121	CB 1021
A3R113	0684-1011		R:FXD COMP 100 OHM 10% 1/4W	01121	CB 1011
A3R114	0684-1021		R:FXD COMP 1000 OHM 10% 1/4W	01121	CB 1021
A3R120	0684-1011		R:FXD COMP 100 OHM 10% 1/4W	01121	CB 1011
A3R123	0684-2211		R:FXD COMP 220 OHM 10% 1/4W	01121	CB 2211
A3R124	0684-1011		R:FXD COMP 100 OHM 10% 1/4W	01121	CB 1011
A3R125	0684-1011		R:FXD COMP 100 OHM 10% 1/4W	01121	CB 1011
A3R126	0684-1011		R:FXD COMP 100 OHM 10% 1/4W	01121	CB 1011
A3R127	0684-1011		R:FXD COMP 100 OHM 10% 1/4W	01121	CB 1011
A3R128	0684-2211		R:FXD COMP 220 OHM 10% 1/4W	01121	CB 2211
A3R129	0684-2201		R:FXD COMP 22 OHM 10% 1/4W	01121	CB 2201
A3R131	0684-2201		R:FXD COMP 22 OHM 10% 1/4W	01121	CB 2201

Table 7-16. Replaceable Chassis Parts

Reference Designation	HP Part Number	Qty	Description	Mfr Code	Mfr Part Number
A3	02152-60009		POWER SUPPLY REGULATOR CARD	28480	02125-60009
A7	02152-60008		TEST CARD	28480	02152-60008
A8	02152-60005		ROM ADDRESS CARD	28480	02152-60005
A9	02152-60007		ROM CARD	28480	02152-60007
A10	02152-60004		D REGISTER CARD	28480	02152-60004
A11	02152-60001		SHIFTER CARD	28480	02152-60001
A12	02152-60002		ADDER CARD	28480	02152-60002
A13	02152-60001		SHIFTER CARD	28480	02152-60001
A14	02152-60002		ADDER CARD	28480	02152-60002
A15	02152-60001		SHIFTER CARD	28480	02152-60001
A16	02152-60002		ADDER CARD	28480	02152-60002
A17	02152-60017		I/O INTERFACE CARD	28480	02152-60017
A109	02152-60012		EAU TIMING CARD	28480	02152-60012
A110	02152-60011		EAU LOGIC CARD	28480	02152-60011
A209	02152-60013		EAU I/O CARD	28480	02152-60013
B1	3160-0072	2	FAN:TUBEAXIAL 115V 60 HZ	28480	3160-0072
B2	3160-0072		FAN:TUBEAXIAL 115V 60 HZ	28480	3160-0072
C7	0160-0128	1	C:FXD CER 2.2 UF 20% 25VDCW	56289	5C152C2S-CML
C11		1			
C13	0180-2223	3	C:FXD ELECT 160,000 UF +75-10% 10VDCW	56289	36D164G010DF2A-DQB
C62	0180-0197		C:FXD ELECT 2.2 UF 10% 20VDCW	56289	150D225X9020A2-DYS
C80	0160-1714	1	C:FXD ELECT 330 UF 10% 6VDCW	28480	0180-1714
CR5	NO NUMBER	1	DIODE: SCR. 40RC510	00000	OBD
CR6	NO NUMBER	1	DIODE: SCR. 40RC510	00000	OBD
CR7	NO NUMBER	1	DIODE: IN1192AR	00000	OBD
CR8	1901-0025	1	DIODE:SILICON 100MA/1V	07263	FD 2387
CR80	1884-0046	1	THYRISTOR:SCR 50V 25A	28480	1884-0046
CR90	1901-0346	1	DIODE:SILICON 10CPIV 1N3209R	04713	1N3209R
CR91	1901-0032	1	DIODE:SILICON 1N3209	04713	1N3209
DS1	1450-0419	1	LIGHT:INDICATOR SELECTED NE-2H	28480	1450-0419
L3	02152-60023	1	RF CHOKE ASSEMBLY	28480	02152-60023
L4	5081-0372	1	CHOKE 3MH	28480	50810372
Q21	1853-0231	1	TRANSISTOR SI PNP	04713	SS9320
Q22	1854-0063	1	TSTR:SI NPN	80131	2N3055
R1	0683-3335		R:FXD COMP 33K OHM 5% 1/4W	01121	CB 3335
R2	0757-0276	2	R:FXD MET FLM 61.9 OHM 1% 1/8W	28480	0757-0276
R5	0757-0346	2	R:FXD MET FLM 10 OHM 1% 1/8W	28480	0757-0346
R6	0757-0346		R:FXD MET FLM 10 OHM 1% 1/8W	28480	0757-0346
R7	0684-0271	1	R:FXD COMP 2.7 OHM 10% 1/4W	01121	CB 27G1
R8	0757-0401	1	R:FXD MET FLM 100 OHM 1% 1/8W	14674	C4
R82	0757-0276		R:FXD MET FLM 61.9 OHM 1% 1/8W	28480	0757-0276
R122	0811-2510	1	R:FXD WW 0.1 OHM 5% 25W	28480	0811-2510
R130	0757-0346	1	R:FXD FLM 10 OHM 1% 1/8W	28480	0757-0346
S1	3101-0003	1	SWITCH:TOGGLE DPST (ON-NONE-OFF)	04009	81024-GT
T1	5080-0378	1	TRANSFORMER	28480	5080-0378

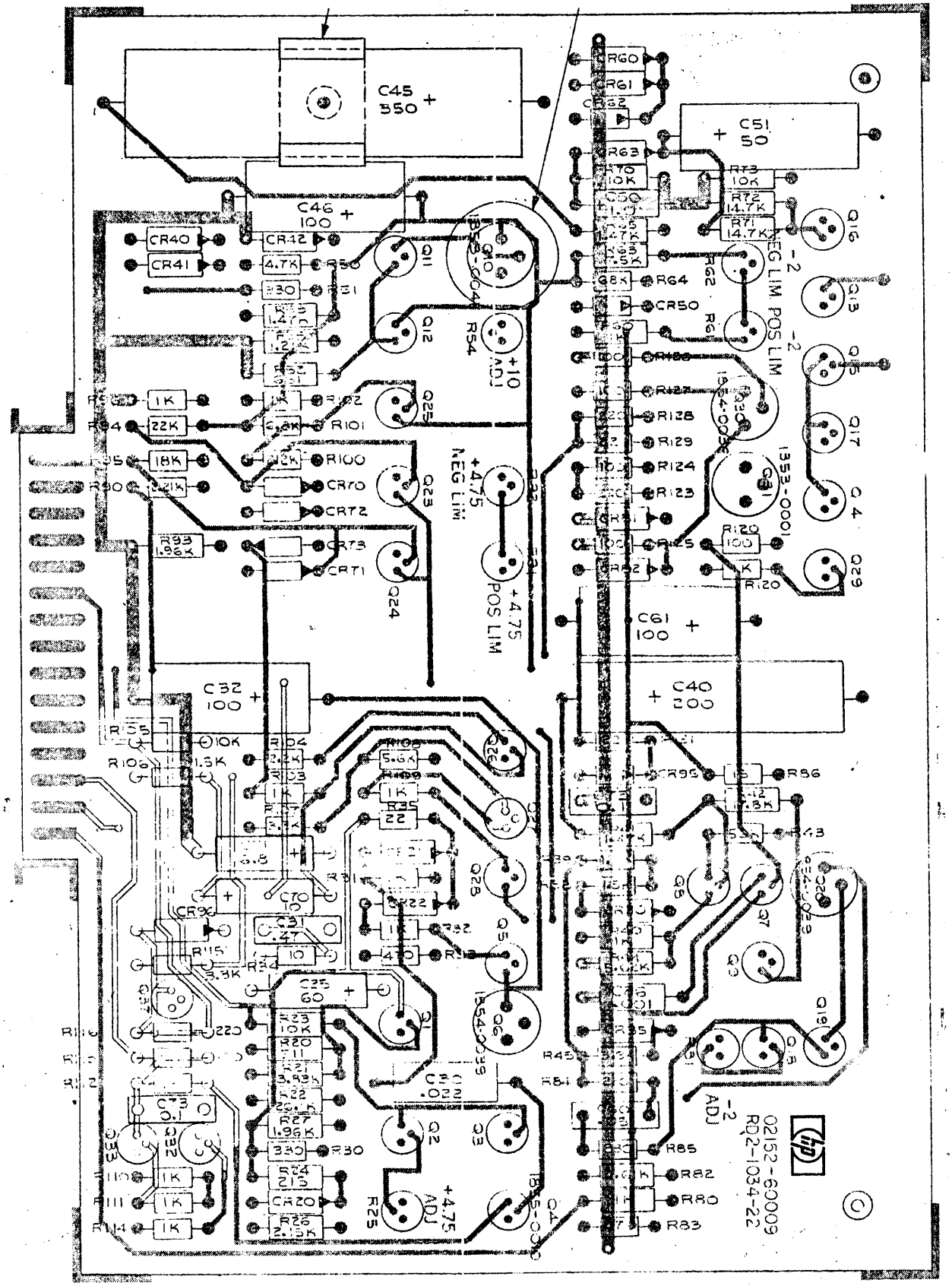


Figure 7-29. Regulator Card A2A3, Parts Location View

NOTE: CAPACITOR VALUES IN MICROFARADS UNLESS OTHERWISE INDICATED

Figure 7-30. FPP Power Supply, Schematic Diagram

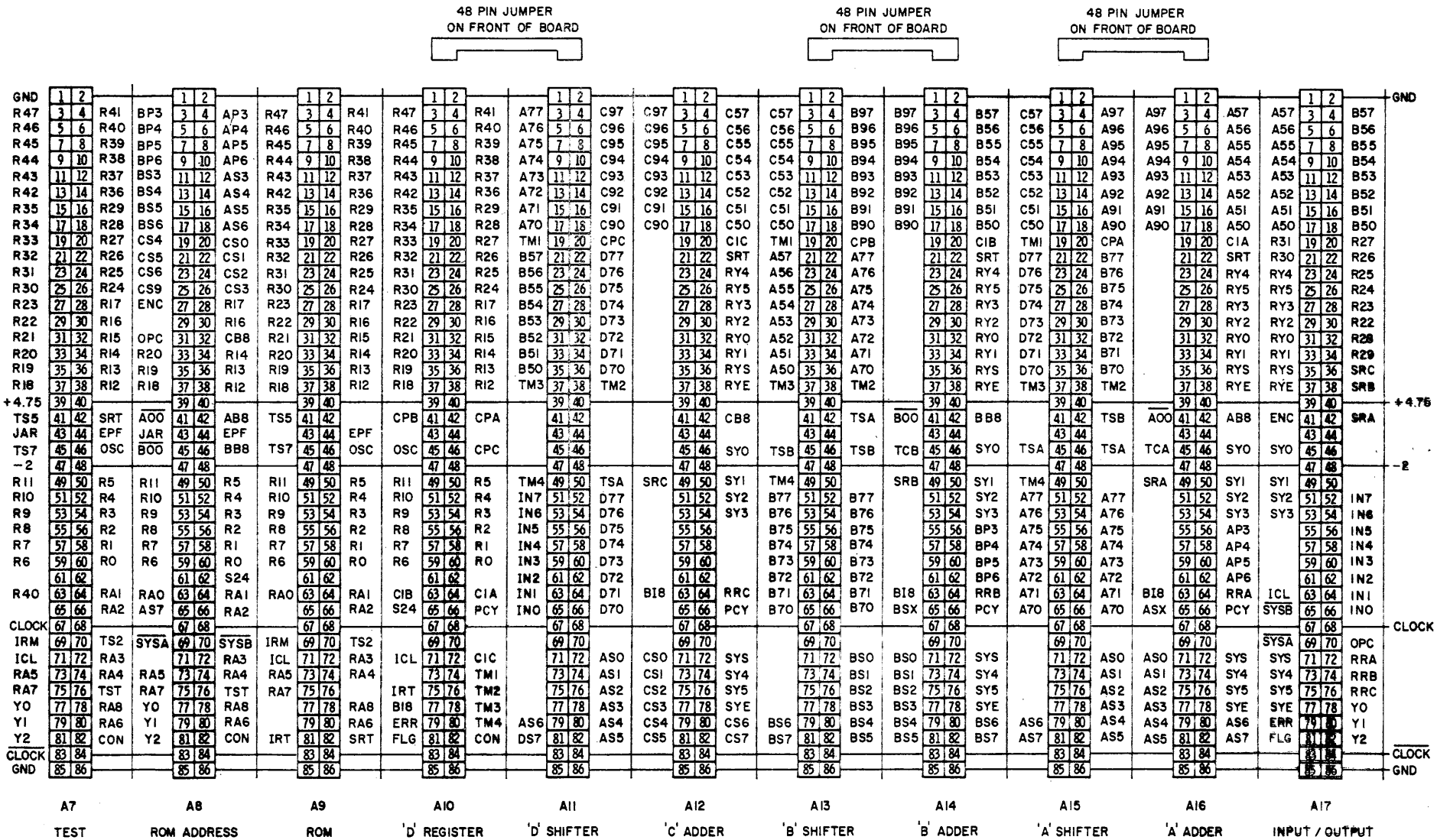


Figure 7-31. Backplane Wiring Diagram

NOTE:

