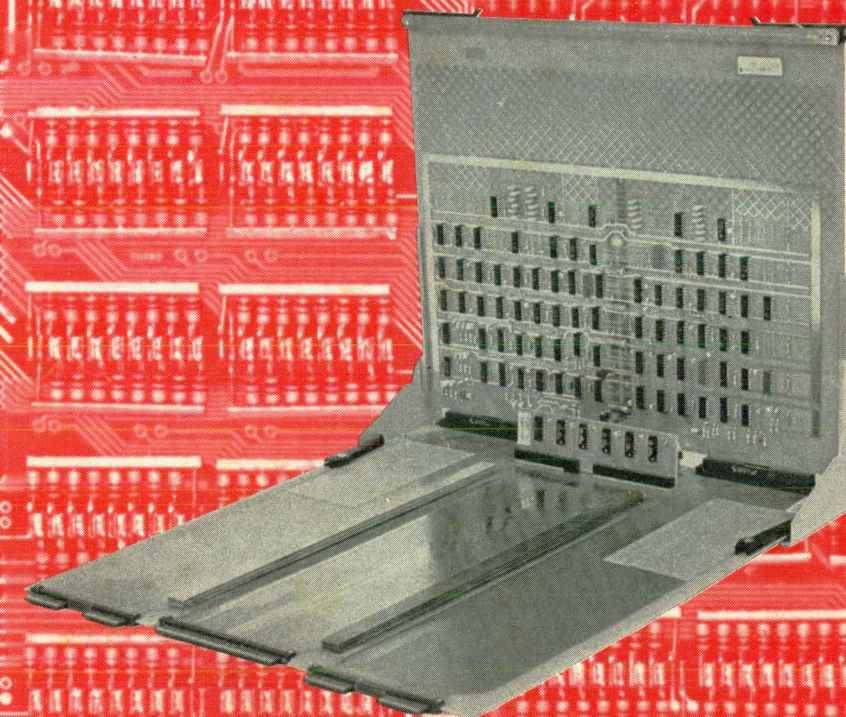
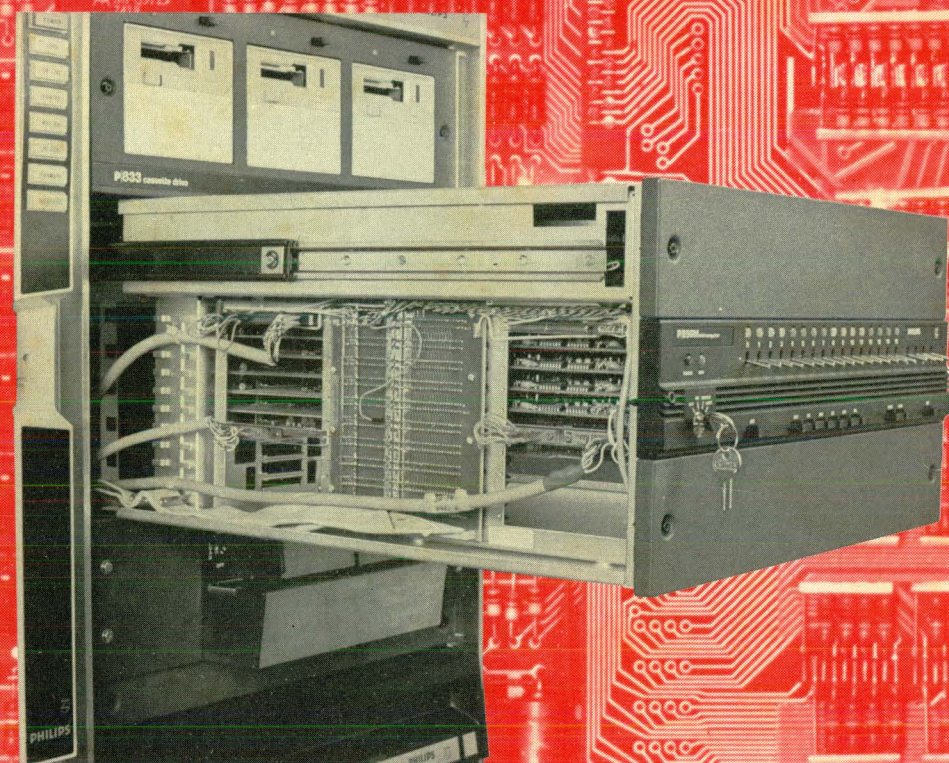




Data
Systems

PHILIPS

P856M/P857M CPU Service Manual



P856M/P857M CPU Service Manual

The information contained in this manual is based on the documentation available at the time of printing, May 1976. However, should any errors or omissions be discovered, or should any user wish to make any suggestions for improving this manual, he is invited to send his comments to:

MANUAL WRITING SMALL COMPUTERS

Philips C.T.I.

4-16, Avenue du Général Leclerc

92260 Fontenay-aux-Roses

France.

A publication of
Philips Data Systems B.V.
Marketing Group Small Computers
Apeldoorn, The Netherlands

Pub. No. 5122 991 26951

MAY 1976

Copyright © by Philips Data Systems B.V. 1976

All rights strictly reserved. Reproduction or issue to third parties in any form whatever is not permitted without written authority from the publisher.

Printed in the Netherlands.

ii	through	viii	A1
1.1	through	1.35	B1
2.1	through	2.71	C1
3.1	through	3.15	D1
4.1	through	4.25	
5.1	through	5.24	
6.1	through	6.6	

<u>Paragraph</u>		<u>Page</u>
I	GENERAL DESCRIPTION	
1.1	System Organization	1-1
1.4	Memory	1-2
1.6	Memory Management Unit (MMU)	1-2
1.7	Floating Point Processor (FPP)	1-2
1.8	I/O Processor (IOP)	1-3
1.9	GP Bus	1-3
1.11	Central Processor Unit	1-3
1.13	CPU Software Registers	1-4
1.15	Bus Controller	1-4
1.16	Interrupt System	1-4
1.17	Internal Interrupts	1-4
1.18	External Interrupts	1-6
1.20	Interrupt Control	1-6
1.21	Interrupt Sequence	1-6
1.22	Interrupt Routine	1-7
1.23	Interrupt Addresses	1-7
1.24	Memory Stack Operation	1-7
1.28	Input/Output Channels	1-7
1.30	Programmed Channel	1-8
1.33	I/O Processor Channel	1-9
1.37	Direct Memory Access Channel	1-10
1.39	Data Communications Channels	1-10
1.40	Clocks	1-11
1.41	CPU Timing Clock	1-11
1.42	Real Time Clock	1-11
1.43	Control Panel	1-11
1.46	Testing	1-11
1.47	Status	1-11
1.48	Data Format	1-11
1.49	Double Precision	1-12
1.51	Logical Data	1-12
1.52	Floating-Point Data	1-12
1.53	Character Handling	1-12
1.54	Operating Modes	1-12
1.56	User Mode	1-13
1.59	Instructions	1-13
1.60	Invalid Instructions	1-13
1.62	Instruction Format	1-13
1.63	Format 0 Instructions (Type T8)	1-13

<u>Paragraph</u>	<u>Page</u>	<u>Paragraph</u>	<u>Page</u>
3.13	3-8	5.19	5-6
3.14	3-8	5.20	5-6
3.15	3-8	5.23	5-6
3.16	3-9	5.24	5-6
3.17	3-9	5.25	5-6
3.18	3-10	5.27	5-10
3.19	3-10	5.28	5-10
3.20	3-10	5.31	5-11
3.21	3-10	5.32	5-11
3.23	3-11	5.33	5-11
3.24	3-11	5.34	5-11
3.25	3-11	5.35	5-11
3.26	3-11	5.39	5-12
3.27	3-11	5.43	5-13
3.28	3-11	5.44	5-13
3.29	3-11	5.45	5-13
3.30	3-11	5.46	5-14
		5.47	5-14
		5.48	5-14
		5.49	5-14
IV MECHANICAL			
4.1	4-1	VI V24 SERIAL CONTROL UNIT	
4.2	4-1	6.1	6-1
4.3	4-1	6.2	6-1
4.4	4-7	6.3	6-2
4.5	4-7	6.4	6-2
4.6	4-7	6.6	6-3
4.7	4-7	6.10	6-4
4.8	4-7	6.11	6-4
4.9	4-7	6.13	6-4
4.10	4-7	6.15	6-5
		6.17	6-5
		6.18	6-5
V POWER SUPPLIES			
5.1	5-1	APPENDIX A INTEGRATED CIRCUITS	
5.2	5-1	APPENDIX B I/O PROCESSOR (IOP)	
5.4	5-2	APPENDIX C MEMORY MANAGEMENT UNIT (MMU)	
5.5	5-3	APPENDIX D FLOATING POINT PROCESSOR (FPP)	
5.6	5-3		
5.7	5-3		
5.8	5-3		
5.9	5-4		
5.10	5-4		
5.14	5-4		
5.15	5-4		

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>	<u>Figure</u>		<u>Page</u>
1.1	P856M/P857M System Block Diagram	1-2	2.3	CPU Bus Control Timing	2-5
1.2	P856M/P857M CPU Block Diagram	1-3	2.4	Data Handling Logic	2-30
1.3A	Interrupts and Breaks	1-4	2.5	Interrupt System	2-43
1.3B	Break Cable Details	1-5	2.6	Sequensor Operating Cycles	2-45
1.4	Programmed Channel Transfers	1-8	2.7	Power On/Off Sequence	2-48
1.5	Multiplexed I/O Processor Channel Transfers	1-9	2.8AA	Instruction Word Logic	2-55
1.6	Direct Memory Access Channel Transfer	1-10	2.8BB	Microcommand Control-Store Logic	2-56
1.7	Operation Terminology, Flow Diagram Key	1-18	2.8CC	Control Store Addressing	2-57
1.8	CPU Operational Flow and Machine State Pointer	1-19	2.8DD	Flag Select	2-58
1.9	Control Panel	1-20	2.8EE	Sequensor (CPU Clock)	2-59
1.10	IPL	1-20	2.8FF	A, D, L Command Logic	2-60
1.11	Addressing	1-21	2.8GG	ALU, M Register	2-61
1.12	Interrupt, Restart, Fault, Trap	1-22	2.8HH	D-Selector, L-Register	2-62
1.13	Store, Fetch, Wait	1-23	2.8JJ	C-Selector, Q-Register	2-63
1.14	Shifting of Move Tables (P857 only)	1-23	2.8KK	S-Register/Counter, Bus Interface	2-64
1.15	OPC 0 (LD, ST)	1-24	2.8LL	A-Bus Selection, IPL, P-Register/Counter	2-65
1.16	OPC 1 (AB)	1-24	2.8MM	Scratchpad	2-66
1.17	OPC 2 (AD, IM)	1-25	2.8NN	PSW, PLR, CR	2-67
1.18	OPC 3 (SU, NG, C2)	1-25	2.8PP	PSW -- GF (General Flip-Flops)	2-68
1.19	OPC 4 (AN, TM, CM, AC, HLT, INH, RIT)	1-26	2.8RR	Start, Reset	2-69
1.20	OPC 5,6 (OR, ENB, LKM, SMD, XR, TNM)	1-26	2.8SS	Interrupt Logic	2-70
1.21	OPC 7 Single Shifts (SL, SR)	1-27	2.8TT	Bus Controller	2-71
1.22	OPC 7 Double Shifts (DL, DR)	1-27	3.1	Control Panels	3-2
1.23	OPC 7 Multiple Load/Store, Table Load/Store (ML/MS, TL/TS)	1-28	3.2	Control Panel Block Diagram	3-5
1.24	OPC 8, 9 I/O Instructions	1-28	3.3A	Control Panel Schematic (Data/Command Half)	3-6
1.25	OPC 8 Multiplication (MU)	1-29	3.3B	Control Panel Schematic (Address Half)	3-7
1.26	OPC 9 Division (DV)	1-29	3.4	CPU Logic Tested by Microdiagnostics	3-9
1.27	OPC 8,9 Floating Point OP, OP Store, Load Store (FA, FS, FM, FD, FSD, FST) P857 only	1-30	3.5	Microdiagnostics Block Diagram	3-12
1.28	OPC 9 Floating Point Conversions (FFL, FFX), P857 only	1-30	3.6	Start and Basic Tests	3-13
1.29	OPC 10 (RF, DA, DAR, DAK, EL, ES)	1-31	3.7	Increment Test, Test-End Display	3-13
1.30	OPC 11 (RB, DS)	1-31	3.8	Logic Tests (T1, T2, T3)	3-14
1.31	OPC 12 (LC, SC, FCR)	1-32	3.9	Memory and CPU/CU Dialogue Tests (Test M)	3-14
1.32	OPC 13 (CC, CW)	1-32	3.10	Run CPU Instruction Tests (Test R)	3-15
1.33	OPC 14 WER and RTN	1-33	3.11	Logic Tested by Basic Tests	3-15
1.34	OPC 14 Execute (EX)	1-33	4.1	P856M/P857M Chassis Configurations	4-1
1.35	OPC 14 Call Function (CF)	1-34	4.2A	M4 Chassis Installation Data	4-2
1.36	OPC 14 MVF and MVSU (P857 only)	1-34	4.2B	M5 Chassis Installation Data	4-3
1.37	OPC 15 (CI, RER, MVB, MVUS)	1-35	4.3	P856M/857M Basic/Extention Chassis Connections	4-4
2.1	P856/857 CPU Block Diagram	2-2	4.4	Circuit-Card Connector Uses	4-5
2.2	Bus Control Block/Timing Diagram	2-5	4.5	CPU Card Layout	4-12
			4.6	General Purpose Card Layout	4-16
			4.7	General Purpose Card Schematic	4-17
			4.8	TAIE Card Schematic	4-18
			4.9	TAIE Card Layout	4-20
			4.10	AIE Card Layout	4-21
			4.11	AIE Card Schematic	4-22
			4.12	Control Panel Layout	4-24