DIAGRAM VIEWED FROM BACK PLANE

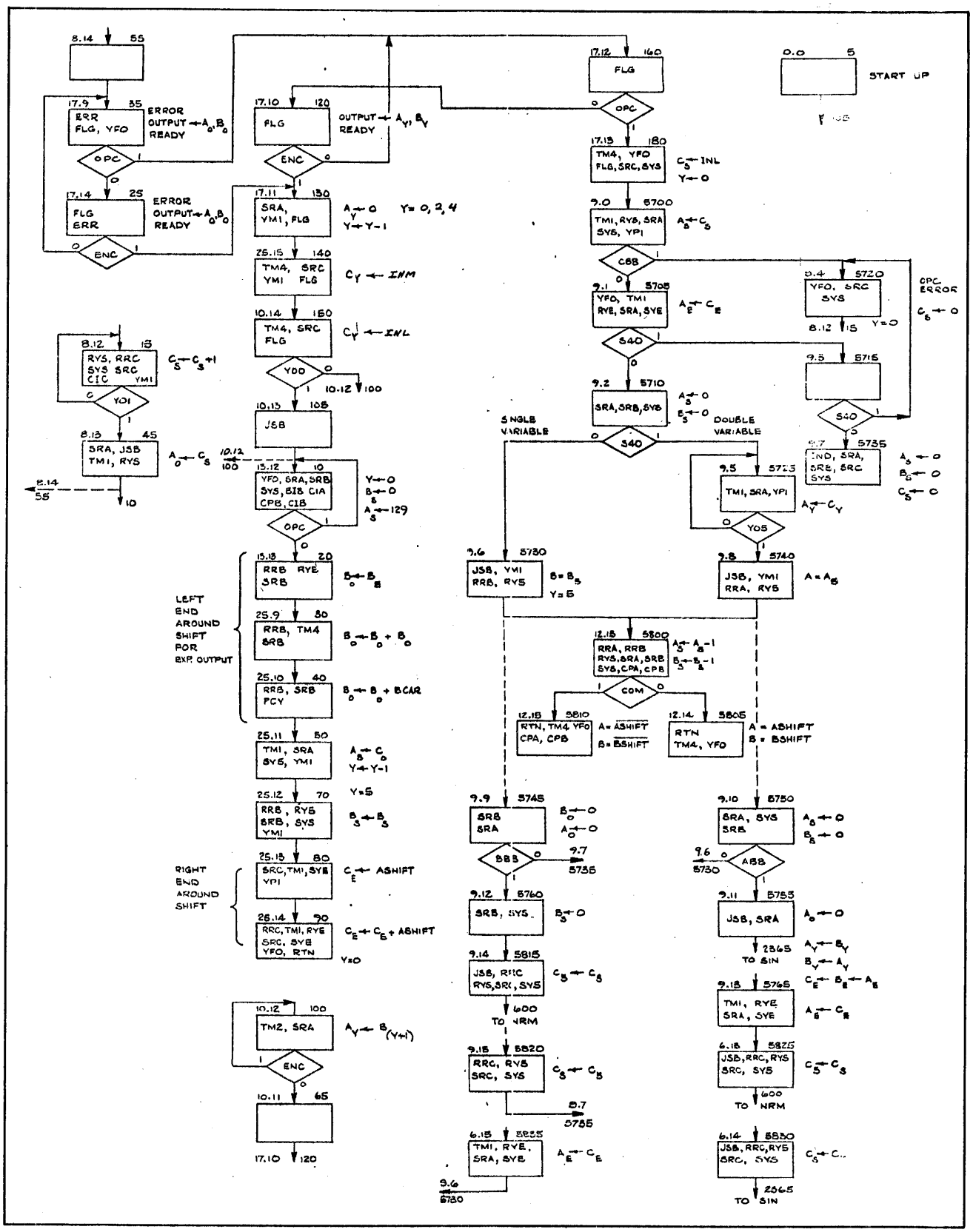


Figure 7-32. Entry Routine Flowchart

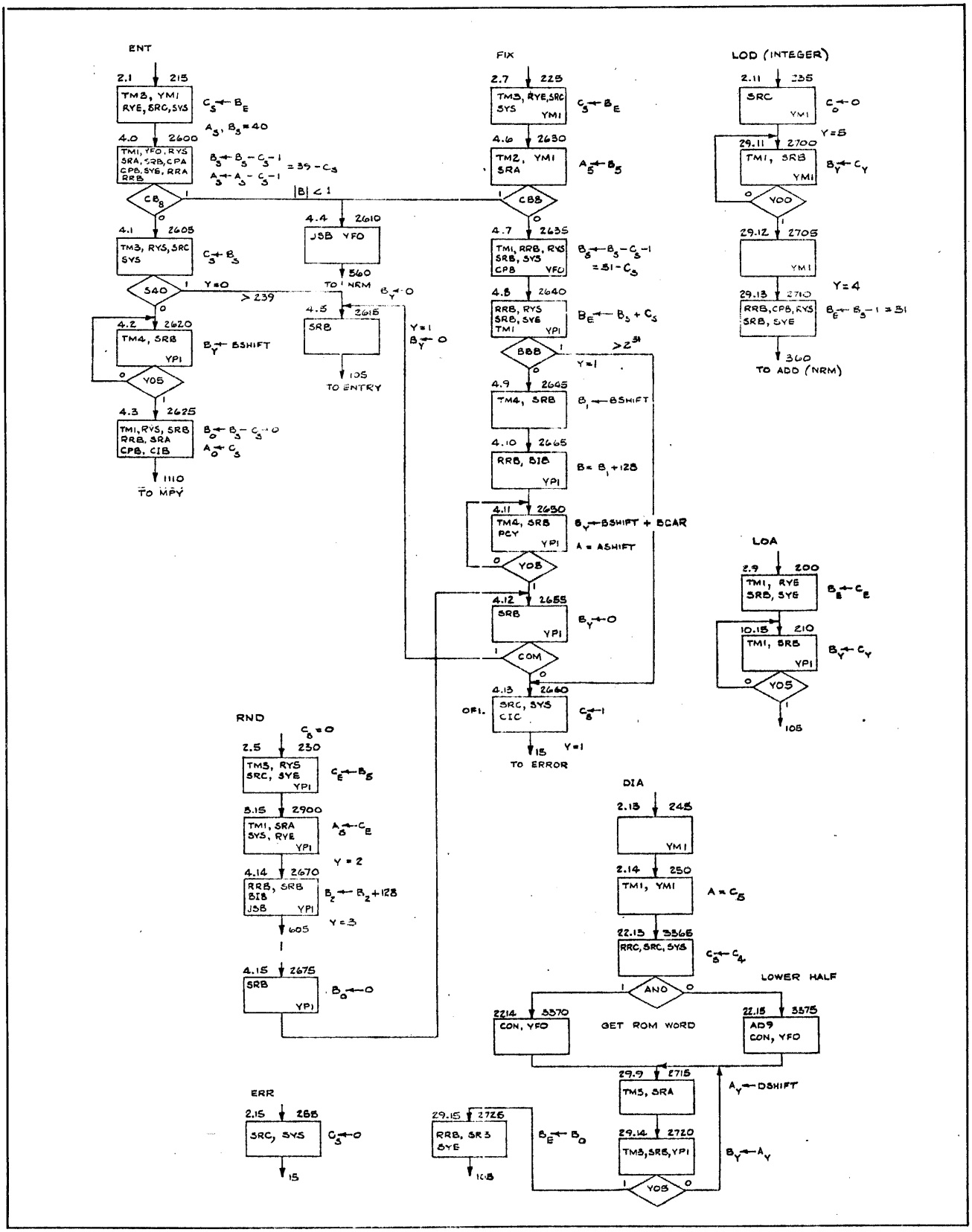


Figure 7-33. Entier/Fix/Load/Load I/Round Flowcharts

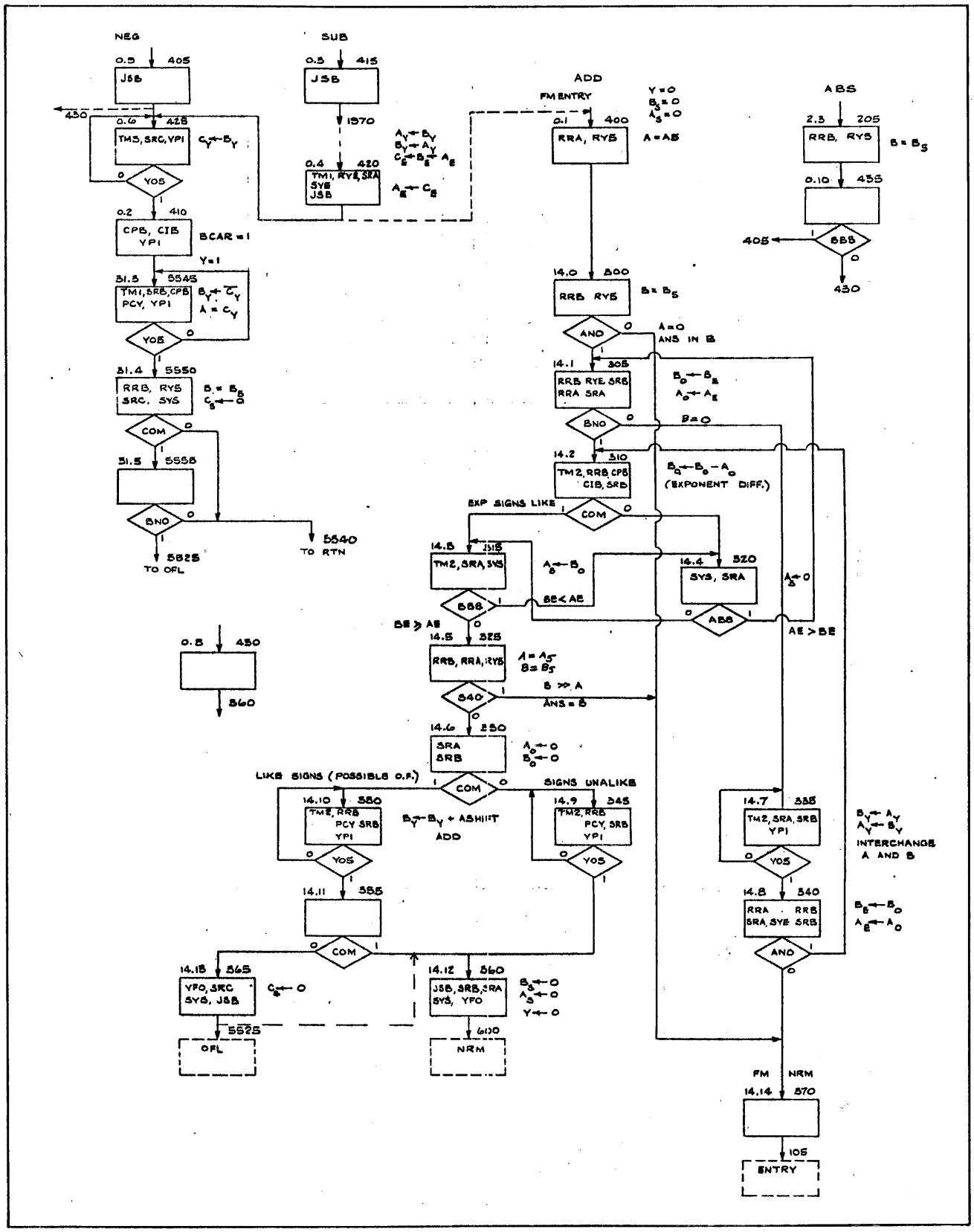


Figure 7-34. Add/Sub/Abs/Neg Flowcharts

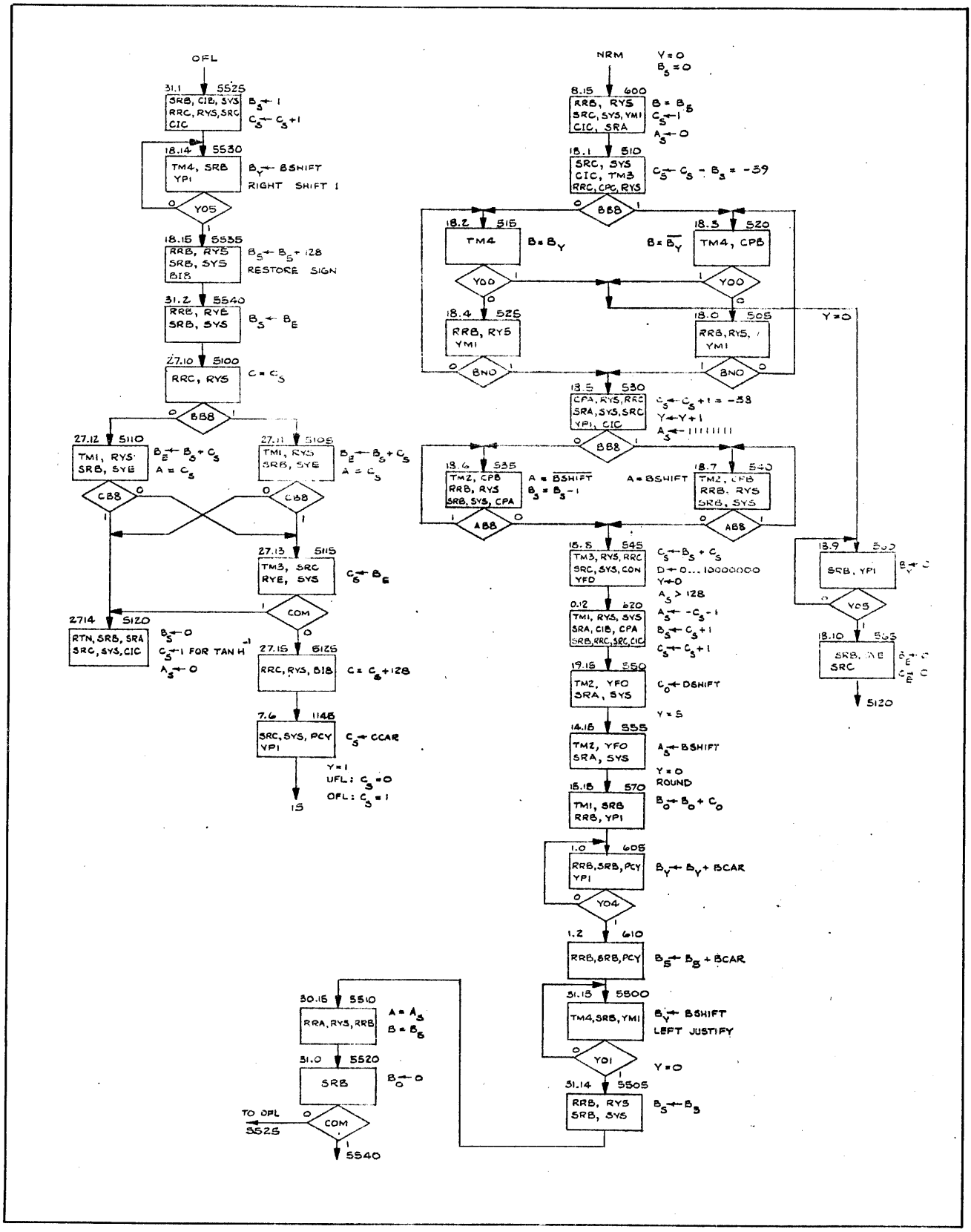


Figure 7-35. Overflow/Normalize Flowcharts

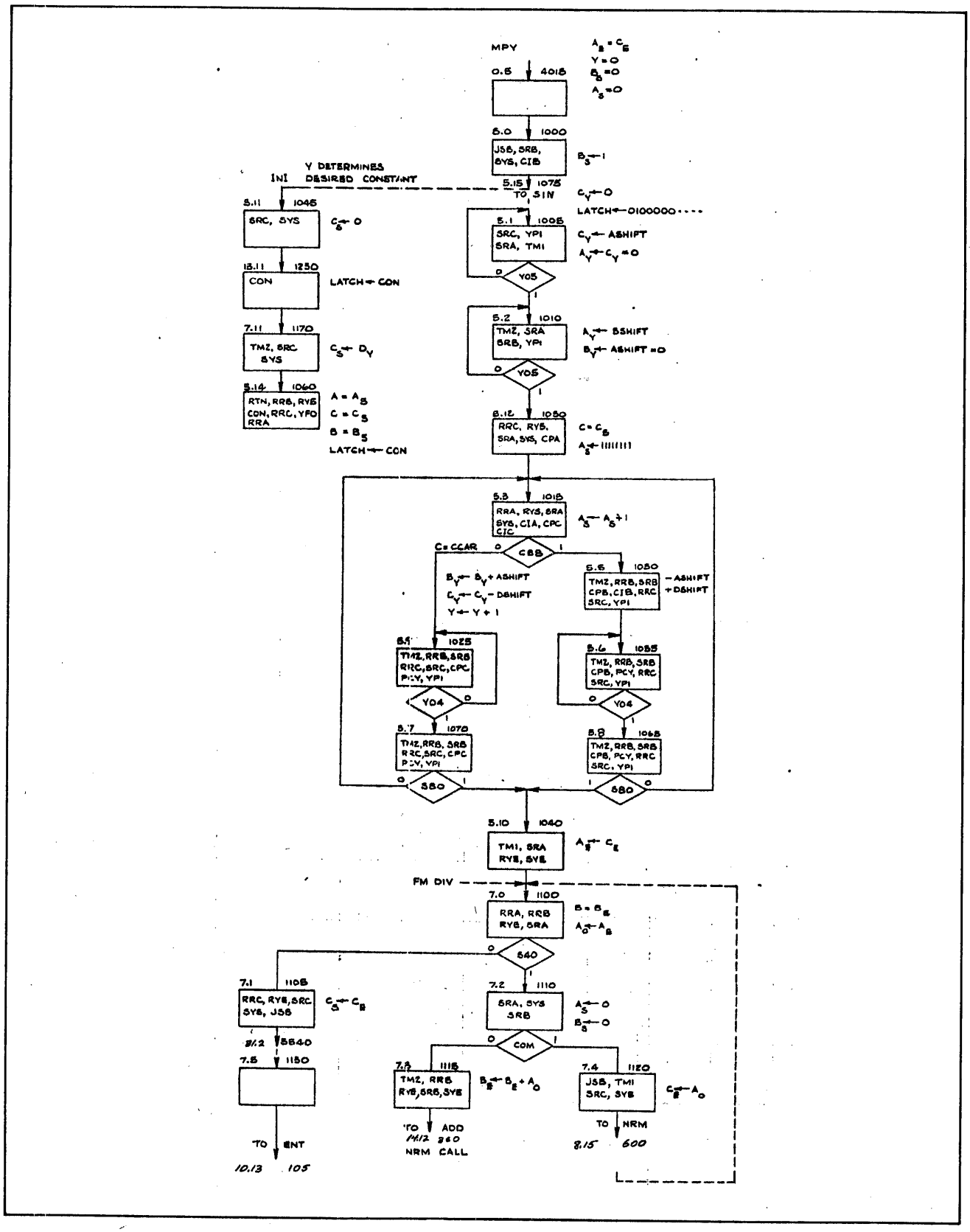


Figure 7-36. Multiply/Initialize Flowcharts

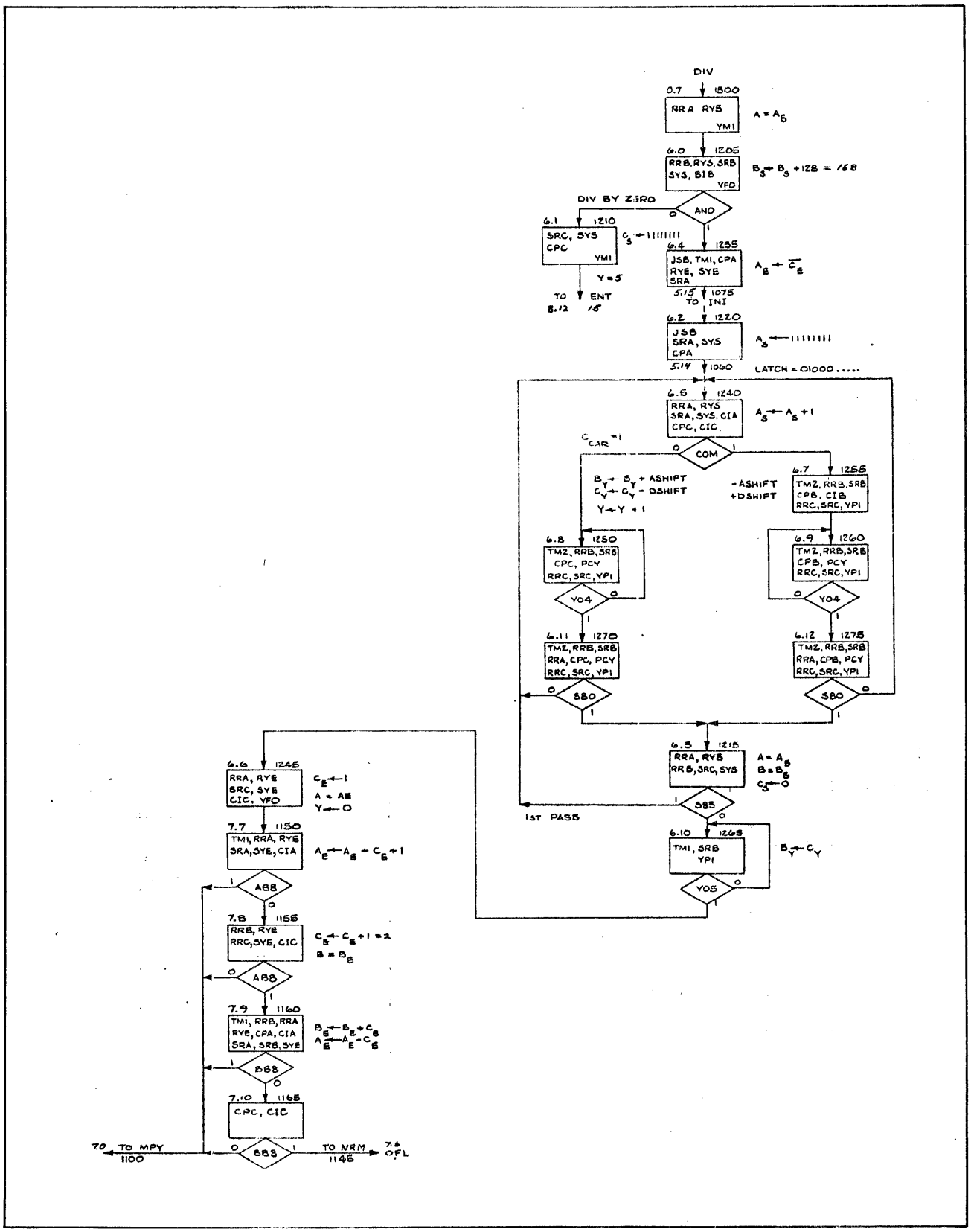


Figure 7-37. Division Routine Flowchart

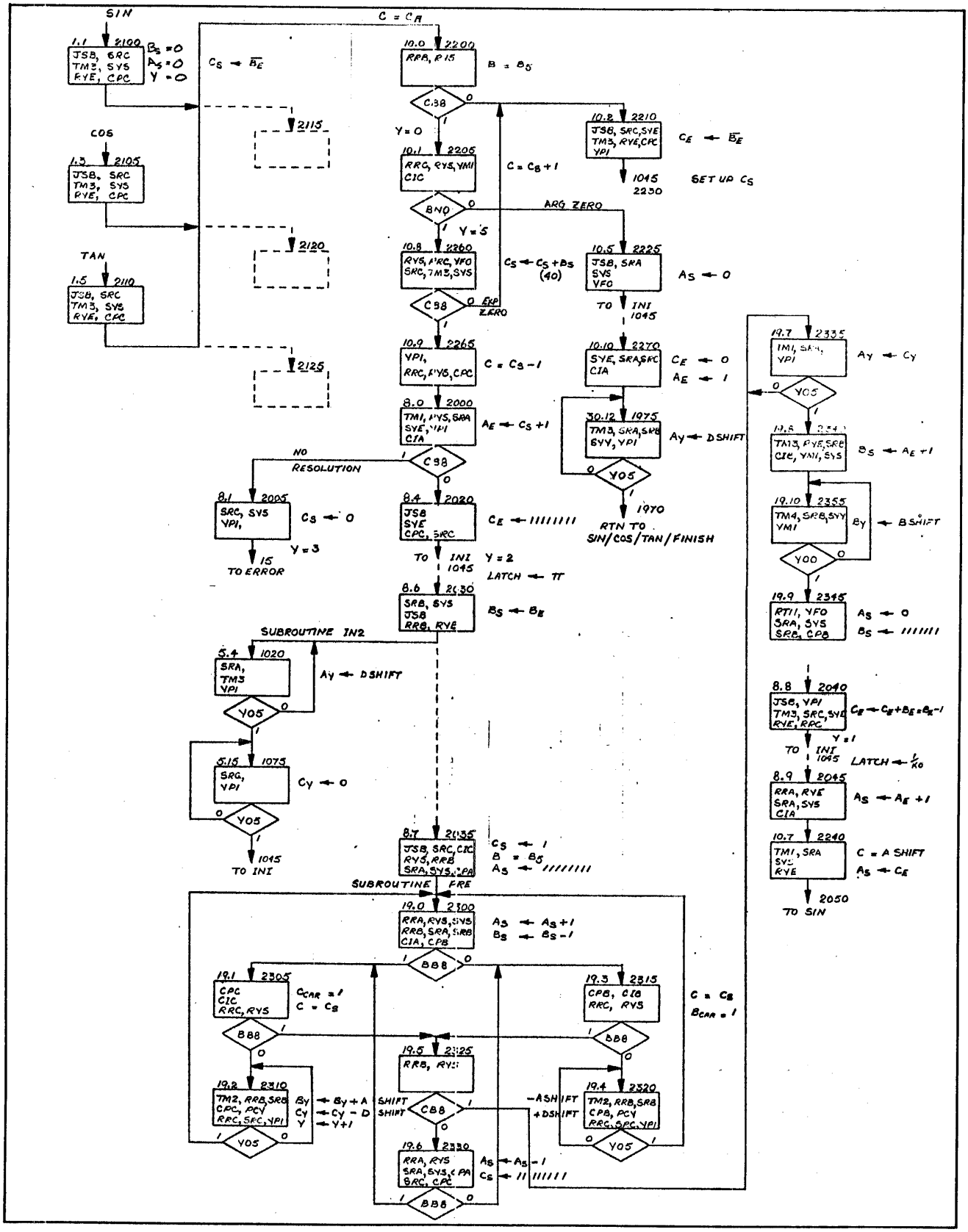


Figure 7-38. Sin/Cos/Tan/Hyper Prescale Flowcharts

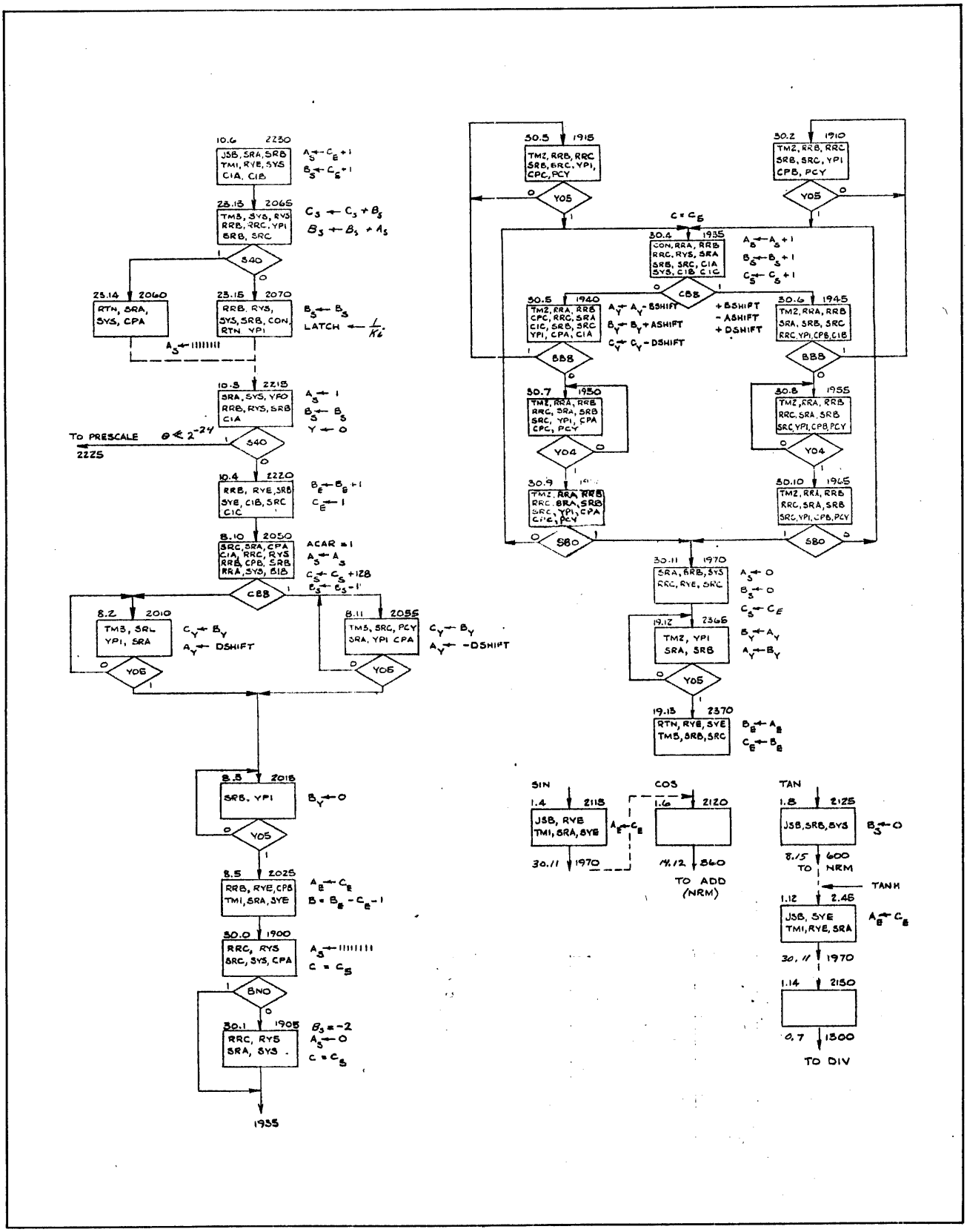


Figure 7-39. Sin Resolver Flowchart

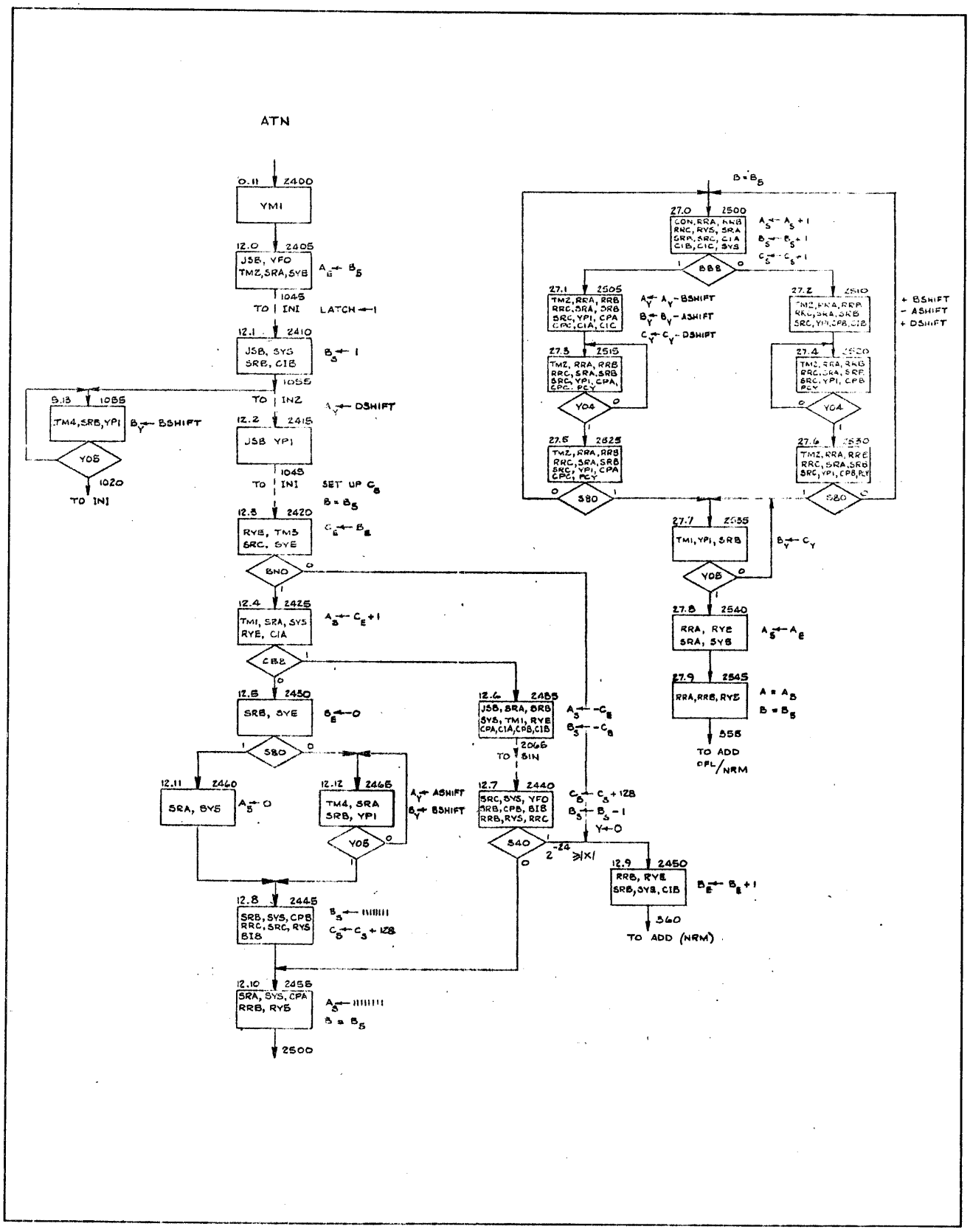


Figure 7-40. Arctan Routine Flowchart

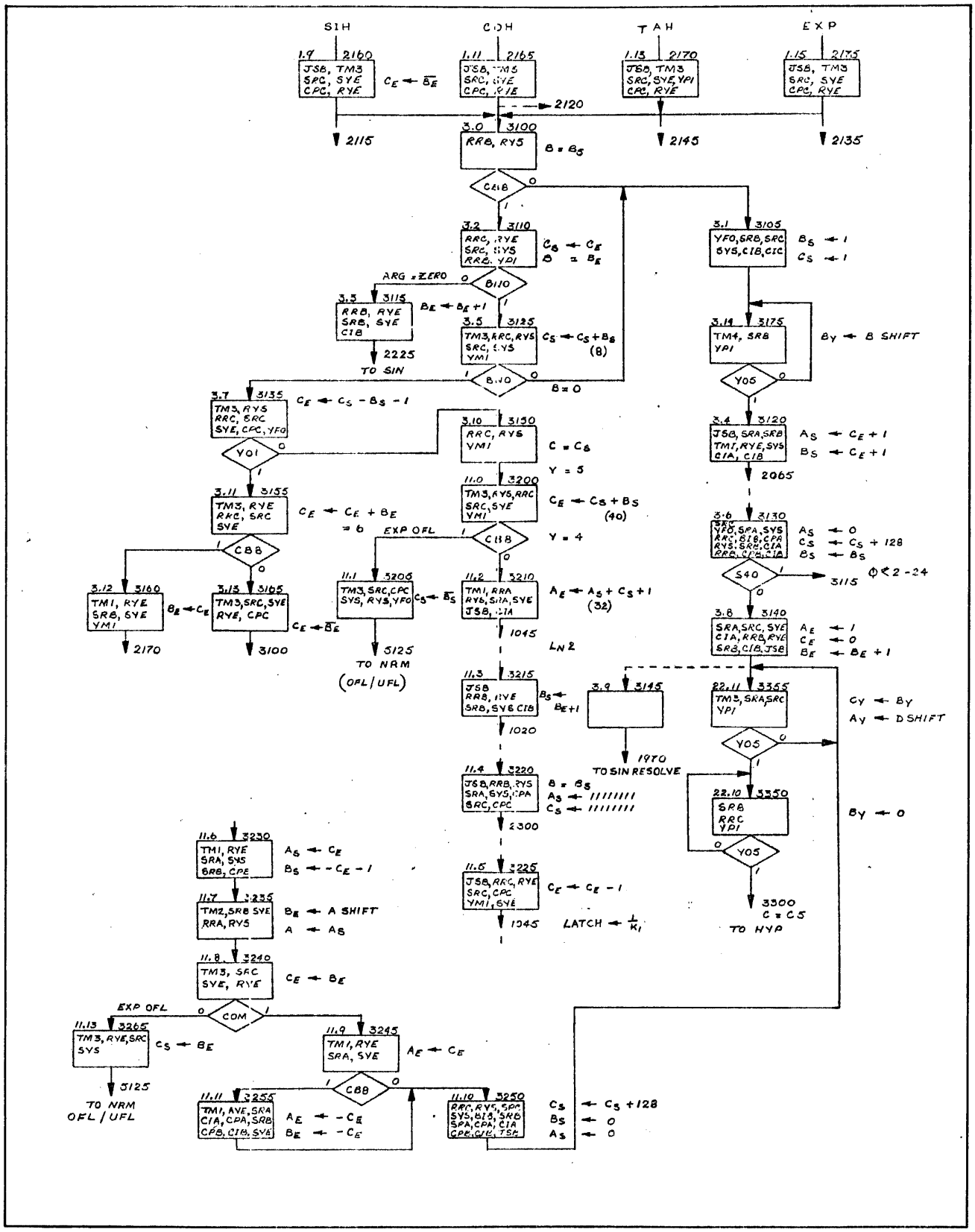


Figure 7-41. Hyperbolic Prescale Flowchart

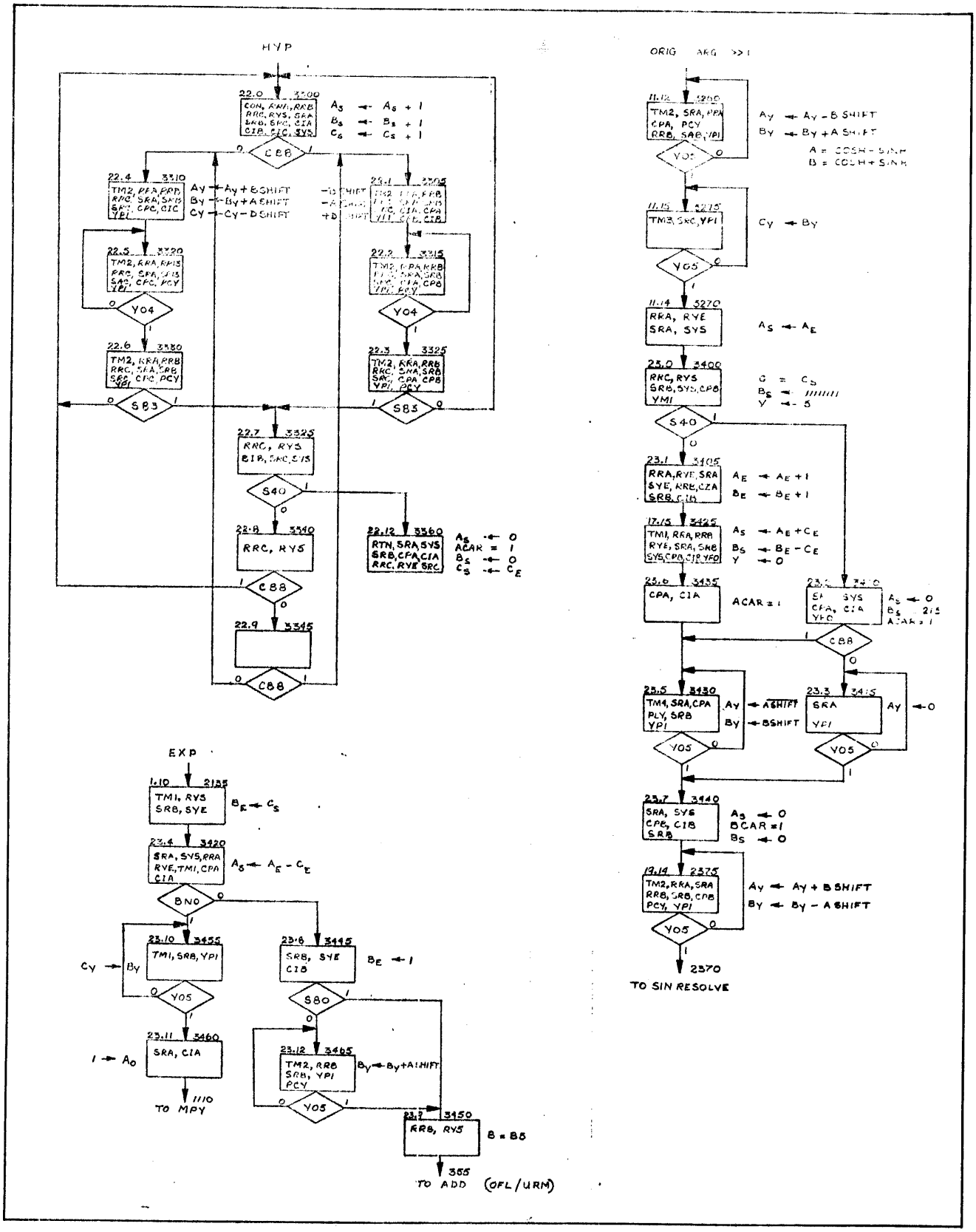


Figure 7-42. Sinh/Cosh Resolve Flowcharts

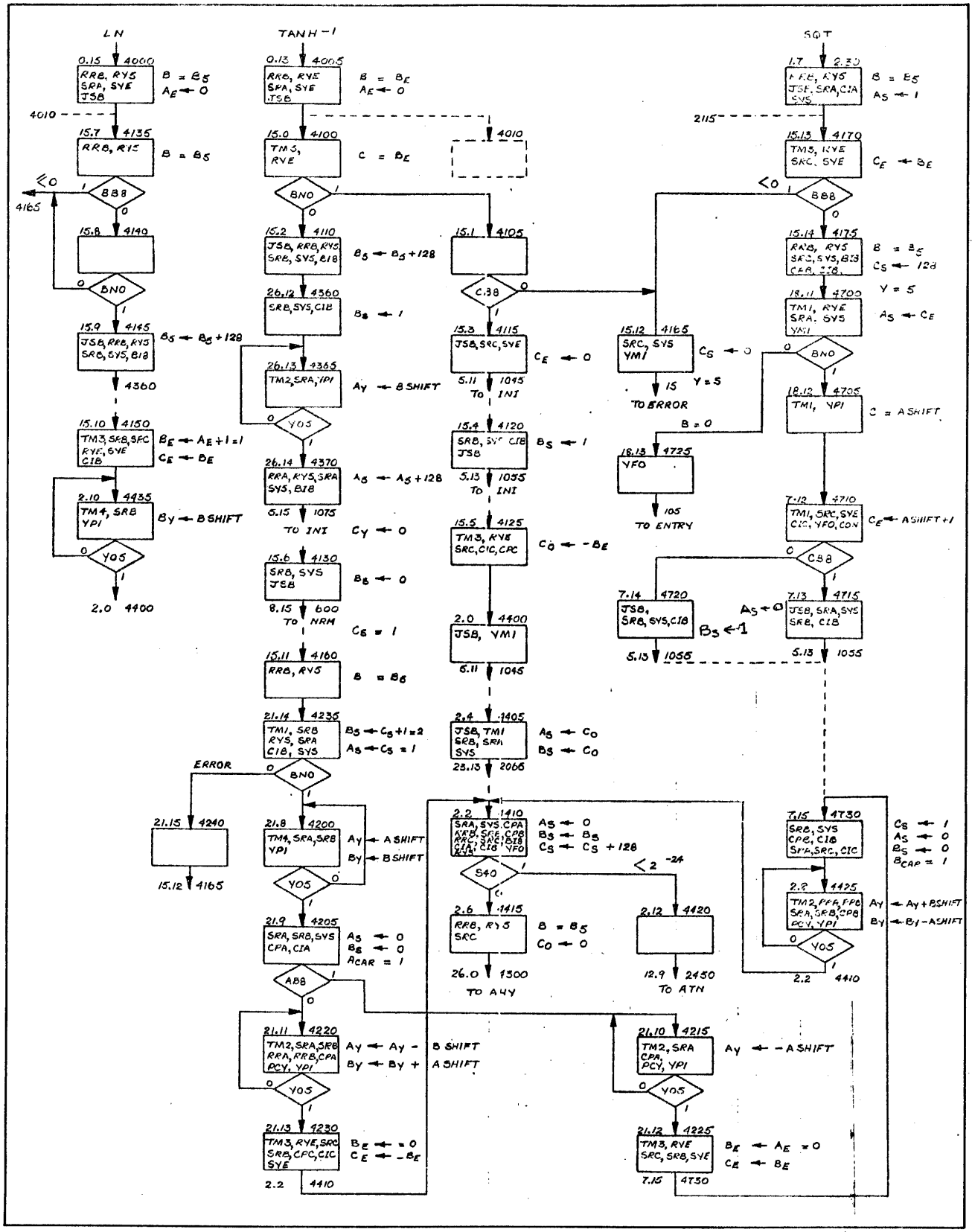


Figure 7-43. Ln/Arctanh/Sqt Prescale Flowcharts

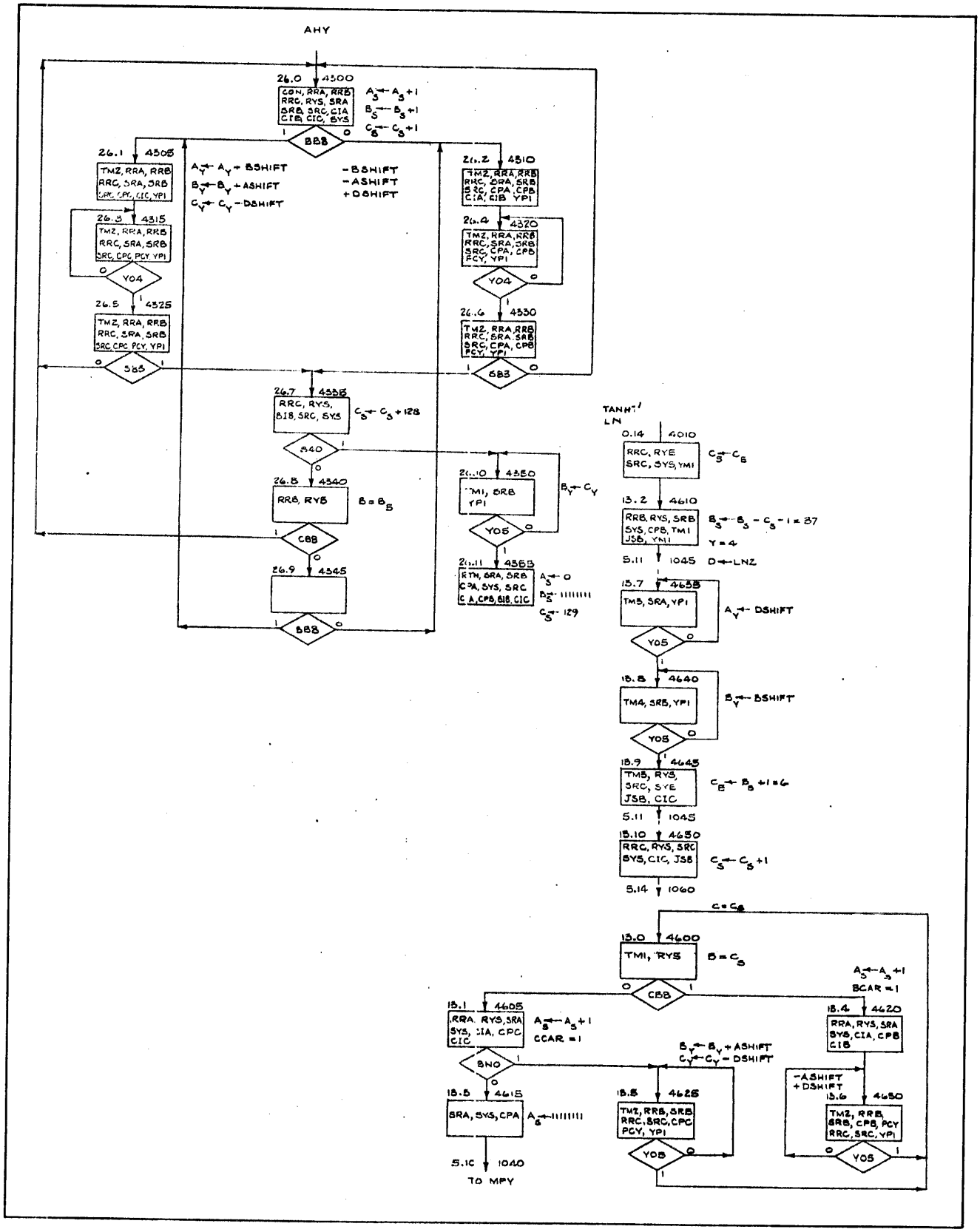


Figure 7-44. Arc Hyper Resolve Flowchart

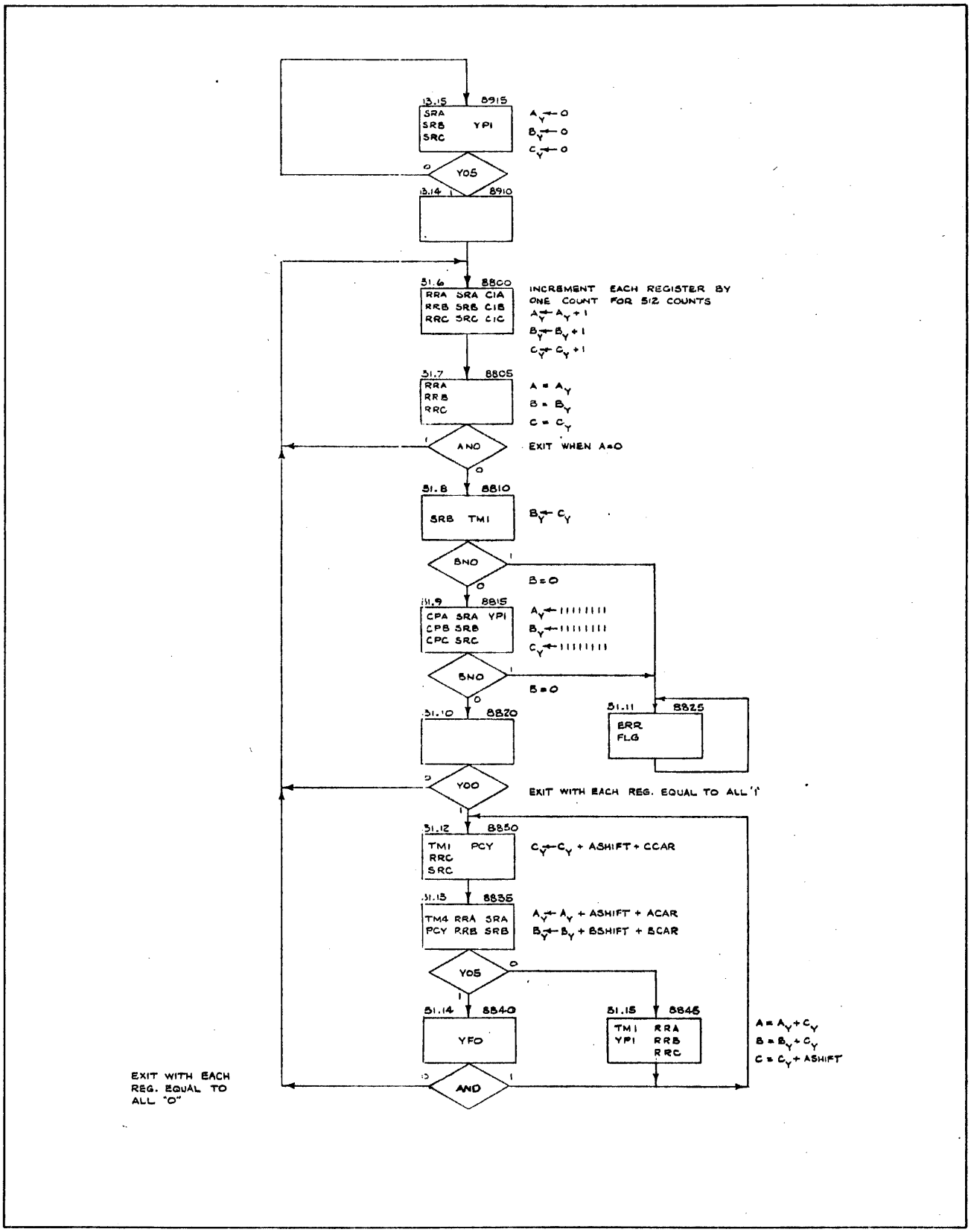


Figure 7-45. Diagnostic Routine Flowchart

Table 7-17. Contents of ROM

ROM ADDRESS		ROM CONTENTS (AS PRINTED OUT)	THIRD WORD AS IN FPP UNIT
OCTAL	PAGE/LINE		
000	00 00	000000 000000 000133	000255
001	00 01	000000 104000 000301	000340
002	00 02	002200 000001 000747	000763
003	00 03	000000 000020 010570	010474
004	00 04	100000 012220 002014	002006
005	00 05	000000 000000 000240	000120
006	00 06	020000 000401 030304	030142
007	00 07	000000 104004 000300	000140
010	00 10	000000 000000 000331	000354
011	00 11	000000 000020 020014	020006
012	00 12	000000 000000 010023	010211
013	00 13	000000 000004 000201	000300
014	00 14	104300 037700 000576	000477
015	00 15	000000 052220 034341	034360
016	00 16	000000 030504 000245	000322
017	00 17	000000 046220 034357	034367
020	01 00	000040 041001 026004	026002
021	01 01	021000 010720 010101	010240
022	01 02	000040 041000 000733	000755
023	01 03	021000 010720 014101	014240
024	01 04	100000 012220 014727	014753
025	01 05	021000 010720 020101	020240
026	01 06	000000 000000 000331	000354
027	01 07	000400 046320 010373	010375
030	01 10	000000 001320 030037	030217
031	01 11	021000 010620 010140	010060
032	01 12	100000 015200 000750	000564
033	01 13	021000 010620 014140	014060
034	01 14	100000 012220 034727	034753
035	01 15	021000 010621 030140	030060
036	01 16	000000 000000 000016	000007
037	01 17	021000 010620 024140	024060
040	02 00	000000 000024 010266	010133
041	02 01	020000 010704 000200	000100
042	02 02	006604 077702 020330	020154
043	02 03	000000 044000 000024	000012
044	02 04	100000 003320 004772	004575
045	02 05	020000 004601 000176	000077
046	02 06	000000 044400 000501	000640
047	02 07	020000 010704 000214	000106
050	02 10	042040 143001 030005	030202
051	02 11	100000 011200 000137	000257
052	02 12	010000 001001 030101	030240
053	02 13	000000 000404 000667	000733
054	02 14	000000 000000 000223	000311
055	02 15	000000 000004 000134	000056
056	02 16	100000 000004 000732	000555
057	02 17	000000 000700 000031	000214

Table 7-17 (Cont'd)

ROM ADDRESS		ROM CONTENTS (AS PRINTED OUT)			THIRD WORD AS IN FPP UNIT
OCTAL	PAGE/LINE				
060	03 00	000000	044000	014044	014022
061	03 01	000300	001702	000174	000076
062	03 02	000000	070701	012152	012065
063	03 03	000200	051200	000113	000245
064	03 04	100600	013320	014772	014575
065	03 05	020000	034704	012056	012027
066	03 06	006604	077702	020007	020203
067	03 07	021000	034602	024127	024253
070	03 10	000600	053620	022726	022553
071	03 11	000000	000000	000727	000753
072	03 12	000000	034004	000141	000260
073	03 13	020000	030600	014271	014334
074	03 14	100000	011204	000072	000035
075	03 15	021000	010600	000140	000060
076	03 16	010000	001001	030311	030344
077	03 17	100000	012301	000234	000116
100	04 00	106000	157302	014050	014024
101	04 01	020000	014700	020112	020045
102	04 02	010000	001001	030106	030043
103	04 03	102200	057000	000344	000162
104	04 04	000000	000022	012522	012451
105	04 05	000000	001000	000133	000255
106	04 06	040000	002004	014350	014164
107	04 07	102000	055302	000220	000110
110	04 10	100000	055201	010073	010235
111	04 11	010000	001000	000224	000112
112	04 12	000004	040001	000226	000113
113	04 13	010040	001001	030171	030274
114	04 14	000000	001001	006253	006325
115	04 15	000100	000700	000031	000214
116	04 16	000004	041021	036040	036020
117	04 17	000000	001001	000230	000114
120	05 00	000200	001320	002276	002137
121	05 01	100000	002401	030044	030022
122	05 02	040000	003001	030130	030054
123	05 03	001500	116300	014053	014225
124	05 04	020000	002001	030236	030117
125	05 05	042200	061401	000254	000126
126	05 06	042040	061401	026320	026150
127	05 07	041040	061401	016164	016072
130	05 10	042040	061401	016164	016072
131	05 11	041040	061401	026057	026227
132	05 12	100000	012202	000340	000160
133	05 13	000000	000700	000267	000333
134	05 14	004000	026300	000246	000123
135	05 15	010000	001001	030251	030324
136	05 16	000000	164072	000000	000000
137	05 17	000000	000401	030367	030373

Table 7-17 (Cont'd)

ROM ADDRESS		ROM CONTENTS (AS PRINTED OUT)	THIRD WORD AS IN FPP UNIT
OCTAL	PAGE/LINE		
140	06 00	000004 055302 004050	004024
141	06 01	001000 000704 000031	000214
142	06 02	004000 002320 012274	012136
143	06 03	000000 144700 032113	032245
144	06 04	104000 012220 004276	004137
145	06 05	001500 116300 006017	006207
146	06 06	000100 110602 000356	000167
147	06 07	042200 061401 000322	000151
150	06 10	041040 061401 026027	026213
151	06 11	042040 061401 026071	026234
152	06 12	100000 001001 030115	030246
153	06 13	041040 161401 016246	016123
154	06 14	042040 161401 016246	016123
155	06 15	000000 034520 034037	034217
156	06 16	000000 024720 036570	036474
157	06 17	100000 012200 000055	000226
160	07 00	000000 152000 020044	020022
161	07 01	000000 030720 012745	012762
162	07 02	000000 003300 006150	006064
163	07 03	040000 051200 000331	000354
164	07 04	100000 000620 000037	000217
165	07 05	000000 000000 000133	000255
166	07 06	000040 000701 000031	000214
167	07 07	100400 112200 002001	002200
170	07 10	000100 070200 002022	002011
171	07 11	104400 153200 010101	010240
172	07 12	001100 000000 010014	010006
173	07 13	040000 000700 000274	000136
174	07 14	100100 000642 014333	014355
175	07 15	000200 003320 036272	036135
176	07 16	000200 001320 036272	036135
177	07 17	002300 003700 000120	000050
200	10 00	100400 016201 014202	014101
201	10 01	000000 000701 000031	000214
202	10 02	020000 002401 030106	030043
203	10 03	000000 001001 030152	030065
204	10 04	001000 000620 014266	014133
205	10 05	102000 052200 000701	000740
206	10 06	000000 051320 016250	016124
207	10 07	004100 046720 020540	020460
210	10 10	020000 030621 022266	022133
211	10 11	000400 112300 000117	000247
212	10 12	006404 177700 014126	014053
213	10 13	024040 002401 030147	030263
214	10 14	000100 034704 024233	024315
215	10 15	100000 016020 034271	034334
216	10 16	000000 000000 000462	000431
217	10 17	000100 046704 000502	000441

Table 7-17 (Cont'd)

ROM ADDRESS		ROM CONTENTS (AS PRINTED OUT)	THIRD WORD AS IN FPP UNIT
OCTAL	PAGE/LINE		
220	11 00	100000 016301 014050	014024
221	11 01	100000 012202 020106	020043
222	11 02	000000 003300 020312	020145
223	11 03	000000 000000 020350	020164
224	11 04	000000 000702 000031	000214
225	11 05	100000 002001 030260	030130
226	11 06	000000 044024 022233	022315
227	11 07	000000 003710 000000	000000
230	11 10	000000 104024 024233	024315
231	11 11	000000 003000 010370	010174
232	11 12	000000 003300 002326	002153
233	11 13	000000 002020 032570	032474
234	11 14	000000 001300 000075	000236
235	11 15	100000 012200 000332	000155
236	11 16	000000 034520 036037	036217
237	11 17	000000 024700 000057	000227
240	12 00	000000 044000 014102	014041
241	12 01	000100 034004 012260	012130
242	12 02	021000 010621 014266	014133
243	12 03	000400 057302 020212	020105
244	12 04	000300 051600 000025	000212
245	12 05	000000 002322 024266	024133
246	12 06	100600 013320 006772	006575
247	12 07	100000 012300 000025	000212
250	12 10	020000 034702 014122	014051
251	12 11	001000 034001 000001	000200
252	12 12	000400 002600 000731	000754
253	12 13	000000 000000 000464	000432
254	12 14	040000 002000 036171	036274
255	12 15	000000 000020 030271	030334
256	12 16	010001 000400 022233	022315
257	12 17	100000 001001 030373	030375
260	13 00	020000 034604 014102	014041
261	13 01	021000 004702 000577	000677
262	13 02	100400 116220 006266	006133
263	13 03	000200 051320 010250	010124
264	13 04	005000 046720 012540	012460
265	13 05	001000 030624 014266	014133
266	13 06	102000 013300 000157	000267
267	13 07	040000 105200 000161	000270
270	13 10	020000 010600 006263	006331
271	13 11	100000 012200 014127	014253
272	13 12	006604 037720 030726	030553
273	13 13	106600 013200 000165	000272
274	13 14	044040 143001 030237	030317
275	13 15	020000 010700 000577	000677
276	13 16	000000 112300 000740	000560
277	13 17	020000 000401 030375	030376

Table 7-17 (Cont'd)

ROM ADDRESS		ROM CONTENTS (AS PRINTED OUT)			THIRD WORD AS IN FPP UNIT
OCTAL	PAGE/LINE				
300	14 00	040000	002222	002266	002133
301	14 01	000200	001320	004272	004135
302	14 02	000000	000021	006266	006133
303	14 03	020000	010600	012051	012224
304	14 04	100400	012300	014254	014126
305	14 05	000000	001200	016227	016313
306	14 06	106600	013320	016772	016575
307	14 07	002004	075702	020123	020251
310	14 10	002004	035700	000225	000312
311	14 11	000200	051200	000331	000354
312	14 12	004000	046300	000541	000660
313	14 13	000000	002100	000221	000310
314	14 14	010000	003001	030221	030310
315	14 15	006000	157300	006337	006357
316	14 16	010000	000032	000000	000000
317	14 17	016000	000032	000000	000000
320	15 00	100000	004000	014050	014024
321	15 01	001500	116300	012152	012065
322	15 02	102000	055324	016266	016133
323	15 03	004000	002300	000264	000132
324	15 04	002600	116300	000255	000326
325	15 05	041040	061401	030240	030120
326	15 06	042040	061401	030300	030140
327	15 07	020000	002001	030360	030170
330	15 10	010000	001001	030023	030211
331	15 11	020100	014620	024266	024133
332	15 12	000100	034720	000274	000136
333	15 13	000000	000040	000366	000173
334	15 14	002604	003302	034271	034334
335	15 15	000000	051000	000463	000631
336	15 16	000000	000000	000755	000766
337	15 17	000000	003401	030375	030376
340	16 00	000000	046300	004303	004341
341	16 01	000000	153000	012344	012162
342	16 02	042200	041000	006206	006103
343	16 03	040000	002300	010250	010124
344	16 04	000000	002300	002142	002061
345	16 05	000000	144000	020334	020156
346	16 06	000000	003000	006065	006232
347	16 07	040000	003001	030360	030170
350	16 10	000000	143200	004305	004342
351	16 11	040040	041001	030071	030234
352	16 12	040040	041001	030127	030253
353	16 13	000000	000000	006271	006334
354	16 14	000000	003322	034037	034217
355	16 15	000000	000722	030743	030761
356	16 16	000000	000000	000133	000255
357	16 17	040000	002302	000377	000377

Table 7-17 (Cont'd)

ROM ADDRESS OCTAL	PAGE/LINE	ROM CONTENTS (AS PRINTED OUT)	THIRD WORD AS IN FPP UNIT
360	17 00	020000 010000 012102	012041
361	17 01	000000 000000 014207	014303
362	17 02	000004 045120 014531	014654
363	17 03	000000 000620 010266	010133