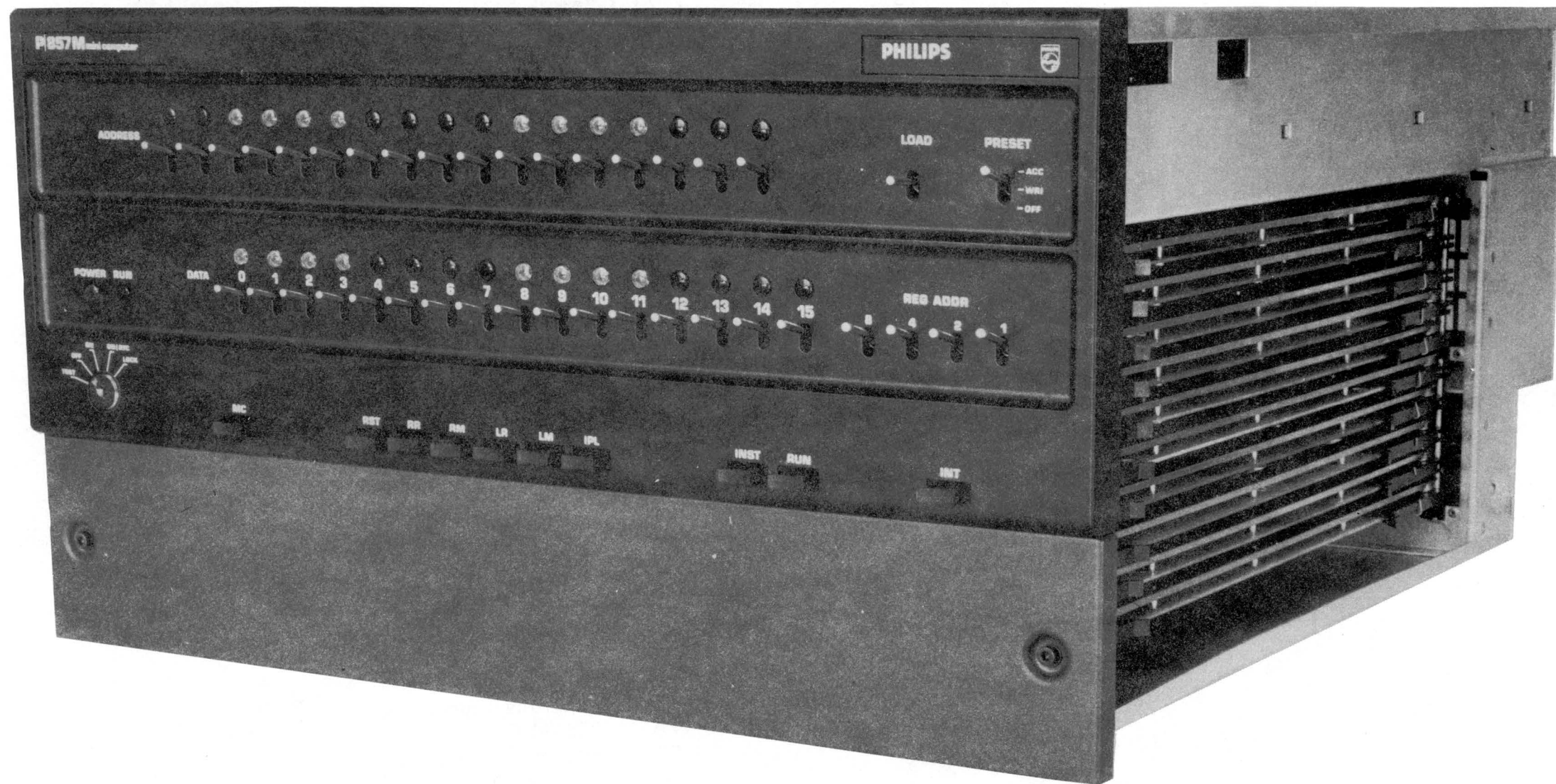
FigurePage

5.1	P857 Power Supply Block Diagram	5-2
5.2	Mains Input Wiring	5-2
5.3 (sheet 1)	Power Supply Inputs and Sequensor Logic	5-7
5.3 (sheet 2)	Power Supply +5V Regulator	5-8
5.3 (sheet 3)	Power Supply +16V Regulator	5-9
5.3 (sheet 4)	Power Supply +18V, -18V, -5V, Supplies	5-10
5.4	Power Sequencing Block Diagram	5-12
5.5	Power Sequence Timing	5-13
5.6	Basic Mounting Box	5-20
5.7	Power Supply Assembly Locations	5-21
5.8	Power Supply Circuit Cards	5-22
5.9	Heat-Sink Assembly	5-23
5.10	Detail of Key and Control-Panel Socket	5-24
6.1	Serial Data Format	6-2
6.2	V24-CU Operational States	6-3
6.3	V24 Serial CU Block Diagram	6-4
6.4	V24 Serial Control Unit Detailed Logic Diagram	6-6

LIST OF TABLES

TablePage

1.1	P856/857 Instruction List	1-14
1.2	Addressing Types	1-16
2.1	Bus Timing Signals	2-6
2.2	Microinstruction Command Bits	2-8
2.3	General Field Command Bit Codes	2-13
2.4A	Control-ROM Microinstruction Listing P857	2-14
2.4B	Control-ROM Microinstruction Listing P856	2-17
2.5A	Control-ROM Binary Content P857	2-20
2.5B	Control-ROM Binary Content P856	2-23
2.6	Instruction Decoder PLA	2-26
2.7	PLA ROM Map	2-28
2.8	ADL Command ROM	2-31
2.9	A-Bus Selection	2-36
2.10	List of Bootstrap Loaded in Memory	2-37
2.11	CR Input Conditions	2-40
2.12	Sequensor T6 Generation	2-47
2.13	Resets and Clears	2-50
2.14	CPU Signal LIST	2-51
2.15	Microinstruction Address Code	2-54
3.1	Control Panel Switches and Lamps	3-3
3.2	Control Panel Interface	3-5
4.1	GP Bus Connector IOM, IOB	4-6
4.2	CPU-A, Connector 1 (V24 CU)	4-8
4.3	Connector 3 (CPU, Mem, IOP, CU)	4-8
4.4	CPU-A Connector-5	4-9
4.5	Control Panel Connector	4-9
4.6	IOP Connectors 4, 5 (Break)	4-10
4.7	Extention Connectors AIE/TAIE	4-10
4.8	Interface Signal List	4-11
4.9A	P856M CPU Parts List	4-13
4.9B	P857M CPU Parts List	4-14
4.10	General Purpose Card Parts List	4-18
4.11	TAIE Parts List	4-20
4.12	AIE Parts List	4-21
5.1A	M4 Parts List Guide	5-15
5.1B	M5 Parts List Guide	5-16
5.1C	Regulator Card P0 Parts List	5-17
5.1D	RST Card P1 Parts List	5-18
5.1E	Sub-Chassis Assembly Parts List	5-19
5.1F	Heat-Sink Assembly Parts List	5-19
5.1G	Heat-Sink Sub-Assembly Parts List	5-19



SECTION I

GENERAL DESCRIPTION

1.1 SYSTEM ORGANIZATION

The P856M/P857M systems consist of a basic central processor unit (CPU) and various independent elements interconnected via a General Purpose Bus. The CPU is contained on a single printed-circuit card. The same card includes a Serial Control unit for connecting an operator's device. Some of the other system elements are: memory modules, control units, input/output processor (IOP), the P857 options floating-point processor (FPP) and memory management unit (MMU).

1.2 A block diagram of the P856M/P857M systems is shown in Figure 1-1. All system elements are interconnected via the GP Bus. The P852M system elements are plug compatible with the P856M/P857M systems and may be connected via the same GP Bus. P850/P855 control units may also be used with the 856/857 system; the 850/855 cards are mounted in their own chassis and connected to the GP Bus via a bus converter unit. Chassis information and wiring is included in Section IV.

1.3 The P857M and P856M systems use the same basic logic design and both systems are based on a single-card CPU and operator's control unit. The two systems use different microprogram control, and the P857M performs an expanded number of functions, including operations with the FPP and MMU options. The specific logic differences are noted throughout the Logic description, Section II. Also, the P857M and P856M systems have different standard control panels as shown in Section III.

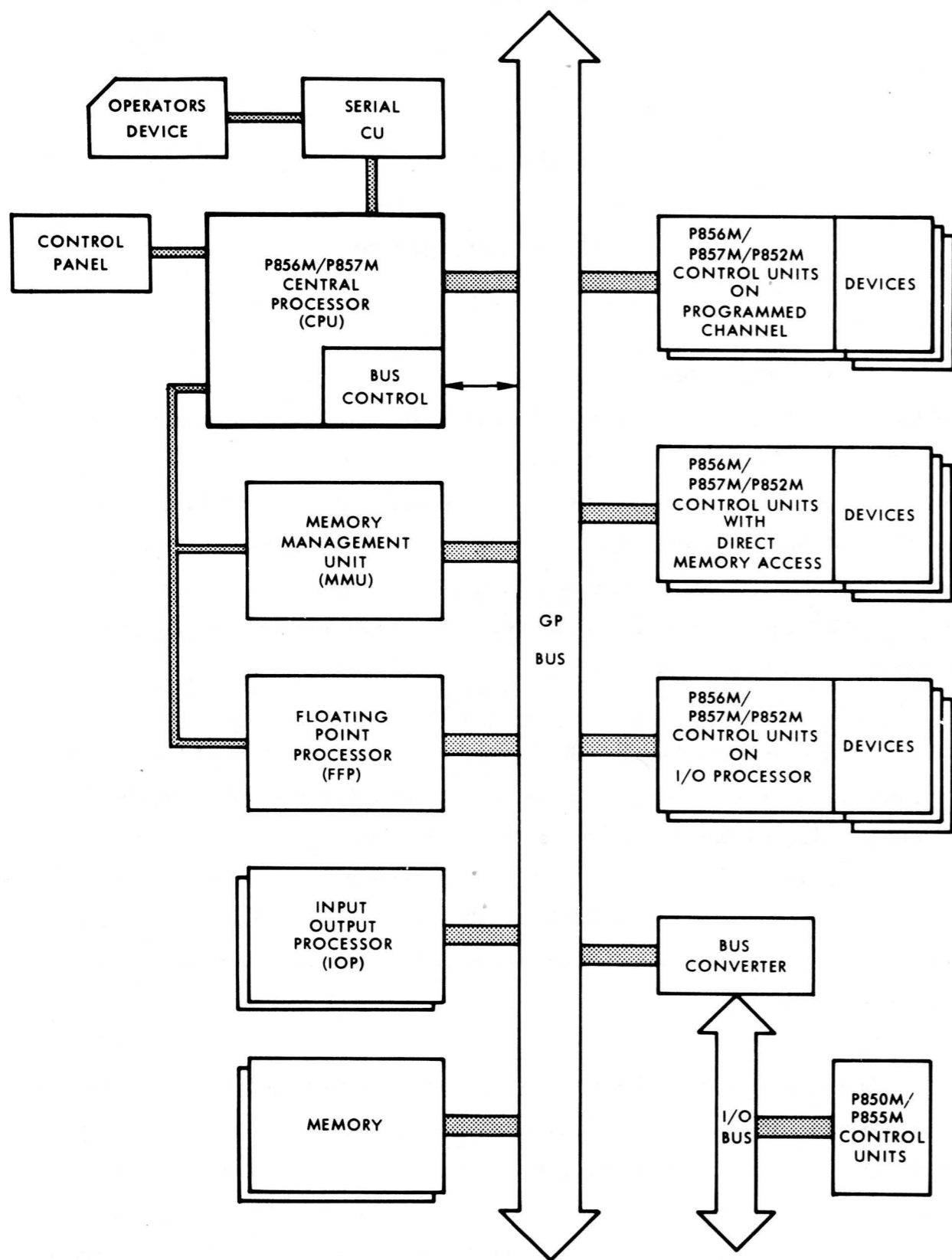


Figure 1-1 P856M/P857M System Block Diagram

1.4 Memory

The P856M maximum memory size is 32k words. The P857M maximum memory size is 32k words with the basic system configuration, 64k words with the MMU option, or 128k words with the MMU and the M5 chassis options. Memories are provided in modular form with up to 16k words per card.