364	17 04	000200 001320 012272	012135
365	17 05	021100 010400 000100	000040
366	17 06	000000 001320 026037	026217
367	17 07	000000 044000 010031	010214
370	17 10	000000 000000 012223	012311
371	17 11	000004 045120 024531	024654
372	17 12	020200 011600 000124	000052
373	17 13	000000 044000 000674	000536
374	17 14	000000 000704 000031	000214
375	17 15	020000 010600 010331	010354
376	17 16	002204 044704 000526	000453
377	17 17	100000 041001 000040	000020
400	20 00	000657 100300 040056	040027
401	20 01	047653 000034 171234	171116
402	20 02	042376 113247 072462	072431
403	20 03	040256 126474 176617	176707
404	20 04	040054 146577 160103	160241
405	20 05	040014 127417 100543	100661
406	20 06	040004 126371 073272	073135
407	20 07	040000 126256 123213	123305
410	20 10	040000 026254 146712	146545
411	20 11	040000 006254 127436	127417
412	20 12	040000 002254 126363	126371
413	20 13	040000 000254 126137	126257
414	20 14	040000 000054 126133	126255
415	20 15	040000 000014 126133	126255
416	20 16	040000 000004 126133	126255
417	20 17	040000 000000 126133	126255
420	21 00	040000 000000 026133	026255
421	21 01	040000 000000 006133	006255
422	21 02	040000 000000 002133	002255
423	21 03	040000 000000 000133	000255
424	21 04	040000 000000 000132	000055
425	21 05	040000 000000 000032	000015
426	21 06	040000 000000 000002	000001
427	21 07	040000 000000 000000	000000
430	21 10	100000 000000 000000	000000
431	21 11	000003 000002 034331	034354
432	21 12	000001 000000 036227	036313
433	21 13	000001 002004 000477	000637
434	21 14	000001 000000 034133	034255
435	21 15	010001 000702 000041	000220
436	21 16	000003 000000 036067	036233
437	21 17	102200 153302 000754	000566

Table 7-17 (Cont'd)

ROM ADDRESS		ROM CONTENTS (AS PRINTED OUT)	THIRD WORD AS IN FPP UNIT
OCTAL	PAGE/LINE		
440	22 00	000000 044004 012152	012065
441	22 01	021100 034700 010106	010043
442	22 02	010000 000000 022222	022111
443	22 03	012000 000000 022022	022011
444	22 04	000000 044004 012112	012045
445	22 05	004100 036701 010316	010147
446	22 06	046000 055300 002015	002206
447	22 07	042000 055300 002017	002207
450	22 10	020000 034742 000030	000014
451	22 11	000000 001001 030065	030232
452	22 12	000000 001200 000575	000676
453	22 13	100000 012104 012271	012334
454	22 14	100000 000001 000370	000174
455	22 15	000000 000002 000133	000255
456	22 16	010000 001001 030337	030357
457	22 17	000004 045100 000745	000762
460	23 00	002400 157300 010142	010061
461	23 01	001100 034000 110112	110045
462	23 02	041040 061401 030100	030040
463	23 03	002200 034000 010212	010105
464	23 04	042040 061401 030200	030100
465	23 05	000000 044000 014316	014147
466	23 06	005000 116700 010142	010061
467	23 07	100000 002001 030360	030170
470	23 10	020200 011304 000564	000472
471	23 11	002000 003332 000000	000000
472	23 12	010000 001004 022123	022251
473	23 13	000000 000400 000027	000213
474	23 14	040000 003001 030233	030315
475	23 15	020000 011630 000000	000000
476	23 16	042040 143001 030333	030355
477	23 17	040000 000404 000337	000357
500	24 00	046672 073324 020537	020657
501	24 01	066754 126613 035703	035741
502	24 02	075346 035777 055017	055207
503	24 03	077256 103544 070203	070301
504	24 04	077652 164413 177661	177730
505	24 05	077752 127223 017330	017154
506	24 06	077772 125351 035720	035550
507	24 07	077776 125256 111673	111735
510	24 10	077777 125252 164574	164476
511	24 11	077777 165252 127051	127224
512	24 12	077777 175252 125323	125351
513	24 13	077777 177252 125137	125257
514	24 14	077777 177652 125127	125253
515	24 15	077777 177752 125127	125253
516	24 16	077777 177772 125127	125253
517	24 17	077777 177776 125127	125253

Table 7-17 (Cont'd)

ROM ADDRESS OCTAL	ROM ADDRESS PAGE/LINE	ROM CONTENTS (AS PRINTED OUT)	THIRD WORD AS IN FPP UNIT
520	25 00	077777 177777 125127	125253
521	25 01	077777 177777 165127	165253
522	25 02	077777 177777 175127	175253
523	25 03	077777 177777 177127	177253
524	25 04	077777 177777 177727	177753
525	25 05	077777 177777 177767	177773
526	25 06	077777 177777 177777	177777
527	25 07	077777 177777 177777	177777
530	25 10	010000 003001 030023	030211
531	25 11	004400 003300 002165	002272
532	25 12	014040 002001 030131	030254
533	25 13	044040 143001 030173	030275
534	25 14	020000 011600 000376	000177
535	25 15	021100 011600 000104	000042
536	25 16	100200 017300 012361	012370
537	25 17	000000 000000 000371	000374
540	26 00	000700 177740 014202	014101
541	26 01	046600 163401 000704	000542
542	26 02	046040 163401 026106	026043
543	26 03	046040 163401 032016	032007
544	26 04	041100 163401 000712	000545
545	26 05	041040 163401 026254	026126
546	26 06	041040 163401 032016	032007
547	26 07	000004 034700 020031	020214
550	26 10	000000 024000 014041	014220
551	26 11	000000 000000 014202	014101
552	26 12	000000 021001 030101	030240
553	26 13	020000 002401 030165	030272
554	26 14	004400 033730 000000	000000
555	26 15	000000 020700 004375	004376
556	26 16	000000 000042 000663	000731
557	26 17	000000 000042 001663	001731
560	27 00	002000 035304 020044	020022
561	27 01	000600 153200 000476	000437
562	27 02	004400 002302 014152	014065
563	27 03	000000 002001 030156	030067
564	27 04	104400 112300 012025	012212
565	27 05	014040 003001 030256	030127
566	27 06	004400 000000 000752	000565
567	27 07	002200 003300 000574	000476
570	27 10	000200 001200 016223	016311
571	27 11	000000 044000 000327	000353
572	27 12	100000 001001 030127	030253
573	27 13	000400 002000 000344	000162
574	27 14	040040 041001 030223	030311
575	27 15	020000 075701 020375	020376
576	27 16	004000 002330 000000	000000
577	27 17	000000 055371 000000	000000

Table 7-17 (Cont'd)