1.5 Memory addressing on the GP Bus lines is by character address, and usually shown in hexadecimal. CPU logic which deals with memory word addresses simply places the word code on the GP Bus lines shifted one bit to the left. This then accesses an even-numbered memory character address. Some of the first (low numbered) memory locations are reserved for hardware-addressed functions, as follows:

Address		Function
Decimal Word	Hexadecimal Character	
0	000	Interrupt list words
62	07C	
63	07E	Trap routine list word
64	080	
127	0FE	
128	100	

↑ OVERFLOW OF STACK

1.6 Memory Management Unit (MMU)

The MMU is a P857 single-card hardware option which uses Virtual Addressing. This option extends the main memory from 32k to 128k words, while still using the 16-bit addressing. The Virtual Addressing system also allows software extension of main store to backing store, via the Direct Memory Access channel. The MMU card uses a dedicated slot adjacent to the CPU card. The MMU is described in Appendix C.

1.7 Floating Point Processor (FPP)

The FPP is a P857 single-card hardware option which performs floating-point arithmetic operations. The FPP uses a dedicated slot (beside the MMU position)

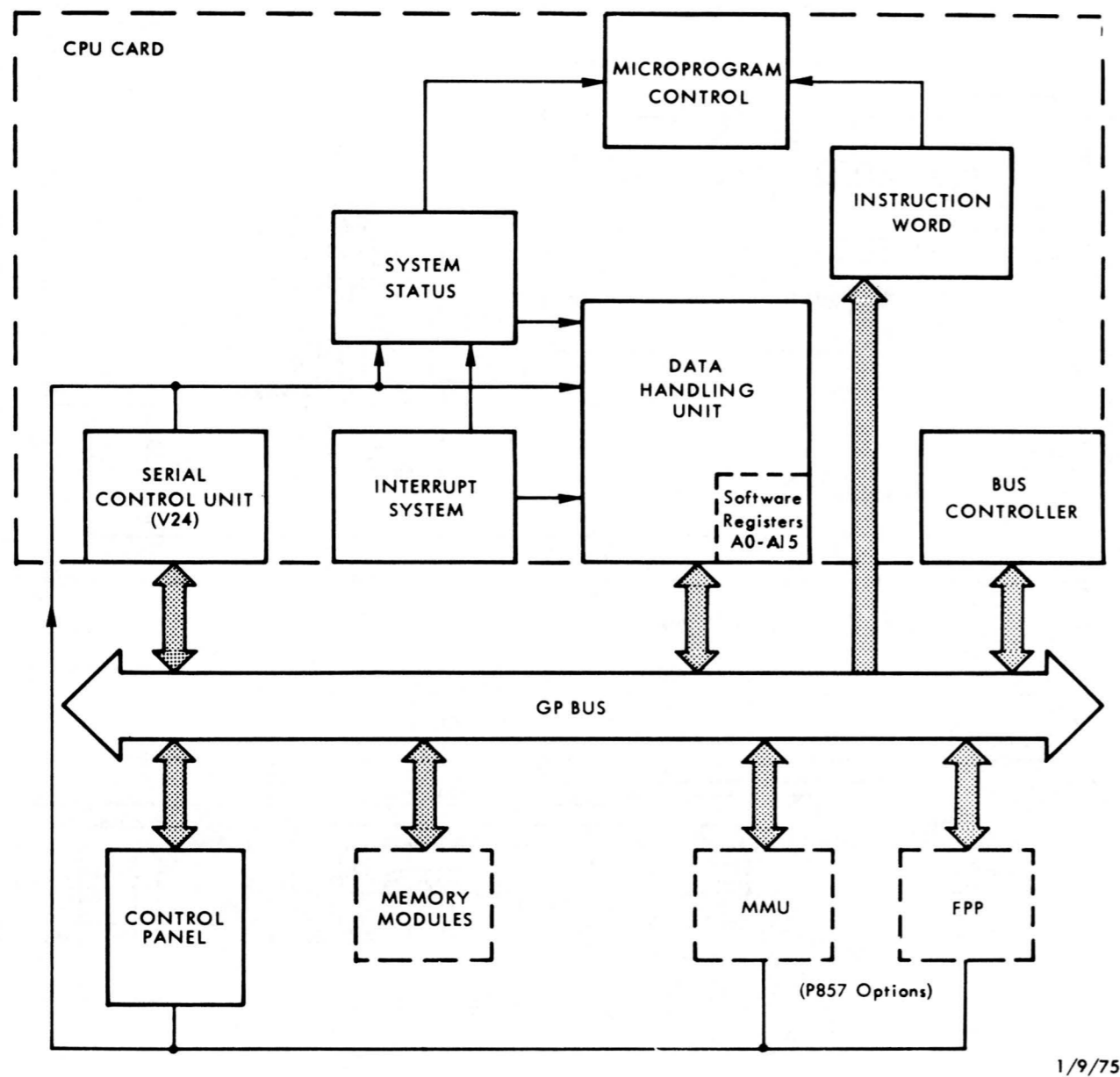


Figure 1-2 P856M/P857M CPU Block Diagram

in the cabinet. Dedicated wiring between the FPP and CPU is used to increase operating speed for the floating point calculations. The FPP is considered an extension of the CPU arithmetic section rather than an independent unit. The FPP is described in Appendix D.

1.8 I/O Processor (IOP)

The IOP is a hardware I/O channel that manages direct data transfers between control units and memory. The IOP multiplexes a number of control units for memory data transfer (eight CUs with the IOP type A). The IOP card can be inserted in any slot of the basic cabinet. The IOP is described in Appendix B. Additional information about I/O channels is given in paragraph 1.23.

1.9 GP BUS

The General Purpose Bus is a 57-line communicating link between all system elements, such as the CPU memory modules, I/O processors, and control units (Figure 1-1). System elements use the GP Bus on a master-slave basis. The CPU operates only as a master; the memory, external registers, and most device control units are slaves; the I/O processor may operate as master or slave (for CU with integrated DMA channel, the DMA can operate as master or slave).

1.10 The Bus Controller logic in the CPU regulates access of masters to the GP Bus. Whenever the Bus is free, the Bus Controller scans the masters in a specific sequence for a Bus-access request. The CPU has direct access to the memory at the completion of each instruction.

1.11 CENTRAL PROCESSOR UNIT

The P856/P857 CPU card contains the complete central processor, the GP-Bus control logic, and a serial control unit. The main CPU logic units and data paths are shown on the block diagram, Figure 1-2. A more detailed block diagram and complete logic diagrams are provided in Section II, CPU Logic description.

1.12 The Data Handling Unit does the processing of all data words accessed by the CPU. This unit also handles the addressing for both data transfers and instruction-word transfers. The Microprogram Control is a read-only memory and associated logic which controls all CPU operations.

1.13 CPU SOFTWARE REGISTERS

A scratchpad comprising sixteen 16-bit registers (A0 to A15) is directly accessible to software. The scratchpad contains 15 working registers (A0 to A14) and a stack pointer (A15). The working registers are used as an operand for some instructions. The scratchpad is located in the Data Handling section of the CPU logic and is connected to the operand-A input of the arithmetic logic unit.

1.14 A 2-bit condition register (CR) is provided for testing operation results. This register is also located in the Data Handling section of the CPU logic.

1.15 BUS CONTROLLER

The Bus Controller scans the GP Bus priority chain for selecting a master, provides control for the memory, and gates input/output data between the GP Bus and the CPU. The Bus Controller is included on the CPU card and is interconnected with the CPU logic.

1.16 INTERRUPT SYSTEM

The Interrupt System is a hardware feature which allows a running program to be interrupted by a higher-priority program. The Interrupt System is used for CPU logic functions and for control-unit I/O channel operations. There are 63 interrupt levels, divided into two groups: internal and external. There is a separate interrupt-request associated with each of the 63 levels.

1.17 Internal Interrupts

The internal interrupts (Figure 1-3A) use the eight highest-priority levels, 0-7. Four of the internal interrupts are pre-wired from CPU logic functions; the other four may be connected to other units in the basic chassis, but are not

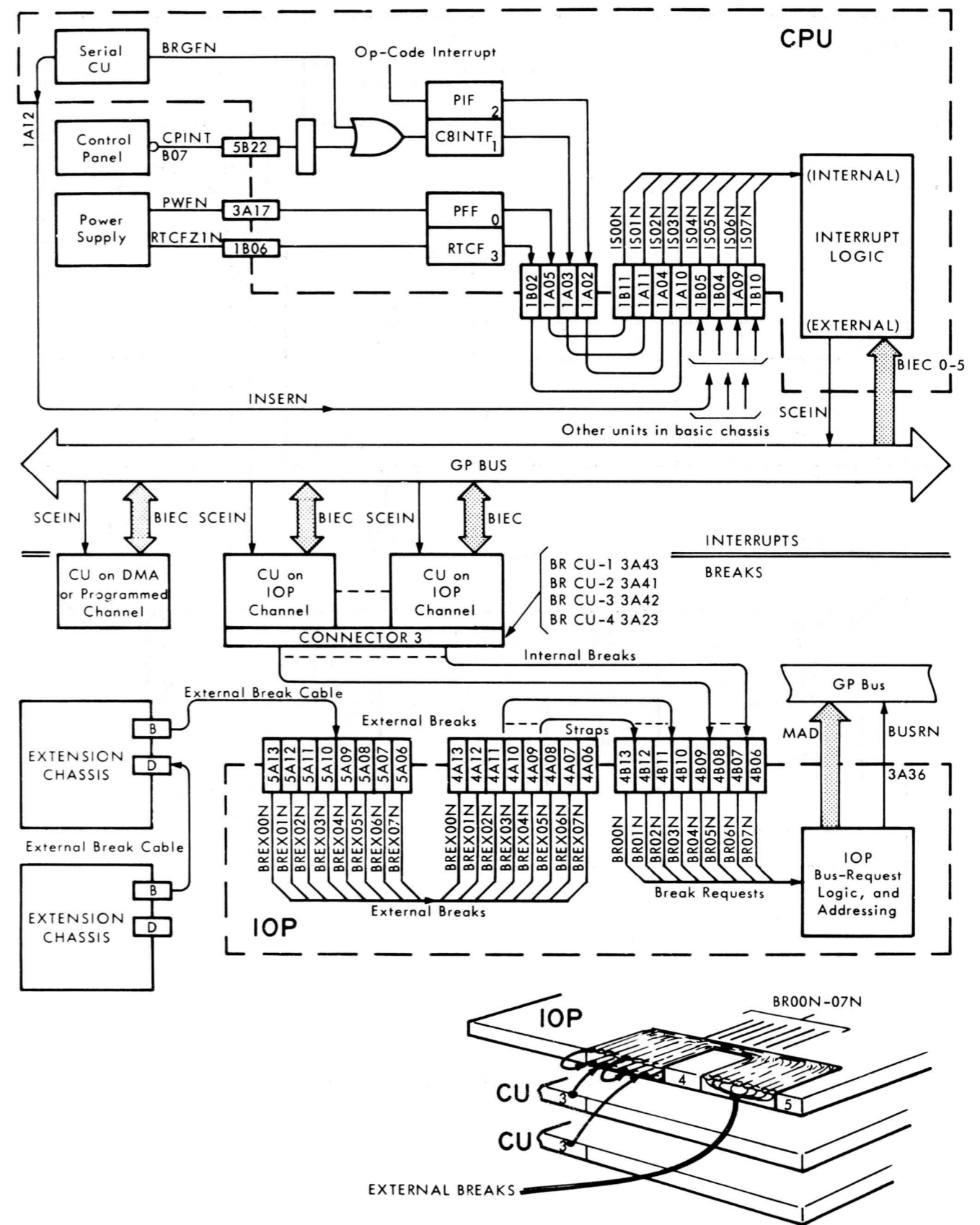


Figure 1-3A Interrupts and Breaks

10/10/75

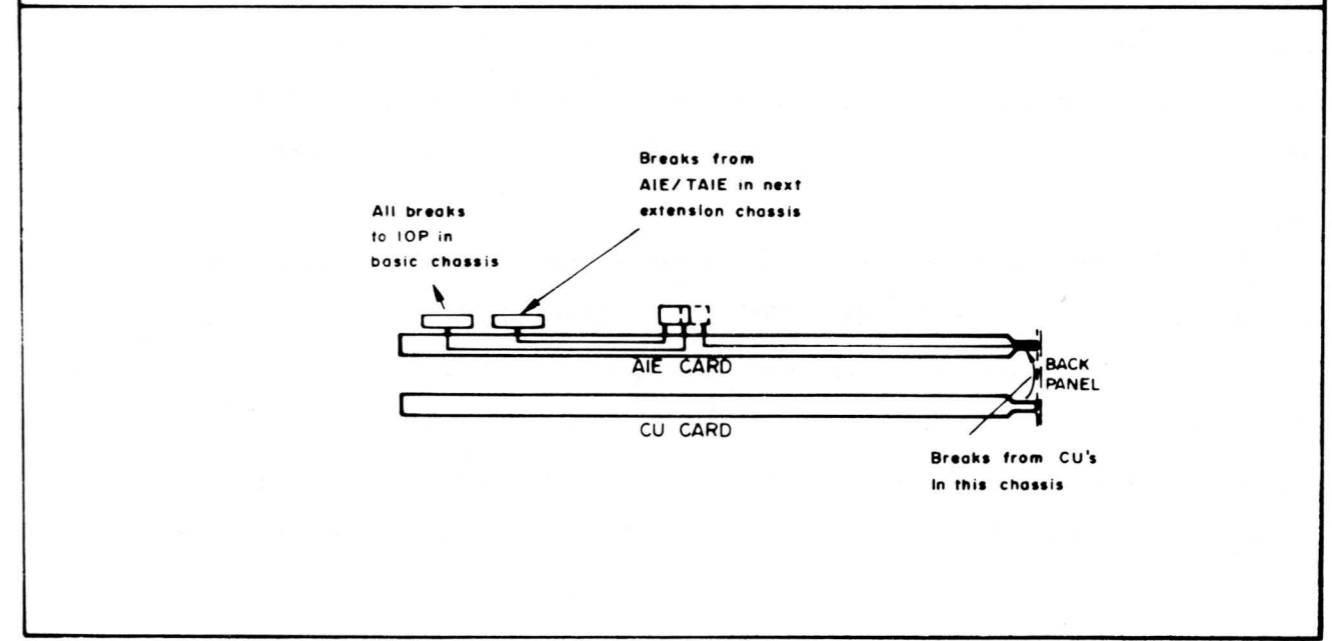
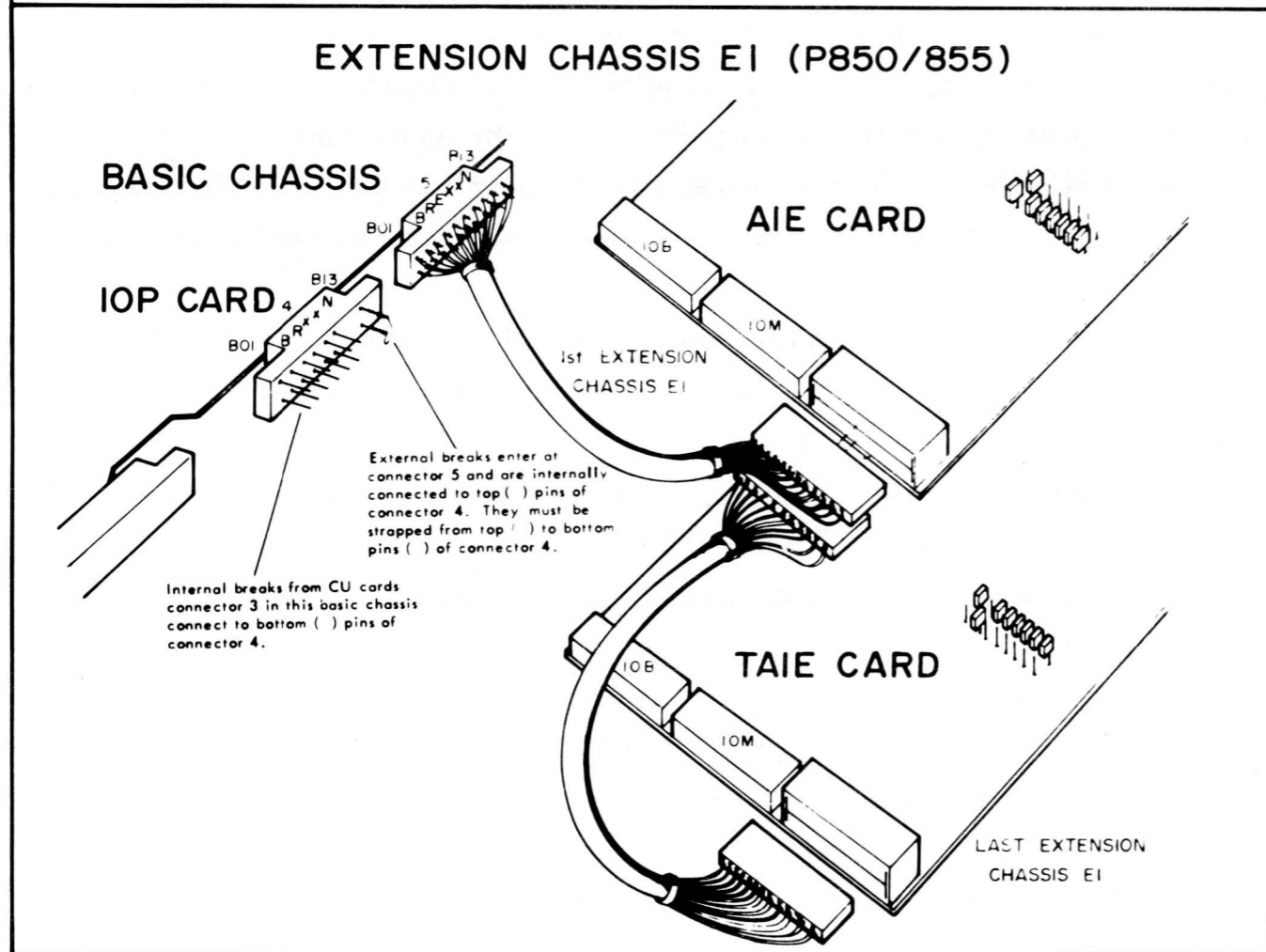
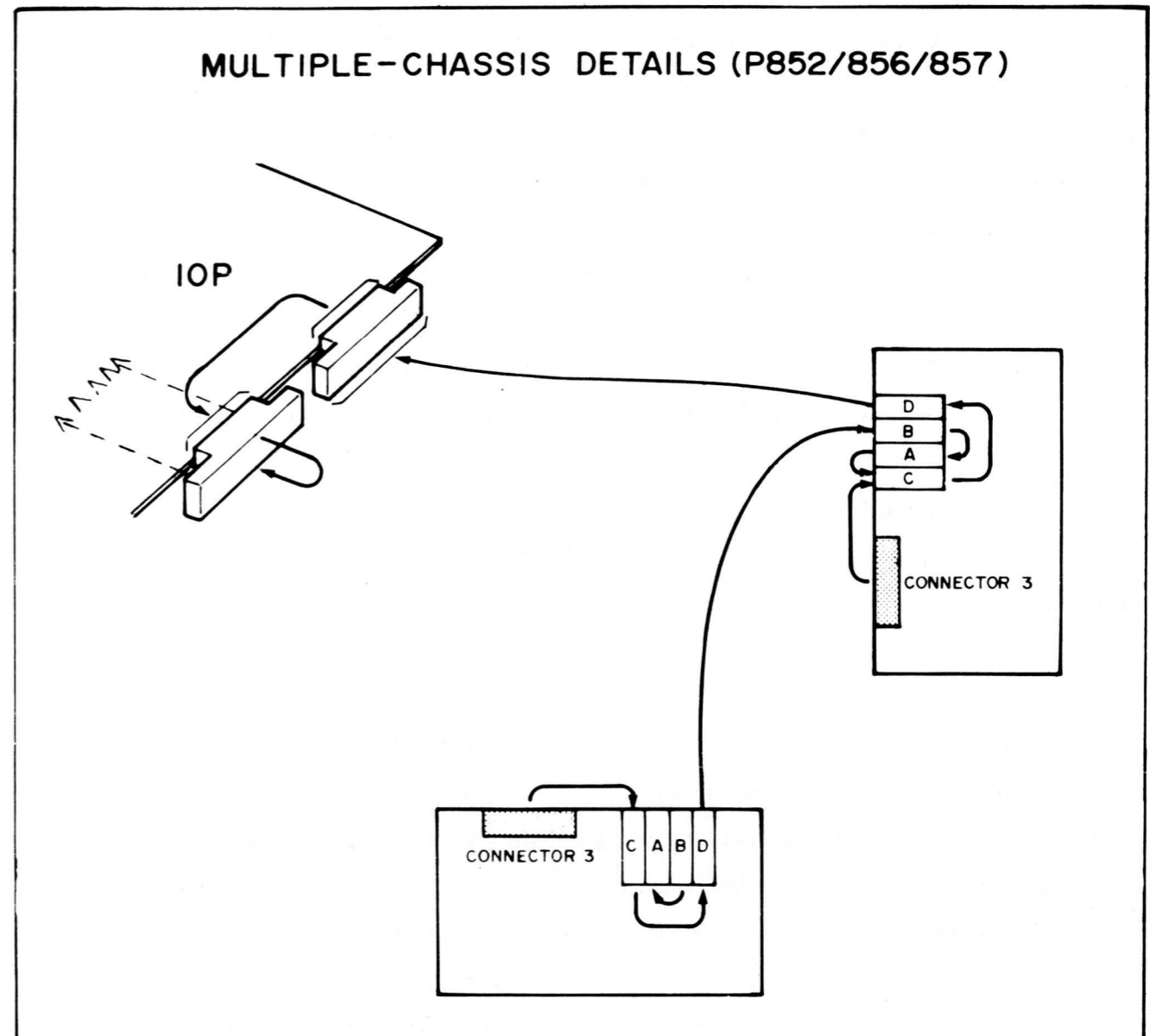
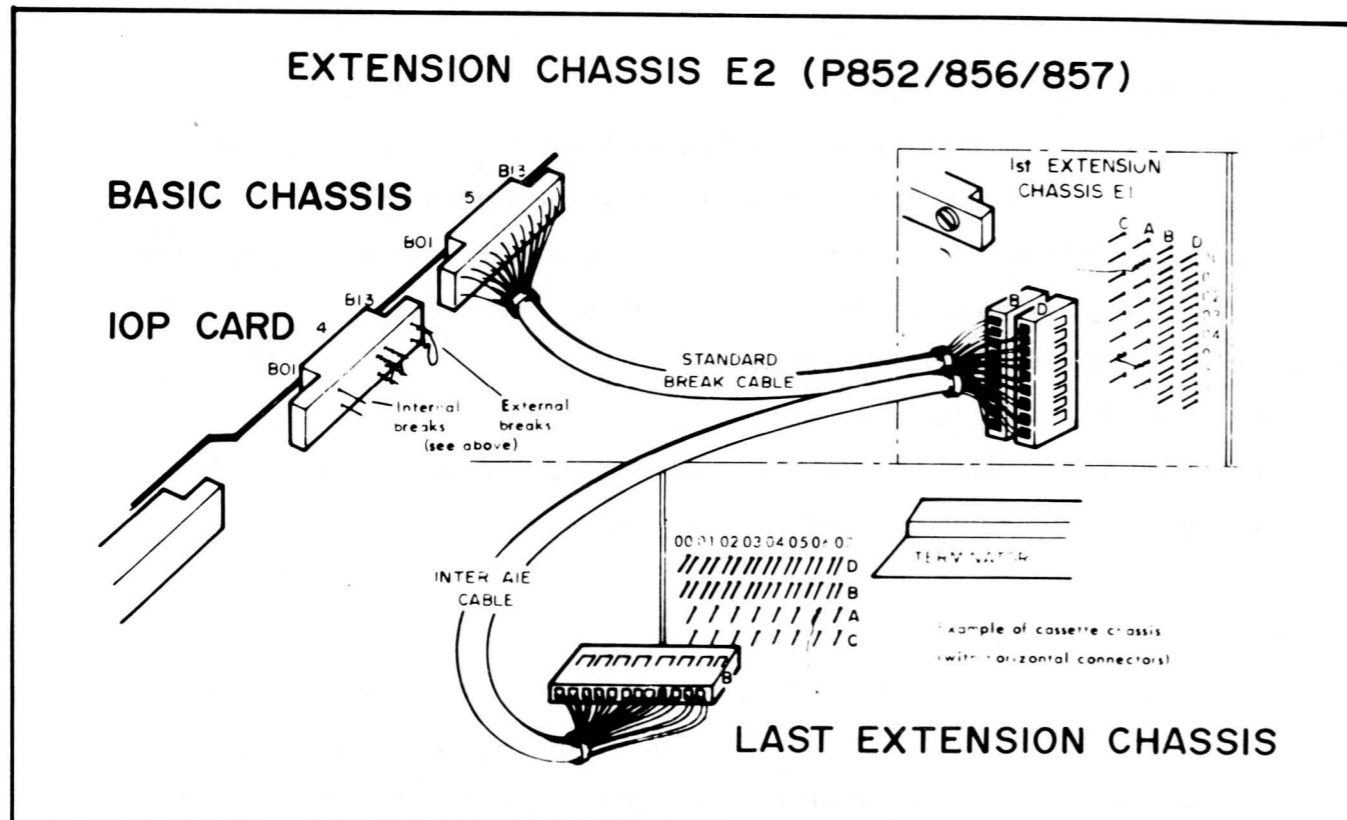


Figure 1-3B Break Cable Details

used for CU I/O channel operations. If any of the levels four through seven are not used for internal interrupts, they may be used by the external interrupts on the BIEC lines. The internal interrupts are assigned as follows:

level 0 - Power failure, Automatic restart.