ROM ADDRESS		ROM CONTENTS (AS PRINTED OUT)	THIRD WORD AS IN FPP UNIT
OCTAL	PAGE/LINE		
600	30 00	030625 174226 004340	004160
601	30 01	043117 124752 132030	132014
602	30 02	040542 135752 002242	002121
603	30 03	040126 022162 075167	075273
604	30 04	040025 061053 046655	046726
605	30 05	040005 053042 043171	043274
606	30 06	040001 052542 021151	021264
607	30 07	040000 052526 021110	021044
610	30 10	040000 012525 061104	061042
611	30 11	040000 002525 053104	053042
612	30 12	040000 000525 052704	052542
613	30 13	040000 000125 052654	052526
614	30 14	040000 000025 052652	052525
615	30 15	040000 000005 052652	052525
616	30 16	040000 000001 052652	052525
617	30 17	040000 000000 052652	052525
620	31 00	040000 000000 012652	012525
621	31 01	040000 000000 002652	002525
622	31 02	040000 000000 000652	000525
623	31 03	040000 000000 000252	000125
624	31 04	040000 000000 000052	000025
625	31 05	040000 000000 000012	000005
626	31 06	040000 000000 000002	000001
627	31 07	040000 000000 000000	000000
630	31 10	040000 000000 000000	000000
631	31 11	010000 041000 000465	000632
632	31 12	000040 041000 000467	000633
633	31 13	100000 002104 000471	000634
634	31 14	000000 055304 000473	000635
635	31 15	100000 000601 000475	000636
636	31 16	100000 030632 000000	000000
637	31 17	010001 000404 000135	000256
640	32 00	000700 177740 010102	010041
641	32 01	041100 163401 000507	000643
642	32 02	046600 163401 000511	000644
643	32 03	041040 163401 026152	026065
644	32 04	046040 163401 026214	026106
645	32 05	041040 163401 032016	032007
646	32 06	046040 163401 032016	032007
647	32 07	000004 034700 020025	020212
650	32 10	000000 044000 014041	014220
651	32 11	000000 000000 010102	010041
652	32 12	100000 001001 030127	030253
653	32 13	006504 003730 000000	000000
654	32 14	000200 001300 000533	000655
655	32 15	040000 002001 030275	030336
656	32 16	000004 106100 000276	000137
657	32 17	054271 005773 164321	164350

Table 7-17 (Cont'd)

ROM ADDRESS		ROM CONTENTS (AS PRINTED OUT)	THIRD WORD AS IN FPP UNIT
OCTAL	PAGE/LINE		
660	33 00	000700 177740 010102	010041
661	33 01	045500 163401 000547	000663
662	33 02	042200 163401 000551	000664
663	33 03	045040 163401 026152	026065
664	33 04	042040 163401 026214	026106
665	33 05	045040 163401 016016	016007
666	33 06	042040 163401 016016	016007
667	33 07	100000 001001 030360	030170
670	33 10	000000 112100 000563	000671
671	33 11	000000 144000 000327	000353
672	33 12	000000 034000 010227	010313
673	33 13	100000 055200 014333	014355
674	33 14	100000 055200 014275	014336
675	33 15	020000 010700 006375	006376
676	33 16	000100 003730 000000	000000
677	33 17	000004 034000 000354	000166
700	34 00	062207 166521 010151	010264
701	34 01	073261 116025 103333	103355
702	34 02	076555 153744 131006	131003
703	34 03	077526 165152 130175	130276
704	34 04	077725 067334 131761	131770
705	34 05	077765 053356 122663	122731
706	34 06	077775 052556 166625	166712
707	34 07	077777 052526 167325	167352
710	34 10	077777 152525 067337	067357
711	34 11	077777 172525 053337	053357
712	34 12	077777 176525 052736	052557
713	34 13	077777 177525 052656	052527
714	34 14	077777 177725 052652	052525
715	34 15	077777 177765 052652	052525
716	34 16	077777 177775 052652	052525
717	34 17	077777 177777 052652	052525
720	35 00	077777 177777 152652	152525
721	35 01	077777 177777 172652	172525
722	35 02	077777 177777 176652	176525
723	35 03	077777 177777 177652	177525
724	35 04	077777 177777 177653	177725
725	35 05	077777 177777 177753	177765
726	35 06	077777 177777 177773	177775
727	35 07	077777 177777 177777	177777
730	35 10	077777 177777 177777	177777
731	35 11	020000 002000 000675	000736
732	35 12	000000 000000 000001	000200
733	35 13	100000 001004 022171	022274
734	35 14	000000 000004 000673	000735
735	35 15	002000 055200 000331	000354
736	35 16	020000 001001 030077	030237
737	35 17	000000 041200 000133	000255

Table 7-17 (Cont'd)

ROM ADDRESS		ROM CONTENTS (AS PRINTED OUT)	THIRD WORD AS IN FPP UNIT
OCTAL	PAGE/LINE		
740	36 00	004000 026300 012050	012024
741	36 01	000000 026300 000711	000744
742	36 02	042040 061401 070110	070044
743	36 03	071040 061401 030150	030064
744	36 04	000700 177740 014254	014126
745	36 05	045500 163401 010346	010163
746	36 06	042200 163401 010005	010202
747	36 07	045040 163401 026362	026171
750	36 10	042040 163401 026025	026212
751	36 11	045040 163401 016226	016113
752	36 12	042040 163401 016226	016113
753	36 13	000000 033700 000570	000474
754	36 14	020000 002001 030227	030313
755	36 15	010000 001004 024275	024336
756	36 16	000000 045300 000737	000757
757	36 17	000000 154000 000741	000760
760	37 00	000000 001000 006044	006022
761	37 01	000300 035700 000534	000456
762	37 02	000000 051300 000565	000672
763	37 03	102040 001001 030150	030064
764	37 04	000000 044700 006112	006045
765	37 05	000000 000000 012102	012041
766	37 06	000700 163400 000757	000767
767	37 07	000000 160000 004015	004206
770	37 10	100000 001000 012067	012233
771	37 11	007000 003401 012127	012253
772	37 12	000000 000000 022330	022154
773	37 13	000003 000000 000767	000773
774	37 14	100040 020400 000773	000775
775	37 15	010040 143000 030375	030376
776	37 16	000000 000002 004330	004154
777	37 17	100000 160001 000771	000774

7-18. Mnemonic Listing of ROM Contents

```
0001  1                              D I S - A S S E M B L Y                    DATE :
0002        PAGE.LINE  LABEL      MNEMONICS/DATA                                  QUAL
0003          0. 0       5    BRN SYY RYY                                        TR
0004          0. 1     400    BRN SYY RY5 RRA                                    TR
0005          0. 2     410    YP1 BRN SYY RYY CIB CPB                            TR
0006          0. 3     415    JSB SYY RYY
0007          0. 4     420    JSB SYF SRA RYF TM1
0008          0. 5    4015    BRN SYY RYY                                        TR
0009          0. 6     425    YP1 BRN SYY SRC RYY TM3                            Y0
0010          0. 7    1300    YM1 BRN SYY RY5 RRA                                TR
0011          0. 8     430    BRN SYY RYY                                        TR
0012          0. 9     405    JSB SYY RYY
0013          0.10     435    BRN SYY RYY                                        BR
0014          0.11    2400    YM1 BRN SYY RYY                                    TR
0015          0.12     620    BRN SYS SRC SRB SRA RYS RRC CIC CIB CPA            TR
0016                          TM1
0017          0.13    4005    JSB SYF SRA RYF RRB
0018          0.14    4010    YM1 BRN SYS SRC RYF RRC                            TR
0019          0.15    4000    JSB SYF SRA RY5 RRB
0020          1. 0     605    YP1 BRN SYY SRB RYY RRB PCY                        Y0
0021          1. 1    2100    JSB SYS SRC RYF CPC TM3
0022          1. 2     610    BRN SYY SRB RYY RRB PCY                            TR
0023          1. 3    2105    JSB SYS SRC RYF CPC TM3
0024          1. 4    2115    JSB SYF SRA RYF TM1
0025          1. 5    2110    JSB SYS SRC RYF CPC TM3
0026          1. 6    2120    BRN SYY RYY                                        TR
0027          1. 7    2130    JSB SYS SRA RY5 RRB CIA
0028          1. 8    2125    JSB SYS SRB RYY
0029          1. 9    2160    JSB SYF SRC RYF CPC TM3
0030          1.10    2135    BRN SYF SRB RY5 TM1                                TR
0031          1.11    2165    JSB SYF SRC RYF CPC TM3
0032          1.12    2145    JSB SYF SRA RYF TM1
0033          1.13    2170    YP1 JSB SYF SRC RYF CPC TM3
0034          1.14    2150    BRN SYY RYY                                        TR
0035          1.15    2175    JSB SYF SRC RYF CPC TM3
0036          2. 0    4400    YM1 JSB SYY RYY
0037          2. 1     215    YM1 BRN SYS SRC RYF TM3                            TR
0038          2. 2    4410    YF0 BRN SYS SRC SRB SRA RYS RRC RRB RIB            S4
0039                          CIB CIA CPB CPA
0040          2. 3     205    BRN SYY RY5 RRB                                    TR
0041          2. 4    4405    JSB SYS SRB SRA RYY TM1
0042          2. 5     230    YP1 BRN SYF SRC RY5 TM3                            TR
0043          2. 6    4415    BRN SYY SRC RY5 RRB                                TR
0044          2. 7     225    YM1 BRN SYS SRC RYF TM3                            TR
0045          2. 8    4425    YP1 BRN SYY SRB SRA RYY RRB RRA PCY CPB            Y0
0046                          TM2
0047          2. 9     200    BRN SYF SRB RYF TM1                                TR
0048          2.10    4435    YP1 BRN SYY SRB RYY TM4                            Y0
0049          2.11     235    YM1 BRN SYY SRC RYY                                TR
0050          2.12    4420    BRN SYY RYY                                        TR
0051          2.13     245    YM1 BRN SYY RYY                                    TR
0052          2.14     250    YM1 BRN SYY RYY TM1                                TR
0053          2.15     255    BRN SYS SRC RYY                                    TR
0054          3. 0    3100    BRN SYY RY5 RRB                                    CR
0055          3. 1    3105    YF0 BRN SYS SRC SRB RYY CIC CIB                    TR
0056          3. 2    3110    YP1 BRN SYS SRC RYF RRC RRB                        BN
0057          3. 3    3115    BRN SYF SRB RYF RRB CIB                            TR
0058          3. 4    3120    JSB SYS SRB SRA RYF CIB CIA TM1
0059          3. 5    3125    YM1 BRN SYS SRC RYS RRC TM3                        BN
0060          3. 6    3130    YF0 BRN SYS SRC SRB SRA RYS RRC RRB RIB            S4
0061                          CIB CIA CPB CPA
0062          3. 7    3135    YF0 BRN SYF SRC RYS RRC CPC TM3                    Y0
0063          3. 8    3140    JSB SYF SRC SRB SRA RYF RRB CIB CIA
0064          3. 9    3145    BRN SYY RYY                                        TR
0065          3.10    3150    YM1 BRN SYY RYS RRC                                TR
0066          3.11    3155    BRN SYF SRC RYF RRC TM3                            CR
0067          3.12    3160    YM1 BRN SYF SRB RYF TM1                            TR
0068          3.13    3165    BRN SYF SRC RYF CPC TM3                            TR
0069          3.14    3175    YP1 BRN SYY SRB RYY TM4                            Y0
0070          3.15    2900    YP1 BRN SYS SRA RYF TM1                            TR
0071          4. 0    2600    YF0 BRN SYS SRB SRA RYS RRB RRA CPB CPA            CB
```

7-18. Mnemonic Listing of ROM Contents (Cont'd)

```
0072                          TM1
0073          4. 1    2605    BRN SYS SRC RYS TM3                          S4
0074          4. 2    2620    YP1 BRN SYY SRB RYY TM4                      Y0
0075          4. 3    2625    BRN SYY SRB SRA RYS RRB CIB CPB TM1          TR
0076          4. 4    2610    YF0 JSB SYY RYY
0077          4. 5    2615    BRN SYY SRB RYY                              TR
0078          4. 6    2630    YM1 BRN SYY SRA RYY TM2                      CB
0079          4. 7    2635    YF0 BRN SYS SRB RYS RRB CPB TM1              TR
0080          4. 8    2640    YP1 BRN SYE SRB RYS RRB TM1                  BR
0081          4. 9    2645    BRN SYY SRB RYY TM4                          TR
0082          4.10    2665    YP1 BRN SYY RYY RRB BI8                      TR
0083          4.11    2650    YP1 BRN SYY SRB RYY PCY TM4                  Y0
0084          4.12    2655    YP1 BRN SYY SRB RYY                          C0
0085          4.13    2660    BRN SYS SRC RYY CIC                          TR
0086          4.14    2670    YP1 JSB SYY SRB RYY RRB BI8
0087          4.15    2675    YP1 BRN SYY SRB RYY                          TR
0088          5. 0    1000    JSB SYS SRB RYY CIB
0089          5. 1    1005    YP1 BRN SYY SRC SRA RYY TM1                  Y0
0090          5. 2    1010    YP1 BRN SYY SRB SRA RYY TM2                  Y0
0091          5. 3    1015    BRN SYS SRA RYS RRA CIC CIA CPC              CB
0092          5. 4    1020    YP1 BRN SYY SRA RYY TM3                      Y0
0093          5. 5    1030    YP1 BRN SYY SRC SRB RYY RRC RRB CIB CPB      TR
0094                          TM2
0095          5. 6    1035    YP1 BRN SYY SRC SRB RYY RRC RRB PCY CPB      Y0
0096                          TM2
0097          5. 7    1070    YP1 BRN SYY SRC SRB RYY RRC RRB PCY CPC      S8
0098                          TM2
0099          5. 8    1065    YP1 BRN SYY SRC SRB RYY RRC RRB PCY CPB      S8
0100                          TM2
0101          5. 9    1025    YP1 BRN SYY SRC SRB RYY RRC RRB PCY CPC      Y0
0102                          TM2
0103          5.10    1040    YF0 BRN SYE SRA RYE TM1                      TR
0104          5.11    1045    BRN SYS SRC RYY                              TR
0105          5.12    1050    BRN SYS SRA RYS RRC CPA                      TR
0106          5.13    1055    YP1 BRN SYY SRB RYY TM4                      Y0
0107          5.14    1060    YF0 RTN CON SYY RYS RRC RRB RRA
0108          5.15    1075    YP1 BRN SYY SRC RYY                          Y0
0109          6. 0    1205    YF0 BRN SYS SRB RYS RRB BI8                  AN
0110          6. 1    1210    YM1 BRN SYS SRC RYY CPC                      TR
0111          6. 2    1220    JSB SYS SRA RYY CPA
0112          6. 3    1215    BRN SYS SRC RYS RRB RRA                      S8
0113          6. 4    1235    JSB SYE SRA RYE CPA TM1
0114          6. 5    1240    BRN SYS SRA RYS RRA CIC CIA CPC              C0
0115          6. 6    1245    YF0 BRN SYE SRC RYE RRA CIC                  TR
0116          6. 7    1255    YP1 BRN SYY SRC SRB RYY RRC RRB CIB CPB      TR
0117                          TM2
0118          6. 8    1250    YP1 BRN SYY SRC SRB RYY RRC RRB PCY CPC      Y0
0119                          TM2
0120          6. 9    1260    YP1 BRN SYY SRC SRB RYY RRC RRB PCY CPB      Y0
0121                          TM2
0122          6.10    1265    YP1 BRN SYY SRB RYY TM1                      Y0
0123          6.11    1270    YP1 BRN SYY SRC SRB RYY RRC RRB RRA PCY      S8
0124                          CPC TM2
0125          6.12    1275    YP1 BRN SYY SRC SRB RYY RRC RRB RRA PCY      S8
0126                          CPB TM2
0127          6.13    5825    JSB SYS SRC RYS RRC
0128          6.14    5830    JSB SYS SRC RYS RRC
0129          6.15    5835    BRN SYE SRA RYE TM1                          TR
0130          7. 0    1100    BRN SYY SRA RYE RRB RRA                      S4
0131          7. 1    1105    JSB SYS SRC RYE RRC
0132          7. 2    1110    BRN SYS SRB SRA RYY                          C0
0133          7. 3    1115    BRN SYE SRB RYE RRB TM2                      TR
0134          7. 4    1120    JSB SYE SRC RYY TM1
0135          7. 5    1130    BRN SYY RYY                                  TR
0136          7. 6    1145    YP1 BRN SYS SRC RYY PCY                      TR
0137          7. 7    1150    BRN SYE SRA RYE RRA CIA TM1                  AB
0138          7. 8    1155    BRN SYE RYE RRC RRB CIC                      AB
0139          7. 9    1160    BRN SYE SRB SRA RYE RRB RRA CIA CPA TM1      BR
0140          7.10    1165    BRN SYY RYY CIC CPC                          BR
0141          7.11    1170    BRN SYS SRC RYY TM2                          TR
0142          7.12    4710    YF0 BRN CON SYE SRC RYY CIC TM1              CR
0143          7.13    4715    JSB SYS SRB SRA RYY CIB
```

7-18. Mnemonic Listing of ROM Contents (Cont'd)

```
0144    7.14   4720  JSB SYS SRB RYY CIB
0145    7.15   4730  BRN SYS SRC SRB SRA RYY CIC CIB CPB          TR
0146    8. 0   2000  YP1 BRN SYE SRA RYS CIA TM1                  CB
0147    8. 1   2005  YP1 BRN SYS SRC RYY                          TR
0148    8. 2   2010  YP1 BRN SYY SRC SRA RYY TM3                  Y0
0149    8. 3   2015  YP1 BRN SYY SRB RYY                         Y0
0150    8. 4   2020  JSB SYE SRC RYY CPC
0151    8. 5   2025  BRN SYE SRA RYE RRB CPB TM1                  TR
0152    8. 6   2030  JSB SYS SRB RYE RRB
0153    8. 7   2035  JSB SYS SRC SRA RYS RRB CIC CPA
0154    8. 8   2040  YP1 JSB SYE SRC RYE RRC TM3
0155    8. 9   2045  BRN SYS SRA RYE RRA CIA                      TR
0156    8.10   2050  BRN SYS SRC SRB SRA RYS RRC RRB RRA BIB      CB
0157                 CIA CPB CPA
0158    8.11   2055  YP1 BRN SYY SRC SRA RYY PCY CPA TM3          Y0
0159    8.12     15  YM1 BRN SYS SRC RYS RRC CIC                 Y0
0160    8.13     45  JSB SYY SRA RYS TM1
0161    8.14     55  BRN SYY RYY                                 TR
0162    8.15    600  YM1 BRN SYS SRC SRA RYS RRB CIC             TR
0163    9. 0   5700  YP1 BRN SYS SRA RYS TM1                     CB
0164    9. 1   5705  YF0 BRN SYE SRA RYE TM1                     S4
0165    9. 2   5710  BRN SYS SRB SRA RYY                         S4
0166    9. 3   5715  BRN SYY RYY                                 S4
0167    9. 4   5720  YF0 BRN SYS SRC RYY                         TR
0168    9. 5   5725  YP1 BRN SYY SRA RYY TM1                     Y0
0169    9. 6   5730  YM1 JSB SYY RYS RRB
0170    9. 7   5735  IND SYS SRC SRB SRA RYY
0171    9. 8   5740  YM1 JSB SYY RYS RRA
0172    9. 9   5745  BRN SYY SRB SRA RYY                         BB
0173    9.10   5750  BRN SYS SRB SRA RYY                         AB
0174    9.11   5755  JSB SYY SRA RYY
0175    9.12   5760  BRN SYS SRB RYY                             TR
0176    9.13   5765  BRN SYE SRA RYE TM1                         TR
0177    9.14   5815  JSB SYS SRC RYS RRC
0178    9.15   5820  BRN SYS SRC RYS RRC                         TR
0179   10. 0   2200  BRN SYY RYS RRB                             CB
0180   10. 1   2205  YM1 BRN SYY RYS RRC CIC                     BN
0181   10. 2   2210  YP1 JSB SYE SRC RYE CPC TM3
0182   10. 3   2215  YF0 BRN SYS SRB SRA RYS RRB CIA             S4
0183   10. 4   2220  BRN SYE SRC SRB RYE RRB CIC CIB             TR
0184   10. 5   2225  YF0 JSB SYS SRA RYY
0185   10. 6   2230  JSB SYS SRB SRA RYE CIB CIA TM1
0186   10. 7   2240  BRN SYS SRA RYE TM1                         TR
0187   10. 8   2260  YF0 BRN SYS SRC RYS RRC TM3                 CB
0188   10. 9   2265  YP1 BRN SYY RYS RRC CPC                     TR
0189   10.10   2270  BRN SYE SRC SRA RYY CIA                     TR
0190   10.11     65  BRN SYY RYY                                 TR
0191   10.12    100  BRN SYY SRA RYY TM2                         FN
0192   10.13    105  JSB SYY RYY
0193   10.14    150  BRN SYY SRC RYY FLG TM4                     Y0
0194   10.15    210  YP1 BRN SYY SRB RYY TM1                     Y0
0195   11. 0   3200  YM1 BRN SYE SRC RYS RRC TM3                 CB
0196   11. 1   3205  YF0 BRN SYS SRC RYS CPC TM3                 TR
0197   11. 2   3210  JSB SYE SRA RYS RRA CIA TM1
0198   11. 3   3215  JSB SYS SRB RYE RRB CIB
0199   11. 4   3220  JSB SYS SRC SRA RYS RRB CPC CPA
0200   11. 5   3225  YM1 JSB SYE SRC RYE RRC CPC
0201   11. 6   3230  BRN SYS SRB SRA RYE CPB TM1                 TR
0202   11. 7   3235  BRN SYE SRB RYS RRA TM2                     TR
0203   11. 8   3240  BRN SYE SRC RYE TM3                         CO
0204   11. 9   3245  BRN SYE SRA RYE TM1                         CP
0205   11.10   3250  JSB SYS SRC SRB SRA RYS RRC BIB CIB CIA
0206                 CPB CPA
0207   11.11   3255  BRN SYE SRB SRA RYE CIB CIA CPB CPA TM1     TR
0208   11.12   3260  YP1 BRN SYY SRB SRA RYY RRB RRA PCY CPA     Y0
0209                 TM2
0210   11.13   3265  BRN SYS SRC RYE TM3                         TR
0211   11.14   3270  BRN SYS SRA RYE RRA                         TR
0212   11.15   3275  YP1 BRN SYY SRC RYY TM3                     Y0
0213   12. 0   2405  YF0 JSB SYE SRA RYY TM2
0214   12. 1   2410  JSB SYS SRB RYY CIB
0215   12. 2   2415  YP1 JSB SYY RYY
```