1 - Operator's interrupt (Control Panel or I/O Console).

2 - Operation Code interrupt (Link to Monitor and Stack Overflow).

3 - Real Time Clock interrupt.

4-7 - Available for other internal interrupts within the basic chassis (serial CU, FPP interrupts, etc.) or external interrupts on the BIEC lines.

The four CPU logic interrupts are set into CPU flip-flops as they occur. They are reset individually by the RIT instruction in the corresponding interrupt program.

1.18 External Interrupts

The external interrupts (Figure 1-3) are assigned priorities 4 to 62, although 4 to 7 may be used by internal interrupts. All I/O control units (except the CPU-integral serial controller) use the external interrupts, including those control units mounted in the basic chassis. Control units on the programmed channel use an external interrupt to transfer each word. Control units on all three channels use the external interrupt to request a status transfer at the end of a data-block transfer. For control units on the IOP channel, word transfers are initiated by break requests (BR) to the IOP; the IOP then makes a Bus Request to obtain control of the GP Bus for the word transfer. A break request is part of the IOP channel (paragraph 1.33) and not part of the interrupt system.

1.19 External interrupt requests are connected to the CPU interrupt logic via a 6-bit code on the GP-Bus BIEC lines. The priority level of a control unit is established by a priority encoder (with a set of jumpers) on the CU itself. Any pending interrupts on the BIEC lines are sampled at the end of the instructions by the scan-interrupt signal SCEIN. The CU priority encoders sample the BIEC lines and only the highest priority external interrupt request is coded onto the lines.

1.20 Interrupt Control

An Enable Interrupt (ENB) instruction is used to enable the CPU interrupt system. The entire interrupt system can be blocked with the Inhibit Interrupt (INH) instruction. The hardware flip-flops generating the internal interrupts are reset individually by the RIT instruction. The hardware that is generating the external interrupts is reset by appropriate CIO instructions. At power-on time, and at every master clear from the control panel:

- the CPU is set to Enable Interrupt mode,
- the current program level is established at 63, and
- all internal and external interrupt requests are reset.

1.21 Interrupt Sequence

The internal/high-priority interrupts (levels 0-7) are sampled at the end of each instruction execution (except Move Table which is sampled early). If there is no internal interrupt, and there has been no external sampling within 2 microseconds, the highest-priority external interrupt (which is coded on the BIEC lines) is sampled. The highest-priority interrupt request is then compared with the priority level of the running program. If the running program is of higher or equal priority to the interrupt request, the program continues. If the interrupt request is of higher priority than the running program, the interrupt sequence is started:

- The current instruction (except Move Table) is completed. For Move Table, registers are updated to allow resuming the instruction at the point it was suspended.
- The program counter (P) is stored in the memory-stack location specified by stack pointer A15. P contains the address of the next instruction (except for Move Table, where P points to the instruction itself). A15 is decremented by 2.
- The program status word is stored in the memory location adjacent to (P), specified by stack pointer A15. A15 is decremented by 2.
- The Inhibit Interrupt state is set.
- The system User Mode flag is reset (unless already reset).
- The priority level register (PLR) is loaded with the new level number.

- An indirect branch is made to the corresponding memory location (paragraph 1.23).
- The interrupt routine is executed.

Note: A Return instruction with a pointer other than A15 can be used independently of the interrupt routine to switch from any program to another under a supervisory program control.

1.22 Interrupt Routine

The following operations must be performed by the interrupt-routine program:

- Some or all of the accumulators (A0 - 15) are saved in the interrupt memory stack.
- The interrupt itself is treated, including a RIT, SST, or other instruction to reset the interrupt signal.
- The accumulators are re-loaded from the stack at the end of the interrupt routine.
- A Return instruction, referring to stack pointer A15, is programmed. During this instruction, the contents of A15 are used (with incrementing) to restore the program counter into P and retrieve the program status word.

1.23 Interrupt Addresses

The first 63 word locations in memory are used for the interrupt-routine list words. The CPU interrupt logic generates a direct six-bit word address for the accepted interrupt. This address code is shifted left one position onto the address lines to produce the memory character address, as follows:

Interrupt Level	Address Code from Interrupt Logic	Bit 0 added, for mem. char. add.	
0	0 0 0 0 0 0	0	0
1	0 0 0 0 0 1	0	2
2	0 0 0 0 1 0	0	4
3	0 0 0 0 1 1	0	6
4	0 0 0 1 0 0	0	8
5	0 0 0 1 0 1	0	A
6	0 0 0 1 1 0	0	C
	etc. to		
62	1 1 1 1 1 0	0	7C

1.24 Memory Stack Operation

The interrupt system utilizes a memory stack with automatic handling. The hardware uses this system stack during the interrupt sequence to save the program status word and the instruction counter of the interrupted program. The software uses the stack: to save and later restore any other parameters of the interrupted program; to link a program to a subroutine; and to return to the main program. The system stack is also used by software for Traps and page faults.

1.25 The stack operates on a last-in first-out basis, controlled by the automatic updating of stack pointer A15. Load, Store, Multiple Load, and Multiple Store can be used as stack-handling instructions when their effective address refers to A15. The Call Function (CF) instruction is a branch with automatic saving of PSW and P into the stack. The Return (RTN) instruction is used at the end of an interrupt routine or a subroutine to restore PSW and P.

1.26 Stack Overflow is signalled by an Operation Code interrupt (level 2, internal interrupt). This signal is generated by hardware when the stack pointer decrements to less than 128_{10} (word address) to indicate that the stack is almost full and to prevent overwriting in the dedicated low address memory locations.

1.27 Additional memory stacks may be used by software, using the scratchpad accumulators A0 to A14 for stack pointers. These software stacks will not have the automatic handling and updating like the system stack which uses A15 as the pointer. All references here to the stack pertain to the automatic-handling stack. All memory stacks may have software limits to stack size established at system generation time.

1.28 INPUT/OUTPUT CHANNELS

The P852M/P856M/P857M systems have three different input/output channels:

- programmed channel,
- multiplex-type I/O-processor channel, and
- direct memory access channel.

All I/O data transfers are via the GP Bus and are timed by the Bus Controller

logic on the CPU card. For all three types of I/O transfers, the program initiates control-unit operation with a CIO Start command. The CU must reply with an interrupt request or a break request (depending on the type of I/O channel) when it is ready for the first word (or character) transfer. The data are then transferred according to the channel type. When a data-block transfer is ended (either complete or early), the CU signals to the CPU with an interrupt request. The program must then issue a Send Status (SST) command to obtain the status word from the CU.

1.29 For transferring a block of data via any I/O channel, the program provides the memory address of the first word and the length of the block to be transferred. During the transfer, these two control words must be updated: the memory address is incremented to select sequential locations in the data block; the block length is decremented to determine when the complete block has been transferred (length = zero). The method of handling this pair of control words depends on the type of I/O channel.

1.30 Programmed Channel

This is an input/output exchange between a CU and memory via the CPU, under complete program control (Figure 1-4). The exchange is word-by-word or character-by-character at up to 40,000 characters per second. On the programmed channel, the address/length control-word set is located in program registers. For each data word or character transferred, the program must access both of these registers to up-date the control words.

1.31 For an output transfer, a Load Register instruction loads the first word from memory into the CPU scratchpad register. The CU signals that it is ready with an Interrupt Request to the CPU. An OTR instruction then transfers the word from the CPU to the CU, while another Load instruction obtains the next word from memory. This procedure continues until the last word of the block is transferred. When the program loads the last data word, its block length is counted to zero. The CPU then sends CIO Halt to the CU along with the last OTR data transfer, and the transfer is ended with the status transfer.

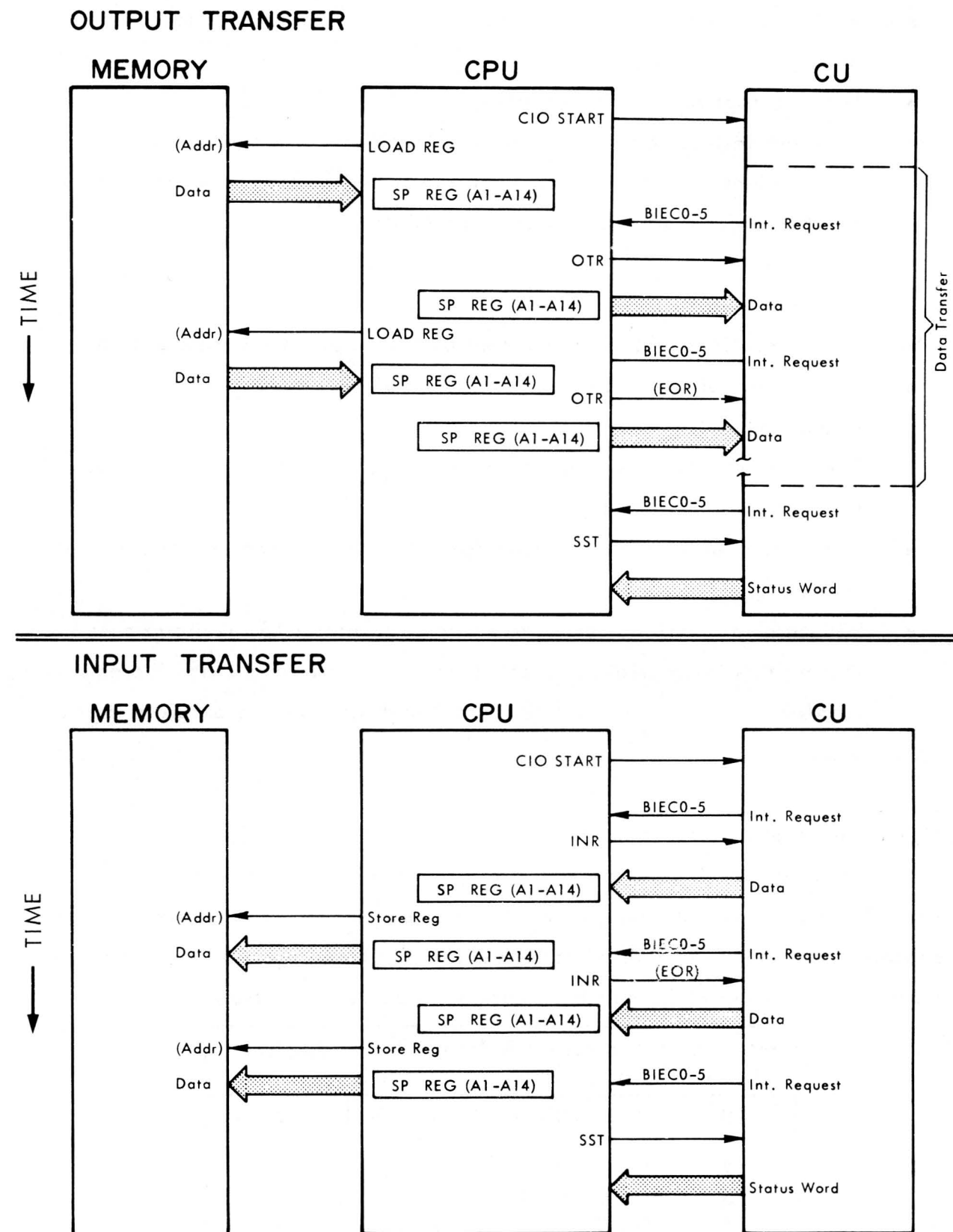


Figure 1-4 Programmed Channel Transfers

1.32 The input transfer sequence is essentially the same as the output transfer. The input sequence, however, begins with an INR command which transfers data from the CU into the CPU scratchpad register. A Store Register instruction then transfers the word from the scratchpad register into the memory.

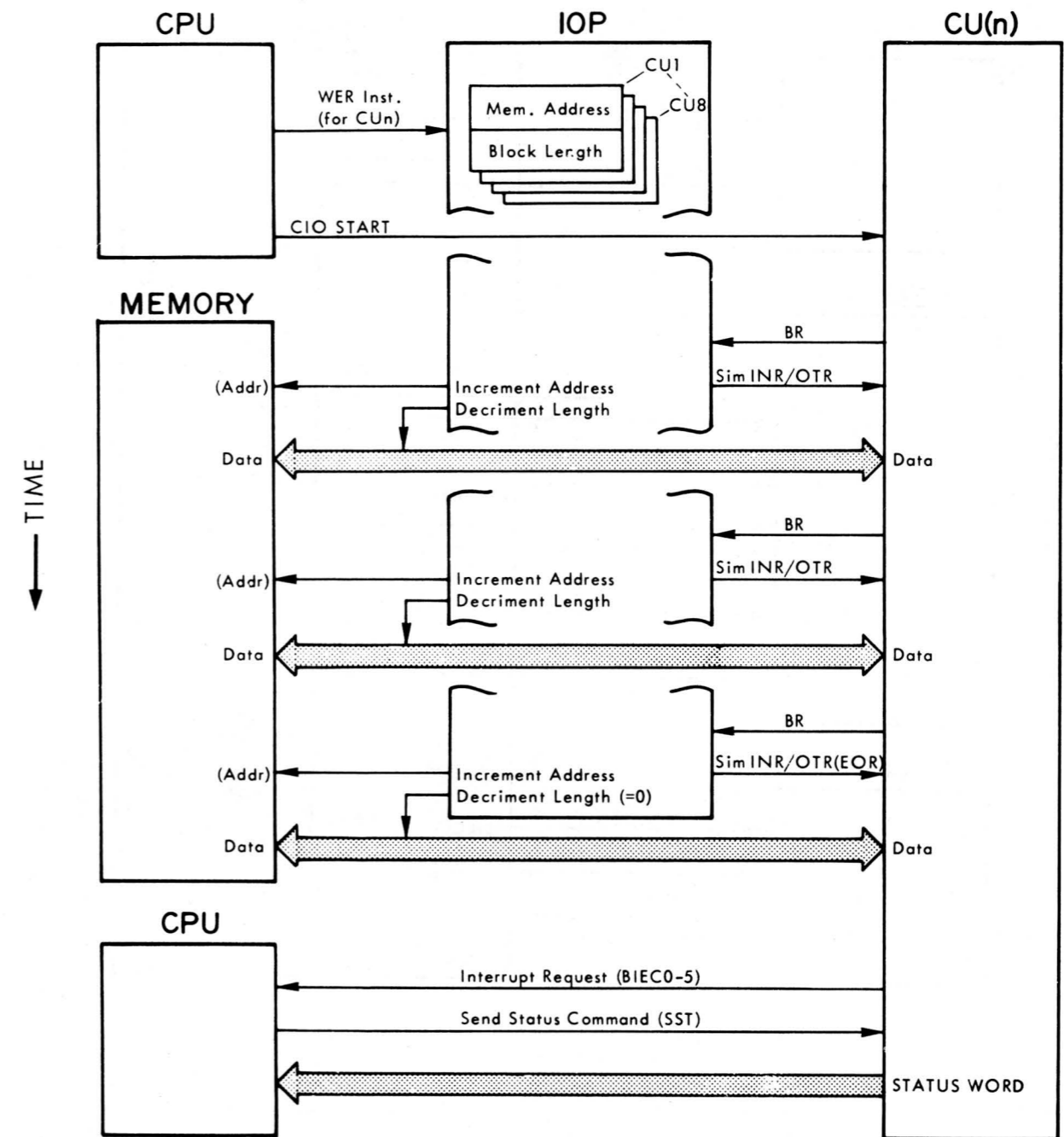
1.33 I/O Processor Channel

The Input/Output Processor (IOP) channel manages data transfers directly between memory and a number of multiplexed control units (Figure 1-5). CPU registers are not used, and there is no need for program control except for starting the exchange and testing status at the completion. The exchange is in blocks at up to one million words per second.

1.34 The IOP is a hardware option that contains a pair of address/length control-word registers for each of its CU channels. At the beginning of a transfer to one CU, the program uses two WER instructions to load this register pair with the starting address and the block length. The IOP logic then provides all GP Bus timing signals to control the data transfers directly between the memory and the CU.

1.35 For input or output transfer, the CU signals that it is ready with a Break Request (BR) to the IOP. The IOP makes a Bus Request to obtain control of the GP Bus (Figure 1-3). The IOP then sends a simulated INR or OTR command to the CU to initiate one word transfer. The INR/OTR command is simulated in that it is generated by the IOP and is not a programmed instruction. The IOP logic updates its control-word registers for each data word. When the block length is counted to zero, the IOP sends End Of Record (EOR) to the CU along with the last simulated INR/OTR data transfer. The data block transfer is ended with an SST command and status transfer between the CU and CPU.

1.36 The IOP can be loaded with the data-exchange control information for a number of control units (up to eight for the type-A IOP). The IOP then multiplexes the exchanges between the CUs and the memory.



Note -- This diagram shows the sequence for transferring a block of data between memory and a single CU, (n). The IOP can multiplex up to eight CUs; the operation shown is thus duplicated for each CU.

Figure 1-5 Multiplexed I/O Processor Channel Transfers

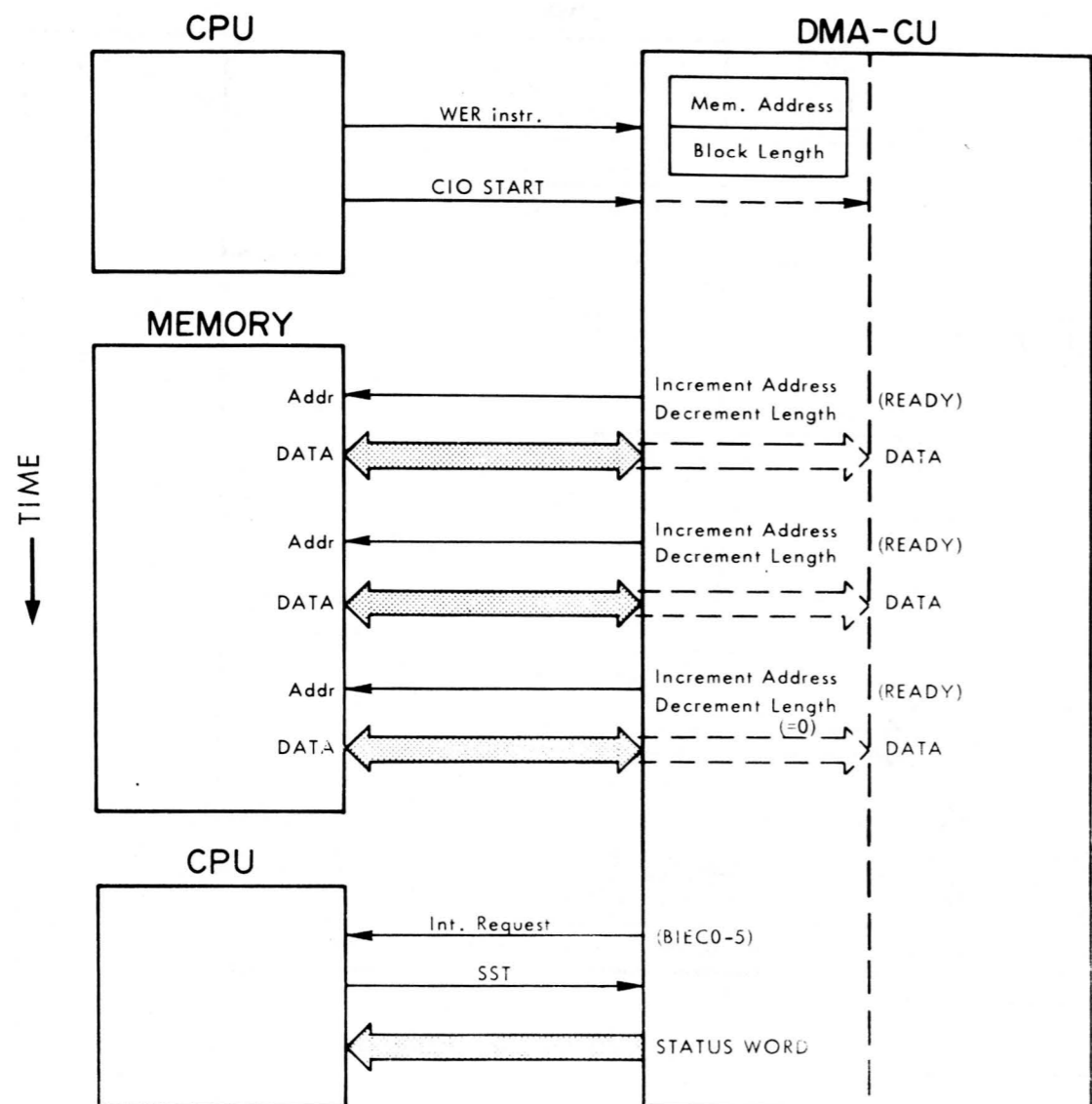
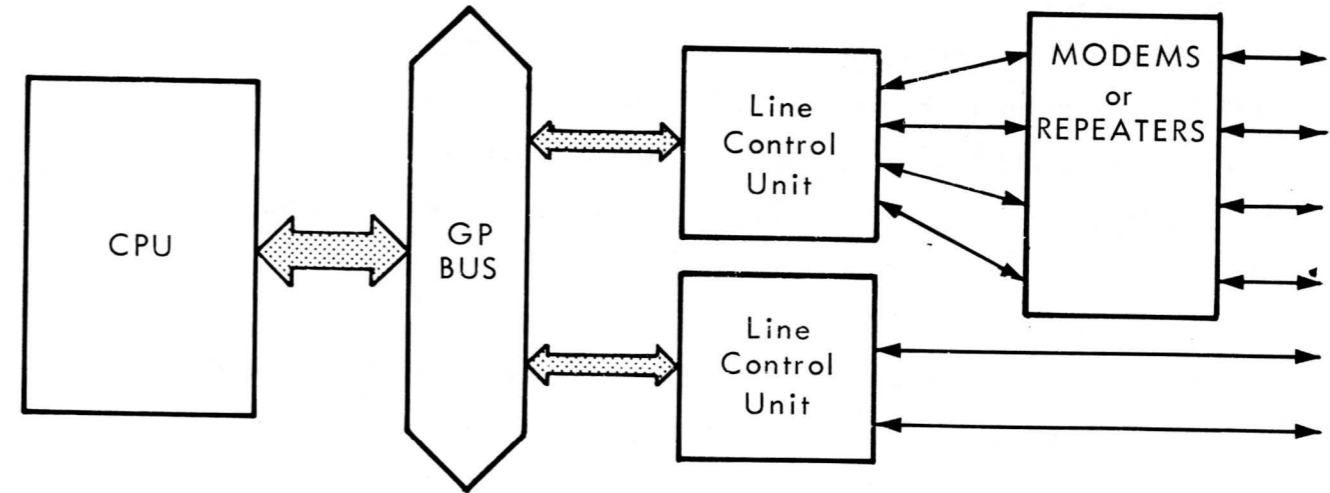


Figure 1-6 Direct Memory Access Channel Transfer

1.37 Direct Memory Access Channel
 The DMA channel manages data transfers directly between memory and a single high-speed control unit (Figure 1-6). CPU registers are not used, and there is no need for program control except for starting the exchange and testing status at the completion. The DMA channel is a hardware channel included as a part of the high-speed control unit (DMA-CU card).

1.38 At the beginning of a transfer, the program uses two WER instructions to load the starting address and block length into the DMA control-word register. The DMA logic then provides all GP Bus timing signals to control the data transfers directly between the memory and the DMA CU. The DMA logic also updates the control-word register for each data word and detects when the complete block has been transferred. The data block transfer is ended with an SST command and status transfer between the CU and CPU.

1.39 Data Communications Channels
 Data communication with the P856M/P857M system is accomplished with various Line Control Units operating via the programmed channel or the multiplexed IOP channel. The Line Control Units connect to the GP Bus in the same manner as any other control unit cards. The Line Control units are connected to the communications lines, either directly or via modem interface or repeater units (following diagram).



Some Line Control Units used with the system are:

- SLCU Synchronous line control units
- ALCU Asynchronous line control units
- ALM Asynchronous low speed multiplexers

1.40 CLOCKS

1.41 CPU Timing Clock

The CPU logic timing is controlled by the Sequensor which provides the clock signals AP, BP, and T1 through T10. The Sequensor is described in Section II.

1.42 Real Time Clock

A Real Time Clock produces an interrupt signal every 20ms. The real-time clock pulse is produced directly from the power supply and is described in Section V.

1.43 CONTROL PANEL

1.44 The standard control panel for the P856M system is the same as for the P852M. This panel is directly interchangeable between the two systems. The Extended control panel used by the P857M is available as an option for the P856M system. Description and operation of the control panel is provided in Section III.

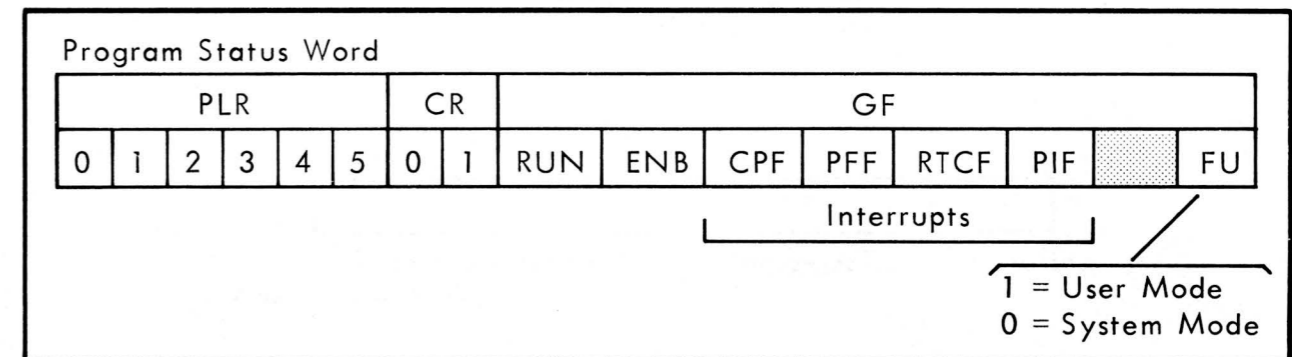
1.45 The standard control panel for the P857M system is the Extended Control Panel. This panel has, in addition to the standard Data and Control facilities, complete Addressing facilities, including the ability to stop on preset addresses. The Data/Control half of the panel is the same as the P852M/P856M control panel, but with the addition of a TEST position on the key switch for performing automatic microdiagnostic tests. Description and operation is provided in Section III.