7-18. Mnemonic Listing of ROM Contents (Cont'd)

```
0216    12. 3   2420  BRN SYE SRC RYE TM3                          BN
0217    12. 4   2425  BRN SYS SRA RYE CIA TM1                      CB
0218    12. 5   2430  BRN SYE SRB RYY                             SB
0219    12. 6   2435  JSB SYS SRB SRA RYE CIB CIA CPB CPA TM1
0220    12. 7   2440  YF0 BRN SYS SRC SRB RYS RRC RPB BIB CPB     S4
0221    12. 8   2445  BRN SYS SRC SRB RYS RRC BIB CPB             TR
0222    12. 9   2450  BRN SYE SRB RYE RRB CIB                     TR
0223    12.10   2455  BRN SYS SRA RYS RRB CPA                     TR
0224    12.11   2460  BRN SYS SRA RYY                             TR
0225    12.12   2465  YP1 BRN SYY SRB SRA RYY TM4                 Y0
0226    12.13   5800  BRN SYS SRB SRA RYS RRB RRA CPB CPA         CO
0227    12.14   5805  YF0 RTN SYY RYY TM4
0228    12.15   5810  YF0 RTN SYY RYY CPB CPA TM4
0229    13. 0   4600  BRN SYY RYS TM1                             CB
0230    13. 1   4605  BRN SYS SRA RYS RRA CIC CIA CPC             BN
0231    13. 2   4610  YM1 JSB SYS SRB RYS RRB CPB TM1
0232    13. 3   4615  BRN SYS SRA RYY CPA                         TR
0233    13. 4   4620  BRN SYS SRA RYS RRA CIB CIA CPB             TR
0234    13. 5   4625  YP1 BRN SYY SRC SRB RYY RRC RRB PCY CPC     Y0
0235                  TM2
0236    13. 6   4630  YP1 BRN SYY SRC SRB RYY RPC RRB PCY CPB     Y0
0237                  TM2
0238    13. 7   4635  YP1 BRN SYY SRA RYY TM3                     Y0
0239    13. 8   4640  YP1 BRN SYY SRB RYY TM4                     Y0
0240    13. 9   4645  JSB SYE SRC RYS CIC TM3
0241    13.10   4650  JSB SYS SRC RYS RRC CIC
0242    13.11   1230  BRN CON SYY RYY                             TR
0243    13.12     10  YF0 BRN SYS SRB SRA RYY BIB CIB CIA CPB     OP
0244    13.13     20  BRN SYY SRB RYE RRB                         TR
0245    13.14   8910  BRN SYY RYY                                 TR
0246    13.15   8915  YP1 BRN SYY SRC SRB SRA RYY                 Y0
0247    14. 0    300  BRN SYS SRA RYS RRB                         AN
0248    14. 1    305  BRN SYY SRB SRA RYE RRB RRA                 BN
0249    14. 2    310  BRN SYY SRB RYY RRB CIB CPB TM2             CO
0250    14. 3    315  BRN SYS SRA RYY TM2                         BB
0251    14. 4    320  BRN SYS SRA RYY                             AB
0252    14. 5    325  BRN SYY RYS RRB RRA                         S4
0253    14. 6    330  BRN SYY SRB SRA RYY                         CO
0254    14. 7    335  YP1 BRN SYY SRB SRA RYY TM2                 Y0
0255    14. 8    340  BRN SYE SRB SRA RYY RRB RRA                 AN
0256    14. 9    345  YP1 BRN SYY SRB RYY RRB PCY TM2             Y0
0257    14.10    350  YP1 BRN SYY SRB RYY RPB PCY TM2             Y0
0258    14.11    355  BRN SYY RYY                                 CO
0259    14.12    360  YF0 JSB SYS SRB SRA RYY
0260    14.13    365  YF0 JSB SYS SRC RYY
0261    14.14    370  BRN SYY RYY                                 TR
0262    14.15    555  YF0 BRN SYS SRA RYY TM2                     TR
0263    15. 0   4100  BRN SYY RYE TM3                             BN
0264    15. 1   4105  BRN SYY RYY                                 CB
0265    15. 2   4110  JSB SYS SRB RYS RRB BIB
0266    15. 3   4115  JSB SYE SRC RYY
0267    15. 4   4120  JSB SYS SRB RYY CIB
0268    15. 5   4125  BRN SYY SRC RYE CIC CPC TM3                 TR
0269    15. 6   4130  JSB SYS SRB RYY
0270    15. 7   4135  BRN SYY RYS RRB                             BB
0271    15. 8   4140  BRN SYY RYY                                 BN
0272    15. 9   4145  JSB SYS SRB RYS RRB BIB
0273    15.10   4150  BRN SYE SRC SRB RYE CIB TM3                 TR
0274    15.11   4160  BRN SYY RYS RRB                             TR
0275    15.12   4165  YM1 BRN SYS SRC RYY                         TR
0276    15.13   4170  BRN SYE SRC RYE TM3                         BB
0277    15.14   4175  YM1 BRN SYS SRC RYS RRB BIB CIB CPB         TR
0278    15.15    570  YP1 BRN SYY SRB RYY RRB TM1                 TR
0279    16. 0   9015  0.0000001 10101111 10000000 11000000 01000000
0280    16. 1   8001  0.1001111 10101011 00000000 00011100 11110010
0281    16. 2   8002  0.1000100 11111110 10010110 10100111 01110101
0282    16. 3   8003  0.1000000 10101110 10101101 00111100 11111101
0283    16. 4   8004  0.1000000 00101100 11001101 01111111 11100000
0284    16. 5   8005  0.1000000 00001100 10101111 00001111 10000001
0285    16. 6   8006  0.1000000 00000100 10101100 11111001 01110110
0286    16. 7   8007  0.1000000 00000000 10101100 10101110 10100110
0287    16. 8   8008  0.1000000 00000000 00101100 10101100 11001101
```

7-18. Mnemonic Listing of ROM Contents (Cont'd)

```
0288    16. 9    8009   0.1000000 00000000 00001100 10101100 10101111
0289    16.10    8010   0.1000000 00000000 00000100 10101100 10101100
0290    16.11    8011   0.1000000 00000000 00000000 10101100 10101100
0291    16.12    8012   0.1000000 00000000 00000000 00101100 10101100
0292    16.13    8013   0.1000000 00000000 00000000 00001100 10101100
0293    16.14    8014   0.1000000 00000000 00000000 00000100 10101100
0294    16.15    8015   0.1000000 00000000 00000000 00000000 10101100
0295    17. 0    8016   0.1000000 00000000 00000000 00000000 00101100
0296    17. 1    8017   0.1000000 00000000 00000000 00000000 00001100
0297    17. 2    8018   0.1000000 00000000 00000000 00000000 00000100
0298    17. 3    8019   0.1000000 00000000 00000000 00000000 00000000
0299    17. 4    8020   0.1000000 00000000 00000000 00000000 00000000
0300    17. 5    8021   0.1000000 00000000 00000000 00000000 00000000
0301    17. 6    8022   0.1000000 00000000 00000000 00000000 00000000
0302    17. 7    8023   0.1000000 00000000 00000000 00000000 00000000
0303    17. 8    9020   1.0000000 00000000 00000000 00000000 00000000
0304    17. 9      35   YF0 BRN SYY RYY FLG ERR                          OP
0305    17.10     120   BRN SYY RYY FLG                                  FN
0306    17.11     130   YM1 BRN SYY SRA RYY FLG                          TR
0307    17.12     140   BRN SYY RYY FLG                                  OP
0308    17.13     150   YF0 BRN SYS SRC RYY FLG TM4                      TR
0309    17.14      25   BRN SYY RYY FLG ERR                              FN
0310    17.15    3425   YF0 BRN SYS SRB SRA RYF RRB RRA CIB CPB          TR
0311                    TM1
0312    18. 0     505   YM1 BRN SYY RYS RRB                              BN
0313    18. 1     510   BRN SYS SRC RYS RRC CIC CPC TM3                  BB
0314    18. 2     515   BRN SYY RYY TM4                                  Y0
0315    18. 3     520   BRN SYY RYY CPB TM4                              Y0
0316    18. 4     525   YM1 BRN SYY RYS RRB                              BN
0317    18. 5     530   YP1 BRN SYS SRC SRA RYS RRC CIC CPA              BB
0318    18. 6     535   BRN SYS SRA RYS RRA CPB CPA TM2                  AB
0319    18. 7     540   BRN SYS SRB RYS RRB CPB TM2                      AB
0320    18. 8     545   YF0 BRN CON SYS SRC RYS RRC TM3                  TR
0321    18. 9     560   YP1 BRN SYY SRB RYY                              Y0
0322    18.10     565   BRN SYF SRC SRB RYY                              TR
0323    18.11    4700   YM1 BRN SYS SRA RYF TM1                          BN
0324    18.12    4705   YP1 BRN SYY RYY TM1                              TR
0325    18.13    4725   YF0 BRN SYY RYY                                  TR
0326    18.14    5530   YP1 BRN SYY SRB RYY TM4                          Y0
0327    18.15    5535   BRN SYS SRB RYS RRB BIB                          TR
0328    19. 0    2300   BRN SYS SRB SRA RYS RRB RRA CIA CPB              BB
0329    19. 1    2305   BRN SYY RYS RRC CIC CPC                          BB
0330    19. 2    2310   YP1 BRN SYY SRC SRB RYY RRC RRB PCY CPC          Y0
0331                    TM2
0332    19. 3    2315   BRN SYY RYS RRC CIB CPB                          BB
0333    19. 4    2320   YP1 BRN SYY SRC SRB RYY RRC RRB PCY CPB          Y0
0334                    TM2
0335    19. 5    2325   BRN SYY RYS RRB                                  CB
0336    19. 6    2330   BRN SYS SRC SRA RYS RRA CPC CPA                  BB
0337    19. 7    2335   YP1 BRN SYY SRA RYY TM1                          Y0
0338    19. 8    2340   YM1 BRN SYS SRB RYF CIB TM3                      TR
0339    19. 9    2345   YF0 RTN SYS SRB SRA RYY CPB
0340    19.10    2355   YM1 BRN SYY SRB RYY TM4                          Y0
0341    19.11    2360   BRN SYY SRC RYY                                  TR
0342    19.12    2365   YP1 BRN SYY SRB SRA RYY TM2                      Y0
0343    19.13    2370   RTN SYF SRC SRB RYF TM3
0344    19.14    2375   YP1 BRN SYY SRB SRA RYY RRB RRA PCY CPB          Y0
0345                    TM2
0346    19.15     550   YM1 BRN SYY SRC RYY TM2                          TR
0347    20. 0    7000   0.1001101 10111010 01110110 11010100 00100001
0348    20. 1    7001   0.1101101 11101100 10101101 10001011 00111011
0349    20. 2    7002   0.1111010 11100110 00111011 11111111 01011010
0350    20. 3    7003   0.1111110 10101110 10000111 01100100 01110000
0351    20. 4    7004   0.1111111 10101010 11101001 00001011 11111111
0352    20. 5    7005   0.1111111 11101010 10101110 10010011 00011110
0353    20. 6    7006   0.1111111 11111010 10101010 11101001 00111011
0354    20. 7    7007   0.1111111 11111110 10101010 10101110 10010011
0355    20. 8    7008   0.1111111 11111111 10101010 10101010 11101001
0356    20. 9    7009   0.1111111 11111111 11101010 10101010 10101110
0357    20.10    7010   0.1111111 11111111 11111010 10101010 10101010
0358    20.11    7011   0.1111111 11111111 11111110 10101010 10101010
0359    20.12    7012   0.1111111 11111111 11111111 10101010 10101010
```

7-18. Mnemonic Listing of ROM Contents (Cont'd)

```
0360    20.13   7013   0.1111111 11111111 11111111 11101010 10101010
0361    20.14   7014   0.1111111 11111111 11111111 11111010 10101010
0362    20.15   7015   0.1111111 11111111 11111111 11111110 10101010
0363    21. 0   7016   0.1111111 11111111 11111111 11111111 10101010
0364    21. 1   7017   0.1111111 11111111 11111111 11111111 11101010
0365    21. 2   7018   0.1111111 11111111 11111111 11111111 11111010
0366    21. 3   7019   0.1111111 11111111 11111111 11111111 11111110
0367    21. 4   7020   0.1111111 11111111 11111111 11111111 11111111
0368    21. 5   7021   0.1111111 11111111 11111111 11111111 11111111
0369    21. 6   7022   0.1111111 11111111 11111111 11111111 11111111
0370    21. 7   7023   0.1111111 11111111 11111111 11111111 11111111
0371    21. 8   4200   YP1 BRN SYY SRB SRA RYY TM4                    Y0
0372    21. 9   4205   BRN SYS SRB SRA RYY CIA CPA                    AB
0373    21.10   4215   YP1 BRN SYY SRA RYY PCY CPA TM4                Y0
0374    21.11   4220   YP1 BRN SYY SRB SRA RYY PRB RRA PCY CPA        Y0
0375                   TM2
0376    21.12   4225   BRN SYE SRC SRB RYE TM3                        TR
0377    21.13   4230   BRN SYE SRC SRB RYE CIC CPC TM3                TR
0378    21.14   4235   BRN SYS SRB SRA RYS CIB TM1                    BN
0379    21.15   4240   BRN SYY RYY                                    TR
0380    22. 0   3300   BRN CON SYS SRC SRB SRA RYS RRC RRB RRA        CB
0381                   CIC CIB CIA
0382    22. 1   3305   YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA        TR
0383                   CIB CIA CPB CPA TM2
0384    22. 2   3315   YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA        Y0
0385                   PCY CPB CPA TM2
0386    22. 3   3325   YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA        SB
0387                   PCY CPB CPA TM2
0388    22. 4   3310   YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA        TR
0389                   CIC CPC TM2
0390    22. 5   3320   YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA        Y0
0391                   PCY CPC TM2
0392    22. 6   3330   YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA        SB
0393                   PCY CPC TM2
0394    22. 7   3335   BRN SYS SRC RYS RRC BIB                        S4
0395    22. 8   3340   BRN SYY RY5 RRC                                CB
0396    22. 9   3345   BRN SYY RYY                                    CB
0397    22.10   3350   YP1 BRN SYY SRB RYY RRC                        Y0
0398    22.11   3355   YP1 BRN SYY SRC SRA RYY TM3                    Y0
0399    22.12   3360   RTN SYS SRC SRB SRA RYE RRC CIA CPA
0400    22.13   3365   BRN SYS SRC RYY RRC                            AN
0401    22.14   3370   YF0 BRN CON SYY RYY                            TR
0402    22.15   3375   AD9 YF0 BRN CON SYY RYY                        TR
0403    23. 0   3400   YM1 BRN SYS SRB RYS RRC CPB                    S4
0404    23. 1   3405   BRN SYE SPB SRA RYE RRB RRA CIB CIA            TR
0405    23. 2   3410   YF0 BRN SYS SRA RYY CIA CPA                    CB
0406    23. 3   3415   YP1 BRN SYY SRA RYY                            Y0
0407    23. 4   3420   BRN SYS SRA RYE RRA CIA CPA TM1                BN
0408    23. 5   3430   YP1 BRN SYY SRB SRA RYY PCY CPA TM4            Y0
0409    23. 6   3435   BRN SYY RYY CIA CPA                            TR
0410    23. 7   3440   BRN SYS SRB SRA RYY CIB CPB                    TR
0411    23. 8   3445   BRN SYE SRB RYY CIB                            SB
0412    23. 9   3450   BRN SYY RY5 RRB                                TR
0413    23.10   3455   YP1 BRN SYY SRB RYY TM1                        Y0
0414    23.11   3460   BRN SYY SRA RYY CIA                            TR
0415    23.12   3465   YP1 BRN SYY SRB RYY RRB PCY TM2                Y0
0416    23.13   2065   YP1 BRN SYS SRC SRB RYS RRC RRB TM3            S4
0417    23.14   2060   RTN SYS SRA RYY CPA
0418    23.15   2070   YP1 RTN CON SYS SRB RYS RRB
0419    24. 0   8100   0.0110001 10010101 11111000 10010110 00001000
0420    24. 1   8101   0.1000110 01001111 10101001 11101010 10110100
0421    24. 2   8102   0.1000001 01100010 10111011 11101010 00000100
0422    24. 3   8103   0.1000000 01010110 00100100 01110010 01111010
0423    24. 4   8104   0.1000000 00010101 01100010 00101011 01001101
0424    24. 5   8105   0.1000000 00000101 01010110 00100010 01000110
0425    24. 6   8106   0.1000000 00000001 01010101 01100010 00100010
0426    24. 7   8107   0.1000000 00000000 01010101 01010110 00100010
0427    24. 8   8108   0.1000000 00000000 00010101 01010101 01100010
0428    24. 9   8109   0.1000000 00000000 00000101 01010101 01010110
0429    24.10   8110   0.1000000 00000000 00000001 01010101 01010101
0430    24.11   8111   0.1000000 00000000 00000000 01010101 01010101
0431    24.12   8112   0.1000000 00000000 00000000 00010101 01010101
```

7-18. Mnemonic Listing of ROM Contents (Cont'd)

```
0432    24.13   8113  0.1000000 00000000 00000000 00000101 01010101
0433    24.14   8114  0.1000000 00000000 00000000 00000001 01010101
0434    24.15   8115  0.1000000 00000000 00000000 00000000 01010101
0435    25. 0   8116  0.1000000 00000000 00000000 00000000 00010101
0436    25. 1   8117  0.1000000 00000000 00000000 00000000 00000101
0437    25. 2   8118  0.1000000 00000000 00000000 00000000 00000001
0438    25. 3   8119  0.1000000 00000000 00000000 00000000 00000000
0439    25. 4   8120  0.1000000 00000000 00000000 00000000 00000000
0440    25. 5   8121  0.1000000 00000000 00000000 00000000 00000000
0441    25. 6   8122  0.1000000 00000000 00000000 00000000 00000000
0442    25. 7   8123  0.1000000 00000000 00000000 00000000 00000000
0443    25. 8   8124  0.1000000 00000000 00000000 00000000 00000000
0444    25. 9     30  BRN SYY SRB RYY RRB TM4                       TR
0445    25.10     40  BRN SYY SRB RYY RRB PCY                       TR
0446    25.11     50  YM1 BRN SYS SRA RYY TM1                       TR
0447    25.12     70  YM1 BRN SYS SRB RYS RRB                       TR
0448    25.13     80  YP1 BRN SYE SRC RYY TM1                       TR
0449    25.14     90  YF0 RTN SYE SRC RYE RRC TM1
0450    25.15    140  YM1 BRN SYY SRC RYY FLG TM4                   TR
0451    26. 0   4300  BRN CON SYS SRC SRB SRA RYS RRC RRB RRA       BB
0452                  CIC CIB CIA
0453    26. 1   4305  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       TR
0454                  CIC CPC TM2
0455    26. 2   4310  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       TR
0456                  CIB CIA CPB CPA TM2
0457    26. 3   4315  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       Y0
0458                  PCY CPC TM2
0459    26. 4   4320  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       Y0
0460                  PCY CPB CPA TM2
0461    26. 5   4325  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       SB
0462                  PCY CPC TM2
0463    26. 6   4330  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       SB
0464                  PCY CPB CPA TM2
0465    26. 7   4335  BRN SYS SRC RYS RRC BIB                       S4
0466    26. 8   4340  BRN SYY RYS RRB                               CB
0467    26. 9   4345  BRN SYY RYY                                   BB
0468    26.10   4350  YP1 BRN SYY SRB RYY TM1                       Y0
0469    26.11   4355  RTN SYS SRC SRB SRA RYY BIB CIC CIA CPB
0470                  CPA
0471    26.12   4360  BRN SYS SRB RYY CIB                           TR
0472    26.13   4365  YP1 BRN SYY SRA RYY TM2                       Y0
0473    26.14   4370  BRN SYS SRA RYS RRA BIB                       TR
0474    26.15   9035  0.1011000 10111001 00001011 11111011 11101000
0475    27. 0   2500  BRN CON SYS SRC SRB SRA RYS RRC RRB RRA       BB
0476                  CIC CIB CIA
0477    27. 1   2505  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       TR
0478                  CIC CIA CPC CPA TM2
0479    27. 2   2510  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       TR
0480                  CIB CPB TM2
0481    27. 3   2515  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       Y0
0482                  PCY CPC CPA TM2
0483    27. 4   2520  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       Y0
0484                  PCY CPB TM2
0485    27. 5   2525  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       SB
0486                  PCY CPC CPA TM2
0487    27. 6   2530  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA       SB
0488                  PCY CPB TM2
0489    27. 7   2535  YP1 BRN SYY SRB RYY TM1                       Y0
0490    27. 8   2540  BRN SYS SRA RYE RRA                           TR
0491    27. 9   2545  BRN SYY RYS RRB RRA                           TR
0492    27.10   5100  BRN SYY RYS RRC                               BB
0493    27.11   5105  BRN SYE SRB RYS RRB TM1                       CB
0494    27.12   5110  BRN SYE SRB RYS RRB TM1                       CB
0495    27.13   5115  BRN SYS SRC RYE TM3                           C0
0496    27.14   5120  RTN SYS SRC SRB SRA RYY CIC
0497    27.15   5125  BRN SYY RYS RRC BIB                           TR
0498    28. 0   7100  0.1100100 10000111 11101101 01010001 00010000
0499    28. 1   7101  0.1110110 10110001 10011100 00010101 10000110
0500    28. 2   7102  0.1111101 01101101 11010111 11100100 10110010
0501    28. 3   7103  0.1111111 01010110 11101010 01101010 10110000
0502    28. 4   7104  0.1111111 11010101 01101110 11011100 10110011
0503    28. 5   7105  0.1111111 11110101 01010110 11101110 10100101
```