1.46 TESTING

The P857M contains an automatic testing feature in the form of a microprogrammed diagnostic built in to the CPU logic. This automatic microdiagnostic operates through the control panel to test approximately 70% of the CPU logic. Successful running of the microdiagnostic indicates that sufficient parts of the CPU function for loading of test programs. The automatic microdiagnostic tests and the test programs are described in Section III.

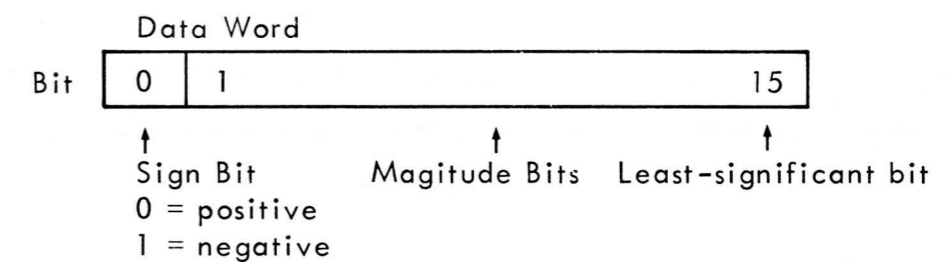
1.47 STATUS

System status is contained in a sixteen-bit Program Status Word (PSW). The use and operation of the status word is described in Section II (Paragraph 2.84). The status word contents are shown in the following diagram.



1.48 DATA FORMAT

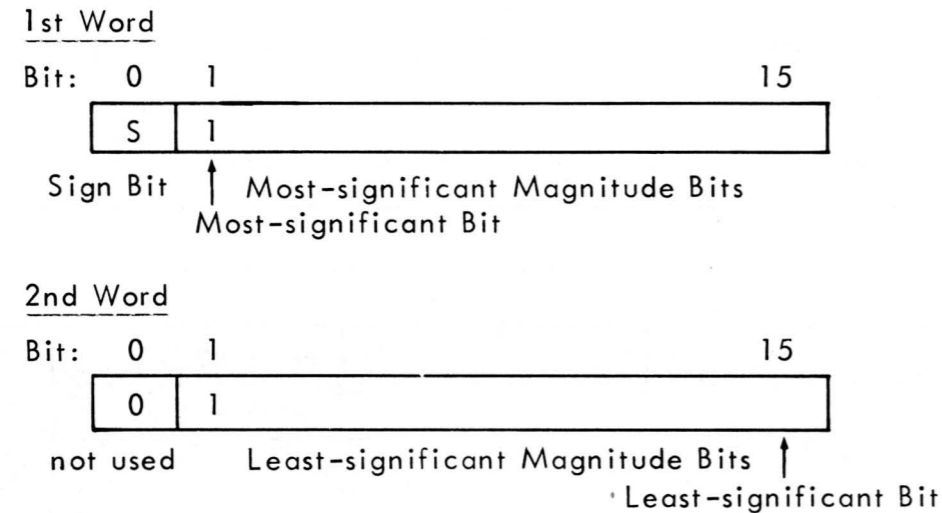
The data are handled in 16-bit words with 15 magnitude bits and bit 0 as a sign bit. Bit 1 is the most-significant data bit. The data word is handled in the CPU and on the GP Bus with negative logic: a logic 1 is a low level (0V) and a logic 0 is a high level (+5V).



1.49 Double Precision

Double precision is obtained by utilising two successive words to obtain 30 magnitude bits. The sign bit and the 15 most-significant bits are in the first word; the 15 least-significant bits are in the second word. Bit 0 of the second word is not used, and is always zero.

1.50 Double precision is used for the product of the multiplication of two single-precision words, and for some other operations.

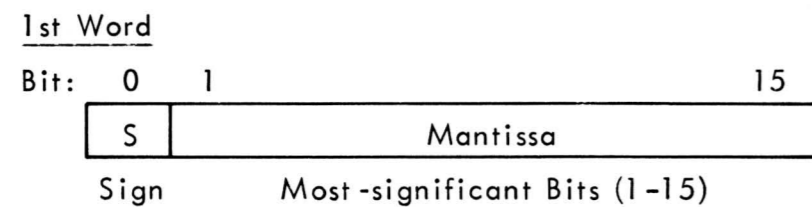


1.51 Logical Data

Logical data, such as the condition of sixteen binary indicators, can be stored in a single data word. This type of data is generally not treated arithmetically by the program but logically by means of boolean operations such as AND, OR, and Exclusive-OR. In this case, bit 0 of a word is not used as a sign bit.

1.52 Floating-Point Data

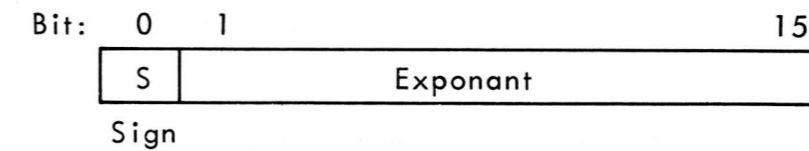
Real, floating-point numbers are contained in three successive words. The mantissa is stored in the first two words as a double-precision integer. The third word contains the exponent, represented as a single-precision integer.



2nd Word



3rd Word



1.53 Character Handling

Character handling is performed by some instructions. The right character of a word is the least significant (bits 8-15) and the left character is the most significant (bits 0-7).

1.54 OPERATING MODES

The CPU can operate in two basic modes: System Mode or User Mode. The mode is specified by bit 15 of the program status word (0 = system; 1 = user).

1.55 System Mode

The System Mode is reserved for the monitor program and system programs. In the System Mode, execution of the complete instruction set is allowed. This mode assures the system resource allocation, protected by the following privileged instructions:

- Control Instructions which modify the CPU state:
HLT -- Halt
INH -- Inhibit Interrupt
RIT -- Reset Internal Interrupt
- All I/O Instructions:
CIO, OTR, INR, SST, TST
- External Register Instructions:
WER, RER
- (P857M) MMU-related instructions, Extended Load or Store, and Segment Table Load or Store:

ELR, EL, ESR, ES,
TLR, TL, TSR, TS

- Some other instructions are reserved for System Mode when they modify the contents of the stack pointer (A15). Refer to Table 1-1.

1.56 User Mode

This mode is reserved for the user program execution. If a program in User Mode attempts to execute a privileged instruction, a Trap routine is performed: the program parameters (P and PSW) are saved in the stack, and the program branches to the address contained in memory location 7E.

1.57 When a program running in User Mode needs system allocation, a Link to Monitor (LKM) instruction executes a call to the monitor which sets the CPU to System Mode.

1.58 When a program running in System Mode is complete, and ready to allow user programs to run, either a Set Mode (SMD) instruction or a Return (RTN) instruction with R2=15 can be used. The User-Mode indicator in the PSW is then set to 1.

1.59 INSTRUCTIONS

The P856/857 instruction set is divided into ten groups:

- load and store
- arithmetic
- logical
- character handling
- branch
- shift
- table handling
- control
- input/output
- external transfers

Table 1-1 lists the instructions and indicates the operating flow diagram for

each instruction. The instruction formats are described in paragraph 1.62. Instructions are specified by addressing mode as well as the op-code. The addressing mode is described in paragraph 1.65 and listed in Table 1-2.

1.60 Invalid Instructions

An invalid instruction code initiates a Trap microprogram (paragraph 1-55, Figure 1-12). The program must determine what action (such as interrupt) to take following a Trap.

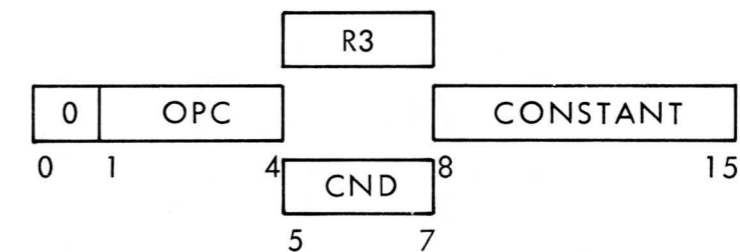
1.61 Some P857 instructions require the Floating Point Processor (FPP) or the Memory Management Unit (MMU). Both of these system extensions are optional. If the FPP is not present, the corresponding instructions initiate the Trap. If the MMU is not present, the corresponding instructions give no significant result when they are executed.

1.62 Instruction Format

There are two instruction formats, indicated by bit 0 of the instruction word:

- Format 0 Constant handling instructions.
- Format 1 Memory reference or register-to-register operations.

1.63 Format 0 Instructions (Type T8)



OPC : Operation code

R3 : Register (scratchpad A0-A7) on which the operation is performed.

CND : Condition for relative branching (when specified by OPC).

Table 1-1 P856/857 Instruction List

Flow Diagram 1:	Mnemonic	OPC	L/S (Bit 15)	Address Type	Instruction Name
Load and Store Instructions					
15	LDK	0	0	T8,2	Load constant
	LDR	0	0	T1,3	Load reg/reg; update stack pointer
	LD	0	0	T4-7	Load register
	STR	0	1	T3	Store reg/reg; update stack pointer
	ST	0	1	T4-7	Store register
23	MLK	7	0	T2	Multiple load constant
	MLR	7	0	T3	Multiple load register
	ML	7	0	T4-7	Multiple load
	MSR	7	1	T3	Multiple store register
	MS	7	1	T4-7	Multiple store
29	ELR	10	0	T3	Extended load register (MMU)
	EL	10	0	T4-7	Extended load (MMU)
	ESR	10	1	T3	Extended store register (MMU)
	ES	10	1	T4-7	Extended store (MMU)
Arithmetic Instructions					
17	ADK	2	0	T8,2	Add constant
	ADR	2	0/1	T1,3	Add reg/reg
	AD	2	0/1	T4-7	Add
	IMR	2	1	T3	Increment memory/reg
	IM	2	1	T4-7	Increment memory
18	SUK	3	0	T8,2	Subtract constant
	SUR	3	0/1	T1,3	Subtract reg/reg
	SU	3	0/1	T4-7	Subtract
	NGR	3	1	T1	Negate register
	C2R	3	1	T3	Two's complement/reg
	C2	3	1	T4-7	Two's complement
25	MUK	8	0	T2	Multiply with constant
	MUR	8	0	T1,3	Multiply reg/reg
	MU	8	0	T4-7	Multiply
26	DVK	9	0	T2	Divide with constant
	DVR	9	0	T1,3	Divide reg/reg
	DV	9	0	T4-7	Divide
29	DAK	10	0	T2	Double add with constant
	DAR	10	0	T1,T3	Double add reg/reg
	DA	10	0	T4-7	Double add

Flow Diagram 1:	Mnemonic	OPC	L/S (Bit 15)	Address Type	Instruction Name
Arithmetic Instructions					
30	DSK	11	0	T2	Double subtract with constant
	DSR	11	0	T1,3	Double subtract reg/reg
	DS	11	0	T4-7	Double subtract
28	FFL	9	0	T1	Integer to floating point (FPP)
	FFX	9	1	T1	Floating point to integer (FPP)
27	FADR	9	0/1	T3	Floating-point addition/reg (FPP)
	FAD	9	0/1	T4-7	Floating-point addition (FPP)
	FSUR	9	0/1	T3	Floating-point subtract/reg (FPP)
	FSU	9	0/1	T4-7	Floating-point subtract (FPP)
	FMUR	9	0/1	T3	Floating-point multiply/reg (FPP)
	FMU	9	0/1	T4-7	Floating-point multiply (FPP)
	FDVR	9	0/1	T3	Floating-point divide/reg (FPP)
	FDV	9	0/1	T4-7	Floating-point divide (FPP)
Logical Instructions					
19	ANK	4	0	T8,2	Logical AND with constant (R3≠0, R1≠0)
	ANR	4	0/1	T1,3	Logical AND reg/reg (R1≠0)
	AN	4	0/1	T4-7	Logical AND (R1≠0)
20	ORK	5	0	T8,2	Logical OR with constant (R3≠0, R1≠0)
	ORR	5	0/1	T1,3	Logical OR reg/reg (R1≠0)
	OR	5	0/1	T4-7	Logical OR (R1≠0)
	XRK	6	0	T8,2	EXclusive OR with constant (R1≠0)
	XRR	6	0/1	T1,3	EXclusive OR reg/reg (R1≠0)
	XR	6	0/1	T4-7	EXclusive OR (R1≠0)
32	CWC	13	X	T8	Compare word to short constant
	CWK	13	0	T2	Compare word with constant
	CWR	13	0/1	T1,3	Compare word reg/reg
	CW	13	0	T4-7	Compare word
37	C1R	15	0/1	T1,3	One's complement reg/reg
	C1	15	0/1	T4-7	One's complement
19	TM	4	1	T1	Test mask
20	TNM	6	1	T1	Test not mask
19	CMR	4	1	T3	Clear memory/reg
	CM	4	1	T4-7	Clear memory