7-18. Mnemonic Listing of ROM Contents (Cont'd)

```
0504          28. 6    7106  0.1111111 11111101 01010101 01101110 11101101
0505          28. 7    7107  0.1111111 11111111 01010101 01010110 11101110
0506          28. 8    7108  0.1111111 11111111 11010101 01010101 01101110
0507          28. 9    7109  0.1111111 11111110 11110101 01010101 01010110
0508          28.10    7110  0.1111111 11111111 11111101 01010101 01010101
0509          28.11    7111  0.1111111 11111111 11111111 01010101 01010101
0510          28.12    7112  0.1111111 11111111 11111111 11010101 01010101
0511          28.13    7113  0.1111111 11111111 11111111 11110101 01010101
0512          28.14    7114  0.1111111 11111111 11111111 11111101 01010101
0513          28.15    7115  0.1111111 11111111 11111111 11111111 01010101
0514          29. 0    7116  0.1111111 11111111 11111111 11111111 11010101
0515          29. 1    7117  0.1111111 11111111 11111111 11111111 11110101
0516          29. 2    7118  0.1111111 11110111 11111111 11111111 11111101
0517          29. 3    7119  0.1111111 11111111 11111111 11111111 11111111
0518          29. 4    7120  0.1111111 11111111 11111111 11111111 11111111
0519          29. 5    7121  0.1111111 11111111 11111111 11111111 11111111
0520          29. 6    7122  0.1111111 11111111 11111111 11111111 11111111
0521          29. 7    7123  0.1111111 11111111 11111111 11111111 11111111
0522          29. 8    7124  0.1111111 11111111 11111111 11111111 11111111
0523          29. 9    2715  BRN SYY SRA RYY TM3                              TP
0524          29.10    9040  0.0000000 00000000 00000000 00000000 00000000
0525          29.11    2700  YM1 BRN SYY SRB RYY TM1                          Y0
0526          29.12    2705  YM1 BRN SYY RYY                                  TR
0527          29.13    2710  BRN SYE SRB RYS RRB CPB                          TR
0528          29.14    2720  YP1 BRN SYY SRB RYY TM3                          Y0
0529          29.15    2725  BRN SYE SRB RYY RRB                              TR
0530          30. 0    1900  BRN SYS SRA RY5 RRC CPA                          BN
0531          30. 1    1905  BRN SYS SRA RY5 RRC                              TR
0532          30. 2    1910  YP1 BRN SYY SRC SRB RYY RRC RRB PCY CPB          Y0
0533                         TM2
0534          30. 3    1915  YP1 BRN SYY SRC SRB RYY RRC RRB PCY CPC          Y0
0535                         TM2
0536          30. 4    1935  BRN CON SYS SRC SRB SRA RYS RRC RRB RRA          CR
0537                         CIC CIB CIA
0538          30. 5    1940  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA          BR
0539                         CIC CIA CPC CPA TM2
0540          30. 6    1945  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA          BR
0541                         CIB CPB TM2
0542          30. 7    1950  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA          Y0
0543                         PCY CPC CPA TM2
0544          30. 8    1955  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA          Y0
0545                         PCY CPB TM2
0546          30. 9    1960  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA          SR
0547                         PCY CPC CPA TM2
0548          30.10    1965  YP1 BRN SYY SRC SRB SRA RYY RRC RRB RRA          SR
0549                         PCY CPB TM2
0550          30.11    1970  BRN SYS SRC SRB SRA RYE RRC                      TR
0551          30.12    1975  YP1 BRN SYY SRA RYY TM3                          Y0
0552          30.13    5500  YM1 BRN SYY SRB RYY TM4                          Y0
0553          30.14    5505  BRN SYS SRB RYS RRB                              TR
0554          30.15    5510  BRN SYY RYS RRB RRA                              TR
0555          31. 0    5520  BRN SYY SRB RYY                                  CO
0556          31. 1    5525  BRN SYS SRC SRB RYS RRC CIC CIB                  TR
0557          31. 2    5540  BRN SYS SRB RYE RRB                              TR
0558          31. 3    5545  YP1 BRN SYY SRB RYY PCY CPB TM1                  Y0
0559          31. 4    5550  BRN SYS SRC RYS RRB                              CO
0560          31. 5    5555  BRN SYY RYY                                      BN
0561          31. 6    8800  BRN SYY SRC SRB SRA RYY RRC RRB RRA CIC          TR
0562                         CIB CIA
0563          31. 7    8805  BRN SYY RYY RRC RRB RRA                          AN
0564          31. 8    8810  BRN SYY SRB RYY TM1                              BN
0565          31. 9    8815  YP1 BRN SYY SRC SRB SRA RYY CPC CPB CPA          BN
0566          31.10    8820  BRN SYY RYY                                      Y0
0567          31.11    8825  BRN SYY RYY FLG ERR                              TR
0568          31.12    8830  BRN SYY SRC RYY RRC PCY TM1                      TR
0569          31.13    8835  BRN SYY SRB SRA RYY RRB RRA PCY TM4              Y0
0570          31.14    8840  YF0 BRN SYY RYY                                  AN
0571          31.15    8845  YP1 BRN SYY RYY RRC RRB RRA TM1                  TR


**END-OF-TAPE
*
```

Table 7-19. Reference Designations and Abbreviations

REFERENCE DESIGNATIONS

A	= assembly	K	= relay	TB	= terminal board
B	= motor	L	= inductor	TP	= test point
BT	= battery	M	= meter	U	= integrated circuit
C	= capacitor	MC	= microcircuit	V	= vacuum tube, neon bulb, photocell, etc.
CR	= diode	P	= plug connector		
DL	= delay line	Q	= transistor	VR	= voltage regulator
DS	= indicator (lamp)	R	= resistor	W	= cable, jumper
E	= misc hardware	RT	= thermistor	X	= socket
F	= fuse	S	= switch	Y	= crystal
FL	= filter	T	= transformer	Z	= tuned cavity, network
J	= receptacle connector				

ABBREVIATIONS

A	= amperes	gnd	= ground(ed)	ph	= Phillips head
ac	= alternating current	gra	= gray	pk	= peak
ad	= anode	grn	= green	p-p	= peak-to-peak
Al	= aluminum			pt	= point
AR	= as required	H	= henries	PIV	= peak inverse voltage
adj	= adjust	Hg	= mercury	PNP	= positive-negative-positive
Assy	= assembly	hr	= hour(s)	PWV	= peak working voltage
		Hz	= hertz	porc	= porcelain
B	= base	hdw	= hardware	posn	= position(s)
bp	= bandpass	hex	= hexagon, hexagonal	pozi	= pozidrive
bfo	= beat frequency oscillator			ph brz	= phosphor bronze
blk	= black	ID	= inside diameter		
blu	= blue	IF	= intermediate frequency	rf	= radio frequency
brn	= brown	in.	= inch, inches	rdh	= round head
brs	= brass	I/O	= input/output	rmo	= rack mount only
Btu	= British thermal unit	int	= internal	rms	= root-mean-square
bwc	= backward wave oscillator	incl	= include(s)	RWV	= reverse working voltage
Be Cu	= beryllium copper	insul	= insulation, insulated	rect	= rectifier
		impgrg	= impregnated	r/min	= revolutions per minute
C	= collector	incand	= incandescent		
cw	= clockwise			s	= second
ccw	= counterclockwise	k	= kilo (10^3), kilohm	SB	= slow-blow
cer	= ceramic			Se	= selenium
cmo	= cabinet mount only	lp	= low pass	Si	= silicon
com	= common			scr	= silicon-controlled rectifier
crt	= cathode-ray tube	m	= milli (10^{-3})	sil	= silver
CTL	= capacitor-transistor logic	M	= mega (10^6), megohm	sst	= stainless steel
cath	= cathode	My	= Mylar	stl	= steel
cd pl	= cadmium plate	mfr	= manufacturer	spcl	= special
Comp	= composition	mom	= momentary	spdt	= single-pole, double-throw
conn	= connector	mtg	= mounting	spst	= single-pole, single-throw
compl	= complete	misc	= miscellaneous	semicond	= semiconductor
		met ox	= metal oxide		
dc	= direct current	mintr	= miniature	Ta	= tantalum
dr	= drive			td	= time delay
DTL	= diode-transistor logic	n	= nano (10^{-9})	Ti	= titanium
depc	= deposited carbon	nc	= normally closed or no connection	tgl	= toggle
dpdt	= double-pole, double-throw			thd	= thread
dpst	= double-pole, single-throw	Ne	= neon	tol	= tolerance
		no.	= number or normally open	TTL	= transistor-transistor logic
E	= emitter	np	= nickel plated	term	= terminal
ext	= external	NPN	= negative-positive-negative		
encap	= encapsulated	NPO	= negative positive zero (zero temperature coefficient)	U (μ)	= micro (10^{-6})
elctlt	= electrolytic				
		NSR	= not separately replaceable	V	= volt(s)
F	= farads	NRFR	= not recommended for field replacement	var	= variable
FF	= flip-flop			vio	= violet
flh	= flat head	OD	= outside diameter	VDCW	= direct current working volts
flm	= film	OBD	= order by description		
fxd	= fixed	orn	= orange	W	= watts
filh	= fillister head	ovh	= oval head	ww	= wirewound
		oxd	= oxide	wht	= white
G	= giga (10^9)			WIV	= working inverse voltage
Ge	= germanium	p	= pico (10^{-12})		
gl	= glass	PC	= printed circuit	yel	= yellow

Table 7-20
CODE LIST OF MANUFACTURERS

The following code numbers are from the Federal Supply Code for Manufacturers Cataloging Handbooks H4-1 (Name to Code) and H4-2 (Code to Name) and their latest supplements. The date of revision and the date of the supplements used appear at the bottom of each page. Alphabetical codes have been arbitrarily assigned to suppliers not appearing in the H4 Handbooks.

Code No.	Manufacturer	Address
00000	U.S.A. Common	Any supplier of U.S.
00136	McCoy Electronics	Mount Holly Springs, Pa.
00213	Sage Electronics Corp.	Rochester, N.Y.
00287	Cemco Inc.	Danielson, Conn.
00334	Humidial	Colton, Calif.
00348	Microtron Co., Inc.	Valley Stream, N.Y.
00373	Garlock Inc.	Cherry Hill, N.J.
00656	Aerovox Corp.	New Bedford, Mass.
00779	Amp. Inc.	Harrisburg, Pa.
00781	Aircraft Radio Corp.	Boonton, N.J.
00815	Northern Engineering Laboratories, Inc.	Burlington, Wis.
00853	Sangamo Electric Co., Pickens Div.	Pickens, S.C.
00866	Goe Engineering Co.	City of Industry, Cal.
00891	Carl E. Holmes Corp.	Los Angeles, Calif.
00929	Microlab Inc.	Livingston, N.J.
01002	General Electric Co., Capacitor Dept.	Hudson Falls, N.Y.
01009	Alden Products Co.	Brockton, Mass.
01121	Allen Bradley Co.	Milwaukee, Wis.
01255	Litton Industries, Inc.	Beverly Hills, Calif.
01281	TRW Semiconductors, Inc.	Lawndale, Calif.
01295	Texas Instruments, Inc., Transistor Products Div.	Dallas, Texas
01349	The Alliance Mfg. Co.	Alliance, Ohio
01589	Pacific Relays, Inc.	Van Nuys, Calif.
01930	Amerock Corp.	Rockford, Ill.
01961	Pulse Engineering Co.	Santa Clara, Calif.
02114	Ferroxcube Corp. of America	Saugerties, N.Y.
02116	Wheelock Signals, Inc.	Long Branch, N.J.
02286	Cole Rubber and Plastics Inc.	Sunnyvale, Calif.
02660	Amphenol-Borg Electronics Corp.	Chicago, Ill.
02735	Radio Corp. of America, Semiconductor and Materials Div.	Somerville, N.J.
02771	Vocaline Co. of America, Inc.	Old Saybrook, Conn.
02777	Hopkins Engineering Co.	San Fernando, Calif.
03508	G.E. Semiconductor Prod. Dept.	Syracuse, N.Y.
03705	Apex Machine & Tool Co.	Dayton, Ohio
03797	Eldema Corp.	Compton, Calif.
03877	Transitron Electric Corp.	Wakefield, Mass.
03888	Pyrofilm Resistor Co., Inc.	Cedar Knolls, N.J.
03954	Singer Co., Diehl Div. Finderne Plant	Sumerville, N.J.
04009	Arrow, Hart and Hegeman Elect. Co.	Hartford, Conn.
04013	Taurus Corp.	Lambertville, N.J.
04062	Arco Electronic Inc.	Great Neck, N.Y.
04222	Hi-Q Division of Aerovox	Myrtle Beach, S.C.
04354	Precision Paper Tube Co.	Wheeling, Ill.
04404	Dymec Division of Hewlett-Packard Co.	Palo Alto, Calif.
04651	Sylvania Electric Products, Microwave Device Div.	Mountain View, Calif.
04713	Motorola, Inc., Semiconductor Prod. Div.	Phoenix, Arizona
04732	Filtron Co., Inc. Western Div.	Culver City, Calif.
04773	Automatic Electric Co.	Northlake, Ill.
04796	Sequoia Wire Co.	Redwood City, Calif.
04811	Precision Coil Spring Co.	El Monte, Calif.
04870	P.M. Motor Company	Westchester, Ill.
04919	Component Mfg. Service Co.	W. Bridgewater, Mass.
05006	Twentieth Century Plastics, Inc.	Los Angeles, Calif.
05277	Westinghouse Electric Corp. Semi-Conductor Dept.	Youngwood, Pa.
05347	Ultronix, Inc.	San Mateo, Calif.
05397	Union Carbide Corp., Linde Div., Kemet Dept.	Cleveland, Ohio
05593	Illumitronic Engineering Co.	Sunnyvale, Calif.
05616	Cosmo Plastic (c/o Electrical Spec. Co.)	Cleveland, Ohio
05624	Barber Colman Co.	Rockford, Ill.
05728	Tiffen Optical Co.	Roslyn Heights, Long Island, N.Y.
05729	Metro-Tel Corp.	Westbury, N.Y.
05783	Stewart Engineering Co.	Santa Cruz, Calif.
05820	Wakefield Engineering Inc.	Wakefield, Mass.
06004	Bassick Co., Div. of Stewart Warner Corp.	Bridgeport, Conn.
06090	Raychem Corp.	Redwood City, Calif.
06175	Bausch and Lomb Optical Co.	Rochester, N.Y.
06402	E.T.A. Products Co. of America	Chicago, Ill.
06540	Amatom Electronic Hardware Co., Inc.	New Rochelle, N.Y.
06555	Beede Electrical Instrument Co., Inc.	Penacook, N.H.
06666	General Devices Co., Inc.	Indianapolis, Ind.
06751	Semcor Div. Components Inc.	Phoenix, Ariz.
06812	Torrington Mfg. Co., West Div.	Van Nuys, Calif.
06980	Varian Assoc. Eimac Div.	San Carlos, Calif.
07088	Kelvin Electric Co.	Van Nuys, Calif.
07126	Digitran Co.	Pasadena, Calif.
07137	Transistor Electronics Corp.	Minneapolis, Minn.
07138	Westinghouse Electric Corp. Electronic Tube Div.	Elmira, N.Y.
07149	Filmohm Corp.	New York, N.Y.
07233	Cinch-Graphik Co.	City of Industry, Calif.
07261	Avnet Corp.	Culver City, Calif.
07263	Fairchild Camera & Inst. Corp. Semiconductor Div.	Mountain View, Calif.
07322	Minnesota Rubber Co.	Minneapolis, Minn.
07387	Birtcher Corp., The	Monterey Park, Calif.
07397	Sylvania Elect. Prod. Inc., Mt. View Operations	Mountain View, Calif.
07700	Technical Wire Products Inc.	Cranford, N.J.
07910	Continental Device Corp.	Hawthorne, Calif.
07933	Raytheon Mfg. Co., Semiconductor Div.	Mountain View, Calif.
07980	Hewlett-Packard Co., Boonton Radio Div.	Rockaway, N.J.
08145	U.S. Engineering Co.	Los Angeles, Calif.
08289	Blinn, Delbert Co.	Pomona, Calif.
08358	Burgess Battery Co.	Niagara Falls, Ontario, Canada
08524	Deutsch Fastener Corp.	Los Angeles, Calif.
08664	Bristol Co., The	Waterbury, Conn.
08717	Sloan Company	Sun Valley, Calif.
08718	ITT Cannon Electric Inc., Phoenix Div.	Phoenix, Arizona
08792	CBS Electronics Semiconductor Operations, Div of C.B.S. Inc.	Lowell, Mass.
08984	Mel-Rain	Indianapolis, Ind.
09026	Babcock Relays Div.	Costa Mesa, Calif.
09134	Texas Capacitor Co.	Houston, Texas
09145	Tech. Ind. Inc. Atohm Elect.	Burbank, Calif.
09250	Electro Assemblies, Inc.	Chicago, Ill.
09569	Mallory Battery Co. of Canada, Ltd.	Toronto, Ontario, Canada
10214	General Transistor Western Corp.	Los Angeles, Calif.
10411	Ti-Tal, Inc.	Berkeley, Calif.
10646	Carborundum Co.	Niagara Falls, N.Y.
11236	CTS of Berne, Inc.	Berne, Ind.
11237	Chicago Telephone of California, Inc.	So. Pasadena, Calif.
11242	Bay State Electronics Corp.	Waltham, Mass.
11312	Teledyne Inc., Microwave Div.	Palo Alto, Calif.
11314	National Seal	Downey, Calif.
11534	Duncan Electronics Inc.	Costa Mesa, Calif.
11711	General Instrument Corp., Semiconductor Div., Products Group	Newark, N.J.
11717	Imperial Electronic, Inc.	Buena Park, Calif.
11870	Melabs, Inc.	Palo Alto, Calif.
12136	Philadelphia Handle Co.	Camden, N.J.
12361	Grove Mfg. Co., Inc.	Shady Grove, Pa.
12574	Gulton Ind. Inc. Data System Div.	Albuquerque, N.M.
12697	Clarostat Mfg. Co.	Dover, N.H.
12728	Elmar Filter Corp.	W. Haven, Conn.
12859	Nippon Electric Co., Ltd.	Tokyo, Japan
12881	Metex Electronics Corp.	Clark, N.J.
12930	Delta Semiconductor Inc.	Newport Beach, Calif.
12954	Dickson Electronics Corp.	Scottsdale, Arizona
13103	Thermolloy	Dallas, Texas
13396	Telefunken (GmbH)	Hanover, Germany
13835	Midland-Wright Div. of Pacific Industries, Inc.	Kansas City, Kansas
14099	Sem-Tech	Newbury Park, Calif.
14193	Calif. Resistor Corp.	Santa Monica, Calif.
14298	American Components, Inc.	Conshohocken, Pa.
14433	ITT Semiconductor, A Div. of Int. Telephone & Telegraph Corp.	West Palm Beach, Fla.
14493	Hewlett-Packard Company	Loveland, Colo.
14655	Cornell Dubilier Electric Corp.	Newark, N.J.
14674	Corning Glass Works	Corning, N.Y.
14752	Electro Cube Inc.	San Gabriel, Calif.
14960	Williams Mfg. Co.	San Jose, Calif.
15203	Webster Electronics Co.	New York, N.Y.
15287	Scionics Corp.	Northridge, Calif.
15291	Adjustable Bushing Co.	N. Hollywood, Calif.
15558	Micron Electronics	Garden City, Long Island, N.Y.
15566	Amprobe Inst. Corp.	Lynbrook, N.Y.
15631	Cabletronics	Costa Mesa, Calif.
15772	Twentieth Century Coil Spring Co.	Santa Clara, Calif.
15801	Fenwal Elect. Inc.	Framingham, Mass.
15818	Ametco Inc.	Mt. View, Calif.
16037	Spruce Pine Mica Co.	Spruce Pine, N.C.
16179	Omni-Spectra Inc.	Detroit, Ill.
16352	Computer Diode Corp.	Lodi, N.J.
16688	Ideal Prec. Meter Co., Inc. De Jur Meter Div.	Brooklyn, N.Y.
16758	Delco Radio Div. of G.M. Corp.	Kokoma, Ind.
17109	Thermonetics Inc.	Canoga Park, Calif.
17474	Tranex Company	Mountain View, Calif.
17675	Hamlin Metal Products Corp.	Akron, Ohio
17745	Angstrohm Prec. Inc.	No. Hollywood, Calif.
17870	McGraw-Edison Co.	Manchester, N.H.
18042	Power Design Pacific Inc.	Palo Alto, Calif.
18083	Clevite Corp., Semiconductor Div.	Palo Alto, Calif.
18324	Signetics Corp.	Sunnyvale, Calif.
18476	Ty-Car Mfg. Co., Inc.	Holliston, Mass.
18486	TRW Elect. Comp. Div.	Des Plaines, Ill.
18583	Curtis Instrument, Inc.	Mt. Kisco, N.Y.
18873	E.I. DuPont and Co., Inc.	Wilmington, Del.
18911	Durant Mfg. Co.	Milwaukee, Wis.
19315	The Bendix Corp., Navigation & Control Div.	Teterboro, N.J.
19500	Thomas A. Edison Industries, Div. of McGraw-Edison Co.	West Orange, N.J.
19589	Concoa	Baldwin Park, Calif.
19644	LRC Electronics	Horseheads, N.Y.
19701	Electra Mfg. Co.	Independence, Kansas

Table 7-20
CODE LIST OF MANUFACTURERS (Cont'd)

Code No.	Manufacturer	Address
20183	General Atronics Corp.	Philadelphia Pa.
21226	Executone, Inc.	Long Island City, N.Y.
21335	Fafnir Bearing Co., The	New Britain, Conn.
21520	Fansteel Metallurgical Corp.	N. Chicago, Ill.
23783	British Radio Electronics Ltd.	Washington, D.C.
24455	G.E. Lamp Division	Nela Park, Cleveland, Ohio
24655	General Radio Co.	West Concord, Mass.
24681	Memcor Inc., Comp. Div.	Huntington, Ind.
26365	Gries Reproducer Corp.	New Rochelle. N.Y.
26462	Grobet File Co. of America, Inc.	Carlstadt, N.J.
26992	Hamilton Watch Co.	Lancaster, Pa.
28480	Hewlett-Packard Co.	Palo Alto, Calif.
28520	Heyman Mfg. Co.	Kenilworth, N.J.
33173	G.E. Receiving Tube Dept.	Owensboro, Ky.
35434	Lectrohm Inc.	Chicago, Ill.
36196	Stanwyck Coil Products Ltd.	Hawkesbury, Ontario, Canada
36287	Cunningham, W.H. & Hill, Ltd.	Toronto Ontario, Canada
37942	P.R. Mallory & Co. Inc.	Indianapolis, Ind.
39543	Mechanical Industries Prod. Co.	Akron, Ohio
40920	Miniature Precision Bearings, Inc.	Keene, N.H.
42190	Muter Co.	Chicago, Ill.
43990	C.A. Norgren Co.	Englewood, Colo.
44655	Ohmite Mfg. Co.	Skokie, Ill.
46384	Penn Eng. & Mfg. Corp.	Doylestown, Pa.
47904	Polaroid Corp.	Cambridge. Mass.
48620	Precision Thermometer & Inst. Co.	Southampton, Pa.
49956	Microwave & Power Tube Div.	Waltham, Mass.
52090	Rowan Controller Co.	Westminster, Md.
52983	Sanborn Company	Waltham, Mass.
54294	Shallcross Mfg. Co.	Selma, N.C.
55026	Simpson Electric Co.	Chicago, Ill.
55933	Sonotone Corp.	Elmsford, N.Y.
55938	Raytheon Co. Commercial Apparatus & Systems Div.	So. Norwalk, Conn
56137	Spaulding Fibre Co., Inc.	Tonawanda, N.Y.
56289	Sprague Electric Co.	North Adams, Mass.
59446	Telex Corp.	Tulsa, Okla.
59730	Thomas & Betts Co.	Elizabeth, N.J.
60741	Triplett Electrical Inst. Co.	Bluffton, Ohio
61775	Union Switch and Signal, Div. of Westinghouse Air Brake Co	Pittsburgh, Pa.
62119	Universal Electric Co.	Owosso, Mich.
63743	Ward-Leonard Electric Co.	Mt. Vernon, N.Y.
64959	Western Electric Co., Inc	New York, N.Y.
65092	Weston Inst. Inc. Weston-Newark	Newark, N.J.
66295	Wittek Mfg. Co.	Chicago, Ill.
66346	Minnesota Mining & Mfg. Co. Revere Mincom Div.	St. Paul, Minn.
70276	Allen Mfg. Co.	Hartford, Conn.
70309	Allied Control	New York, N.Y.
70318	Allmetal Screw Product Co., Inc.	Garden City, N.Y.
70485	Atlantic India Rubber Works, Inc.	Chicago, Ill
70563	Amperite Co., Inc	Union City. N.J.
70674	ADC Products Inc	Minneapolis, Minn
70903	Belden Mfg. Co.	Chicago, Ill.
70998	Bird Electronic Corp.	Cleveland, Ohio
71002	Birnbach Radio Co.	New York, N.Y.
71041	Boston Gear Works Div. of Murray Co. of Texas	Quincy, Mass.
71218	Bud Radio, Inc.	Willoughby, Ohio
71286	Camloc Fastener Corp.	Paramus, N.J.
71313	Cardwell Condenser Corp.	Lindenhurst L.I., N.Y.
71400	Bussmann Mfg. Div. of McGraw-Edison Co.	St. Louis, Mo.
71436	Chicago Condenser Corp.	Chicago, Ill.
71447	Calif. Spring Co., Inc.	Pico-Rivera, Calif.
71450	CTS Corp.	Elkhart, Ind.
71468	ITT Cannon Electric Inc.	Los Angeles, Calif.
71471	Cinema, Div. Aerovox Corp.	Burbank, Calif.
71482	C.P. Clare & Co.	Chicago, Ill.
71590	Centralab Div. of Globe Union Inc.	Milwaukee, Wis.
71616	Commercial Plastics Co.	Chicago, Ill.
71700	Cornish Wire Co., The	New York, N.Y.
71707	Coto Coil Co., Inc.	Providence, R.I.
71744	Chicago Miniature Lamp Works	Chicago, Ill.
71785	Cinch Mfg. Co., Howard B. Jones Div.	Chicago, Ill.
71984	Dow Corning Corp.	Midland, Mich.
72136	Electro Motive Mfg. Co., Inc.	Willimantic, Conn.
72619	Dialight Corp.	Brooklyn, N.Y.
72656	Indiana General Corp., Electronics Div.	Keasby, N.J.
72699	General Instrument Corp., Cap. Div.	Newark, N.J.
72765	Drake Mfg. Co.	Harwood Heights, Ill.
72825	Hugh H. Eby Inc.	Philadelphia, Pa.
72928	Gudeman Co.	Chicago, Ill.
72964	Robert M. Hadley Co.	Los Angeles, Calif.
72982	Erie Technological Products, Inc.	Erie, Pa.
73061	Hansen Mfg. Co., Inc.	Princeton, Ind.
73076	H.M. Harper Co.	Chicago, Ill.
73138	Helipot Div. of Beckman Inst., Inc.	Fullerton, Calif.
73293	Hughes Products Division of Hughes Aircraft Co.	Newport Beach, Calif.
73445	Amperex Elect Co.	Hicksville, L.I., N.Y.
73506	Bradley Semiconductor Corp.	New Haven, Conn.
73559	Carling Electric, Inc.	Hartford, Conn.
73586	Circle F Mfg. Co.	Trenton, N.J.
73682	George K. Garrett Co., Div. MSL Industries Inc.	Philadelphia, Pa.
73734	Federal Screw Products Inc.	Chicago, Ill.
73743	Fischer Special Mfg. Co.	Cincinnati, Ohio
73793	General Industries Co The	Elyria, Ohio
73846	Goshen Stamping & Tool Co.	Goshen, Ind.
73899	JFD Electronics Corp.	Brooklyn, N.Y.
73905	Jennings Radio Mfg. Corp.	San Jose, Calif.
73957	Groov-Pin Corp.	Ridgefield, N.J.
74276	Signalite Inc.	Neptune, N.J.
74455	J.H. Winns, and Sons	Winchester, Mass.
74861	Industrial Condenser Corp.	Chicago, Ill.
74868	R.F. Products Division of Amphenol-Borg Electronics Corp.	Danbury, Conn.
74970	E.F. Johnson Co.	Waseca, Minn.
75042	International Resistance Co.	Philadelphia, Pa.
75378	CTS Knights Inc	Sandwich, Ill.
75382	Kulka Electric Corporation	Mt. Vernon, N.Y.
75818	Lenz Electric Mfg. Co.	Chicago, Ill.
75915	Littlefuse, Inc	Des Plaines, Ill.
76005	Lord Mfg Co.	Erie, Pa.
76210	C.W. Marwedel	San Francisco, Calif.
76433	General Instrument Corp., Micamold Division	Newark, N.J.
76487	James Millen Mfg. Co., Inc	Malden Mass.
76493	J.W. Miller Co.	Los Angeles, Calif.
76530	Cinch-Monadnock, Div of United Carr Fastener Corp.	San Leandro, Calif.
76545	Mueller Electric Co.	Cleveland, Ohio
76703	National Union	Newark, N.J.
76854	Oak Manufacturing Co.	Crystal Lake, Ill.
77068	The Bendix Corp., Electrodynamics Div.	N. Hollywood, Calif.
77075	Pacific Metals Co.	San Francisco, Calif.
77221	Phanostran Instrument and Electronic Co.	South Pasadena. Calif.
77252	Philadelphia Steel and Wire Corp.	Philadelphia, Pa.
77342	American Machine & Foundry Co. Potter & Brumfield Div.	Princeton, Ind.
77630	TRW Electronic Components Div.	Camden, N.J.
77638	General Instrument Corp., Rectifier Div.	Brooklyn, N.Y.
77764	Resistance Products Co.	Harrisburg, Pa.
77969	Rubbercraft Corp. of Calif.	Torrance, Calif.
78189	Shakeproof Division of Illinois Tool Works	Elgin, Ill.
78283	Signal Indicator Corp.	New York, N.Y.
78290	Struthers-Dunn Inc.	Pitman, N.J.
78452	Thompson-Bremer & Co.	Chicago, Ill.
78471	Tilley Mfg. Co.	San Francisco, Calif.
78488	Stackpole Carbon Co.	St. Marys, Pa.
78493	Standard Thomson Corp.	Waltham, Mass.
78553	Tinnerman Products, Inc	Cleveland, Ohio
78790	Transformer Engineers	San Gabriel, Calif.
78947	Ucinite Co.	Newtonville, Mass.
79136	Waldes Kohinoor Inc.	Long Island City, N.Y.
79142	Veeder Root, Inc.	Hartford, Conn.
79251	Wenco Mfg. Co.	Chicago, Ill.
79727	Continental-Wirt Electronics Corp.	Philadelphia, Pa.
79963	Zierick Mfg. Corp.	New Rochelle, N.Y.
80031	Mepco Division of Sessions Clock Co.	Morristown. N.J.
80120	Schnitzer Alloy Products Co.	Elizabeth, N.J.
80131	Electronic Industries Association. Any brand Tube meeting EIA Standards-Washington, DC.	
80207	Unimax Switch, Div. Maxon Electronics Corp.	Wallingford, Conn.
80223	United Transformer Corp.	New York, N.Y.
80248	Oxford Electric Corp.	Chicago, Ill.
80294	Bourns Inc.	Riverside, Calif.
80411	Acro Div. of Robertshaw Controls Co.	Columbus, Ohio
80486	All Star Products Inc	Defiance, Ohio
80509	Avery Label Co.	Monrovia, Calif.
80583	Hammarlund Co., Inc.	New York, N.Y.
80640	Stevens, Arnold, Co., Inc	Boston, Mass.
81030	International Instruments Inc	Orange, Conn.
81073	Grayhill Co.	LaGrange, Ill.
81095	Triad Transformer Corp	Venice, Calif.
81312	Winchester Elec Div Litton Ind., Inc.	Oakville, Conn.
81349	Military Specification	
81483	International Rectifier Corp.	El Segundo, Calif.
81541	Airpax Electronics, Inc.	Cambridge, Maryland
81860	Barry Controls, Div. Barry Wright Corp.	Watertown, Mass.
82042	Carter Precision Electric Co.	Skokie, Ill.
82047	Sperti Faraday Inc., Copper Hewitt Electric Div.	Hoboken, N.J.
82142	Jeffers Electronics Division of Speer Carbon Co.	Du Bois, Pa.
82170	Fairchild Camera & Inst. Corp. Space & Defense System Div.	Paramus, N.J.
82209	Maguire Industries, Inc.	Greenwich, Conn.
82219	Sylvania Electric Prod. Inc. Electronic Tube Division	Emporium, Pa.
82376	Astron Corp.	East Newark, Harrison, N.J.
82389	Switchcraft, Inc.	Chicago, Ill.
82647	Metals & Controls Inc. Spencer Products	Attleboro, Mass.
82768	Phillips-Advance Control Co.	Joliet, Ill.
82866	Research Products Corp.	Madison, Wis.
82877	Rotron Mfg. Co., Inc.	Woodstock, N.Y.
82893	Vector Electronic Co.	Glendale, Calif.

From: FSC. Handbook Supplements
H4-1 Dated AUGUST 1966
H4-2 Dated NOV. 1962

Table 7-20

CODE LIST OF MANUFACTURERS (Cont'd)

Code No.	Manufacturer	Address
83058	Carr Fastener Co.	Cambridge, Mass.
83086	New Hampshire Ball Bearing, Inc.	Peterborough, N.H.
83125	General Instrument Corp., Capacitor Div.	Darlington, S.C.
83148	ITT Wire and Cable Div.	Los Angeles, Calif.
83186	Victory Eng. Corp.	Springfield, N.J.
83298	Bendix Corp., Red Bank Div.	Red Bank, N.J.
83315	Hubbell Corp	Mundelein, Ill.
83330	Smith, Herman H., Inc.	Brooklyn, N.Y.
83332	Tech Labs	Palisade's Park, N.J.
83385	Central Screw Co.	Chicago, Ill.
83501	Gavitt Wire and Cable Co. Div. of Amerace Corp.	Brookfield, Mass.
83594	Burroughs Corp. Electronic Tube Div.	Plainfield, N.J.
83740	Union Carbide Corp. Consumer Prod. Div.	New York, N.Y.
83777	Model Eng. and Mfg., Inc.	Huntington, Ind.
83821	Loyd Scruggs Co.	Festus, Mo.
83942	Aeronautical Inst. & Radio Co.	Lodi, N.J.
84171	Arco Electronics Inc.	Great Neck, N.Y.
84396	A.J. Glesener Co., Inc.	San Francisco, Calif.
84411	TRW Capacitor Div.	Ogallala, Neb.
84970	Sarkes Tarzian, Inc.	Bloomington, Ind.
85454	Boonton Molding Company	Boonton, N.J.
85471	A.B. Boyd Co.	San Francisco, Calif.
85474	R.M. Bracamonte & Co.	San Francisco, Calif.
85660	Koiled Kords, Inc.	Hamden, Conn.
85911	Seamless Rubber Co.	Chicago, Ill.
86197	Clifton Precision Products Co., Inc.	Clifton Heights, Pa.
86579	Precision Rubber Products Corp.	Dayton, Ohio
86684	Radio Corp. of America, Electronic Comp. & Devices Div.	Harrison, N.J.
87034	Marco Industries	Anaheim, Calif.
87216	Philco Corporation (Lansdale Division)	Lansdale, Pa.
87473	Western Fibrous Glass Products Co.	San Francisco, Calif.
87664	Van Waters & Rogers Inc.	San Francisco, Calif.
87930	Tower Mfg. Corp.	Providence, R.I.
88140	Cutler-Hammer, Inc.	Lincoln, Ill.
88220	Gould-National Batteries, Inc.	St. Paul, Minn.
88698	General Mills, Inc.	Buffalo, N.Y.
89231	Graybar Electric Co.	Oakland, Calif.
89473	G.E. Distributing Corp.	Schenectady, N.Y.
89665	United Transformer Co.	Chicago, Ill.
90179	US Rubber Co., Consumer Ind. & Plastics Prod. Div.	Passaic, N.J.
90970	Bearing Engineering Co.	San Francisco, Calif.
91146	ITT Cannon Elect, Inc., Salem Div.	Salem, Mass.
91260	Connor Spring Mfg. Co.	San Francisco, Calif.
91345	Miller Dial & Nameplate Co.	El Monte, Calif.
91418	Radio Materials Co.	Chicago, Ill.
91506	Augat Inc.	Attleboro, Mass.
91637	Dale Electronics, Inc.	Columbus, Nebr.
91662	Elco Corp.	Willow Grove, Pa.
91737	Gremar Mfg. Co., Inc.	Wakefield, Mass.
91827	K F Development Co.	Redwood City, Calif.
91886	Malco Mfg. Co., Inc.	Chicago, Ill.
91929	Honeywell Inc., Micro Switch Div.	Freeport, Ill.
91961	Nahm-Bros. Spring Co.	Oakland, Calif.
92180	Tru-Connector Corp.	Peabody, Mass.
92367	Elgeet Optical Co. Inc.	Rochester, N.Y.
92607	Tensolite Insulated Wire Co., Inc.	Tarrytown, N.Y.
92702	IMC Magnetics Corp.	Wesbury Long Island, N.Y.
92966	Hudson Lamp Co.	Kearney, N.J.
93332	Sylvania Electric Prod. Inc. Semiconductor Div.	Woburn, Mass.
93369	Robbins & Myers Inc.	Palisades Park, N.J.
93410	Stevens Mfg. Co., Inc.	Mansfield, Ohio
93929	G.V. Controls	Livingston, N.J.
94137	General Cable Corp.	Bayonne, N.J.
94144	Raytheon Co., Comp. Div., Ind. Comp. Operations	Quincy, Mass.
94148	Scientific Electronics Products, Inc.	Loveland, Colo.
94154	Wagner Elect. Corp., Tung-Sol Div.	Newark, N.J.
94197	Curtiss-Wright Corp. Electronics Div.	East Paterson, N.J.
94222	South Chester Corp.	Chester, Pa.
94330	Wire Cloth Products, Inc.	Bellwood, Ill.
94682	Worcester Pressed Aluminum Corp.	Worcester, Mass.
94696	Magnecraft Electric Co.	Chicago, Ill.
95023	George A. Philbrick Researchers, Inc.	Boston, Mass.
95236	Allies Products Corp.,	Dania, Fla.
95238	Continental Connector Corp.	Woodside, N.Y.
95263	Leecraft Mfg. Co., Inc.	Long Island, N.Y.
95265	National Coil Co.	Sheridan, Wyo.
95275	Vitramon, Inc.	Bridgeport, Conn.
95348	Gordos Corp.	Bloomfield, N.J.
95354	Methode Mfg. Co.	Rolling Meadows, Ill.
95566	Arnold Engineering Co.	Marengo, Ill.
95712	Dage Electric Co., Inc.	Franklin, Ind.
95984	Siemon Mfg. Co.	Wayne, Ill.
95987	Weckesser Co.	Chicago, Ill.
96067	Huggins Laboratories	Sunnyvale, Calif.
96095	Hi-Q Div. of Aerovox Corp.	Olean, N.Y.
96256	Thordarson-Meissner Inc.	Mt. Carmel, Ill.
96296	Solar Manufacturing Co.	Los Angeles, Calif.
96330	Carlton Screw Co.	Chicago, Ill.
96341	Microwave Associates, Inc.	Burlington, Mass.
96501	Excel Transformer Co.	Oakland, Calif.
97464	Industrial Retaining Ring Co.	Irvington, N.J
97539	Automatic & Precision Mfg.	Englewood, N.J.
97979	Reon Resistor Corp.	Yonkers, N.Y.
97983	Litton System Inc., Adler-Westrex Commun. Div.	New Rochelle, N.Y.
98141	R-Troncis, Inc.	Jamaica, N.Y.
98159	Rubber Teck, Inc.	Gardena, Calif.
98220	Hewlett-Packard Co., Moseley Div.	Pasadena, Calif.
98278	Microdot, Inc.	So. Pasadena, Calif.
98291	Sealectro Corp.	Mamaroneck, N.Y.
98376	Zero Mfg. Co.	Burbank, Calif.
98731	General Mills Inc., Electronics Div.	Minneapolis, Minn.
98734	Paeco Div. of Hewlett-Packard Co.	Palo Alto, Calif.
98821	North Hills Electronics, Inc.	Glen Cove, N.Y.
98978	International Electronic Research Corp.	Burbank, Calif.
99109	Columbia Technical Corp.	New York, N.Y.
99313	Varian Associates	Palo Alto, Calif.
99378	Atlee Corp.	Winchester, Mass.
99515	Marshall Ind., Capacitor Div.	Monrovia, Calif.
99707	Control Switch Division, Controls Co. of America	El Segundo, Calif.
99800	Delevan Electronics Corp.	East Aurora, N.Y.
99848	Wilco Corporation	Indianapolis, Ind.
99934	Renbrandt, Inc	Boston, Mass.
99942	Hoffman Electronics Corp. Semiconductor Div.	El Monte, Calif.
99957	Technology Instrument Corp. of Calif.	Newbury Park, Calif.

THE FOLLOWING HP VENDORS HAVE NO NUMBER ASSIGNED IN THE LATEST SUPPLEMENT TO THE FEDERAL SUPPLY CODE FOR MANUFACTURERS HANDBOOK.

0000F	Malco Tool and Die	Los Angeles, Calif.
0000Z	Willow Leather Products Corp.	Newark, N.J.
000AB	ETA	England
000BB	Precision Instrument Components Co.	Van Nuys, Calif.
000CS	Hewlett-Packard Co., Colorado Springs	Colorado Springs, Colorado
000MM	Rubber Eng. & Development	Hayward, Calif.
000NN	A "N" D Mfg. Co.	San Jose, Calif.
000QQ	Cooltron	Oakland, Calif.
000WW	California Eastern Lab.	Burlington, Calif.
000YY	S.K. Smith Co.	Los Angeles, Calif.