Flow Diagram 1:	Mnemonic	OPC	L/S (Bit 15)	Address Type	Instruction Name	Bits			
Character Handling Instructions									
31	ECR	12	0	T1	Exchange character reg/reg (R1≠0)				
	LCK	12	0	T2	Load character with constant (R1≠0)				
	LCR	12	0	T3	Load character/reg (R1≠0)				
	LC	12	0	T4-7	Load character				
	SCR	12	1	T3	Store character/reg (R1≠0)				
	SC	12	1	T4-7	Store character				
32	CCK	13	1	T2	Compare character/constant (R1≠0)				
	CCR	13	1	T3	Compare character/reg (R1≠0)				
	CC	13	1	T4-7	Compare character (R1≠0)				
Branch Instructions									
16	AB	1	0	T8,2	Absolute conditional branch				
	ABR	1	0/1	T1,3	Absolute conditional branch to reg				
	ABI	1	0	T4-7	Absolute branch				
29	RF	10	0	T8	Relative forward conditional branch				
30	RB	11	0	T8	Relative backward conditional branch				
35	CF	14	1	T2	Call function (direct) (R1≠0)				
	CFR	14	1	T1,3	Call function/reg (R1≠0)				
	CFI	14	1	T4-7	Call function (via memory) (R1≠0)				
33	RTN	14	0	T3	Return (R1=0)				
34	EXK	14	1	T2	Execute constant (R1=0)				
	EXR	14	1	T1,3	Execute/register (R1=0)				
	EX	14	1	T4-7	Execute (R1=0)				
Shift Instructions									
21	SLA	7	X	T8	Single left arithmetic (R3≠0)	0	0	0	
	SRA	7	X	T8	Single right arithmetic (R3≠0)	0	0	1	
	SLL	7	X	T8	Single left logical (R3≠0)	0	1	0	
	SRL	7	X	T8	Single right logical (R3≠0)	0	1	1	
	SLC	7	X	T8	Single left circular (R3≠0)	1	1	0	
	SRC	7	X	T8	Single right circular (R3≠0)	1	1	1	
	SLN	7	0	T8	Single left, normalize (R3≠0)	1	0	0	
	SRN	7	0	T8	Single right, normalize (R3≠0)	1	0	1	
	22	DLA	7	X	T8	Double left arithmetic (R3=0)	0	0	0
		DRA	7	X	T8	Double right arithmetic (R3=0)	0	0	1
DLL		7	X	T8	Double left logical (R3=0)	0	1	0	
DRL		7	X	T8	Double right logical (R3=0)	0	1	1	
DLC		7	X	T8	Double left circular (R3=0)	1	1	0	
DRC		7	X	T8	Double right circular (R3=0)	1	1	1	

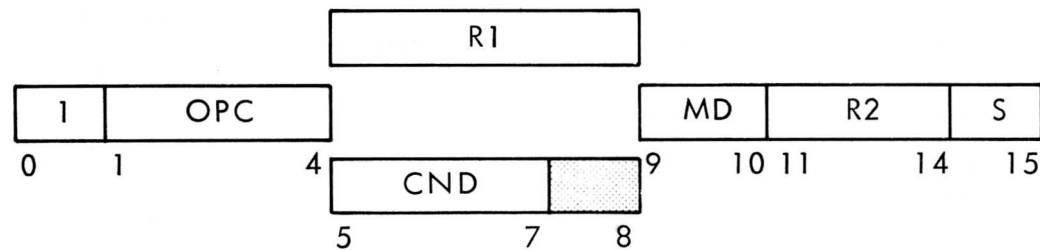
Flow Diagram 1:	Mnemonic	OPC	L/S (Bit 15)	Address Type	Instruction Name	Bits		
Shift Instructions								
22	DLN	7	0	T8	Double left, normalize	1	0	0
	DRN	7	0	T8	Double right, normalize	1	0	1
Table Handling Instructions								
36	MVF	14	0	T8	Move table forward	0	0	0
37	MVB	15	0	T8	Move table backward	0	0	0
36	MVSU	14	0	T8	Move table, system to user (MMU)	1	0	0
37	MVUS	15	0	T8	Move table, user to system (MMU)	1	0	0
Control Instructions								
19	HLT	4		T8	Halt	0	1	1
	INH	4		T8	Inhibit interrupt	1	0	1
	RIT	4	1	T8	Reset internal interrupt	1	1	1
20	ENB	5		T8	Enable interrupt	0	1	0
	LKM	5		T8	Link to monitor	0	0	0
	SMD	5		T8	Set mode	0	0	0
Input/Output Instructions (R3≠0)								
24	CIO	8	X	T8	Control I/O (DA≠0)	1	n	
	OTR	8	X	T8	Output from register (DA≠0)	0	n	
	INR	9	X	T8	Input to register	0	n	
	SST	9	X	T8	Sense status	1	1	
	TST	9	X	T8	Test status	1	0	
External Transfer Instructions								
33	WER	14	X	T8	Write external register (R3≠0)			
37	RER	15	X	T8	Read external register (R3≠0)			
27	FLDR	8	0	T3	Floating-point load/reg (R1=2) (FPP)			
	FLD	8	0	T4-7	Floating-point load (R1=2) (FPP)			
	FSTR	8	1	T3	Floating-point store/reg (R1=2) (FPP)			
	FST	8	1	T4-7	Floating-point store (R1=2) (FPP)			
23	TLR	7	0	T3	Segment table load/reg (R1=0) (MMU)			
	TL	7	0	T4-7	Segment table load (R1=0) (MMU)			
	TSR	7	1	T3	Segment table store/reg (R1=0) (MMU)			
	TS	7	1	T4-7	Segment table store (R1=0) (MMU)			

= P857 only

CND			Branch
5	6	7	
0	n	n	if CR = bits 6,7
1	n	n	if CR ≠ bits 6,7
1	1	1	unconditional

CONSTANT: An 8-bit, positive constant, branch displacement, or OPC extension.

1.64 Format 1 Instructions (Types T1-T7)



OPC : Operation Code

R1 : Register (scratchpad A0-15) on which the operation is performed:

bit 8 = 0 : A0-7

bit 8 = 1 : A8-15

CND : Condition for absolute branching (when specified by OPC).

(conditions same as for format 0)

MD : Addressing mode (Table 1-2).

R2 : Register (scratchpad A0-15) of 2nd operand or address of 2nd operand:

bit 14 = 0 : A0-7

bit 14 = 1 : A8-15

S : Store bit for memory reference instructions:

0 = store result in R1

1 = store result in memory

1.65 Addressing

Format 0 instructions (type 8) address the operand with the R1 field; no second operand is used. Format 1 instructions address the first operand with the R1 field; the second operand is addressed according to the addressing type, T1-T7, as listed in Table 1-2.

Table 1-2 Addressing Types

Type	Format 1 (K00)				Effective Address of Operand	
	MD 9 10 (K9 K10)	11	12	13 14		
T1	0 0	x	x	x x	R2	Register-to-Register. R2 contains the operand.
T2	0 1	0	0	0 0	P	Long Constant. The following word(after the instruction)is the operand.
T3	0 1	non-zero		$\overline{(OR2)}$	(R2)	Address in Register. R2 contains the address (A0-A15) of the operand.
T4	1 0	0	0	0 0	(P)	Address in Next Word. The following word is the operand address.
T5	1 0	non-zero		$\overline{(OR2)}$	(P) + (R2)	Indexed Address. The following word, indexed by (A0-A15), as specified by R2, contains the operand address.
T6	1 1	0	0	0 0	[(P)]	Indirect Address. The following word specifies the location containing the operand address.
T7	1 1	non-zero		$\overline{(OR2)}$	[(P) + (R2)]	Indirect Indexed Address. The following word, indexed by (A0-A15), specifies the location containing the operand address.
T8	Format 0 ($\overline{K00}$)					Short Constant. No 2nd operand is used.

1.66 All addressing uses a 16-bit address word, although only the 15 high-order bits are used for memory selection and for word-handling instructions. For character-handling instructions, the least-significant address bit specifies the character, as follows: bit 15 = 0: left character ; bit 15 = 1: right character.

1.67 OPERATION SEQUENCES

1.68 Microprograms, Microinstructions

CPU operations are controlled by microinstructions (μ Inst) located in the Microinstruction Store (Control ROM). Each microinstruction comprises a single 48-bit word divided into 14 command fields.

1.69 A microprogram performs one part of an instruction (e.g. indirect addressing or execution) or one operating sequence (e.g. Initial Program Loading or an Interrupt routine). The microprogram may consist of a single microinstruction (Fetch, SUK, etc.) or a group of microinstructions accessed in a specified sequence. This sequence may vary according to conditions specified in the microinstructions. An example of microinstructions and microprograms grouped into an instruction is shown in Figure 1-7, Operation Terminology.

1.70 Flow Diagrams

General operational flow for the CPU is shown in Figure 1-8. Detailed flow diagrams for each block are referenced on Figure 1-8 and, for the execute-instruction sequences, in the instruction list (Table 1-2). A key to the flow diagrams is provided in Figure 1-7.

1.71 Figure 1-8 shows the general sequence of computer operations. The computer continuously performs microinstructions. The sequences in which the microinstructions are performed are determined by microinstruction addressing (paragraph 2.26).

1.72 In no program is running and the computer is not being operated, it cycles through the Idle loop. RUNF is described with the PSW, paragraph 2.90. Machine-state-pointer control is described with the microinstruction addressing. RUNF is set by pressing START on the control panel. If the control panel is used (Section III), PUP is set in the CPU (logic diagram CC) and the operation now cycles through the control-panel path, beginning with microinstruction /010. Either automatic restart or IPL may be initiated through this path.

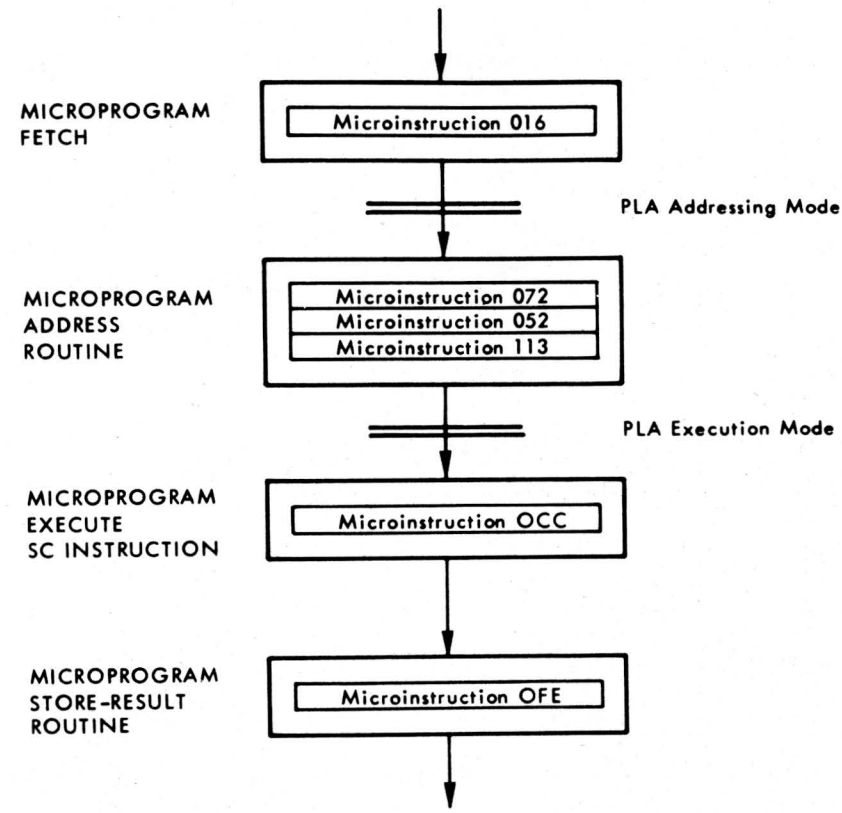
1.73 If RUNF is set and the control panel is not selected, the machine-state-pointer tests for interrupts, and executes the interrupt routine if necessary. KRY is set when the K-register is loaded with the instruction word. KRY is set by the Fetch command during the Fetch routine, some control-panel routines, EX instruction, and some tests.

1.74 Before KRY is set, the computer loops directly to Fetch to load the instruction word into the K register. The instruction must be requested from memory during a preceding microinstruction. When an instruction is loaded, KRY is set and the PLA addressing mode is used to select the next microinstruction address. For most instructions, the addressing routine is used to obtain the operand. Instruction word addressing and decoding is described in paragraphs 2.37 and 2.48.

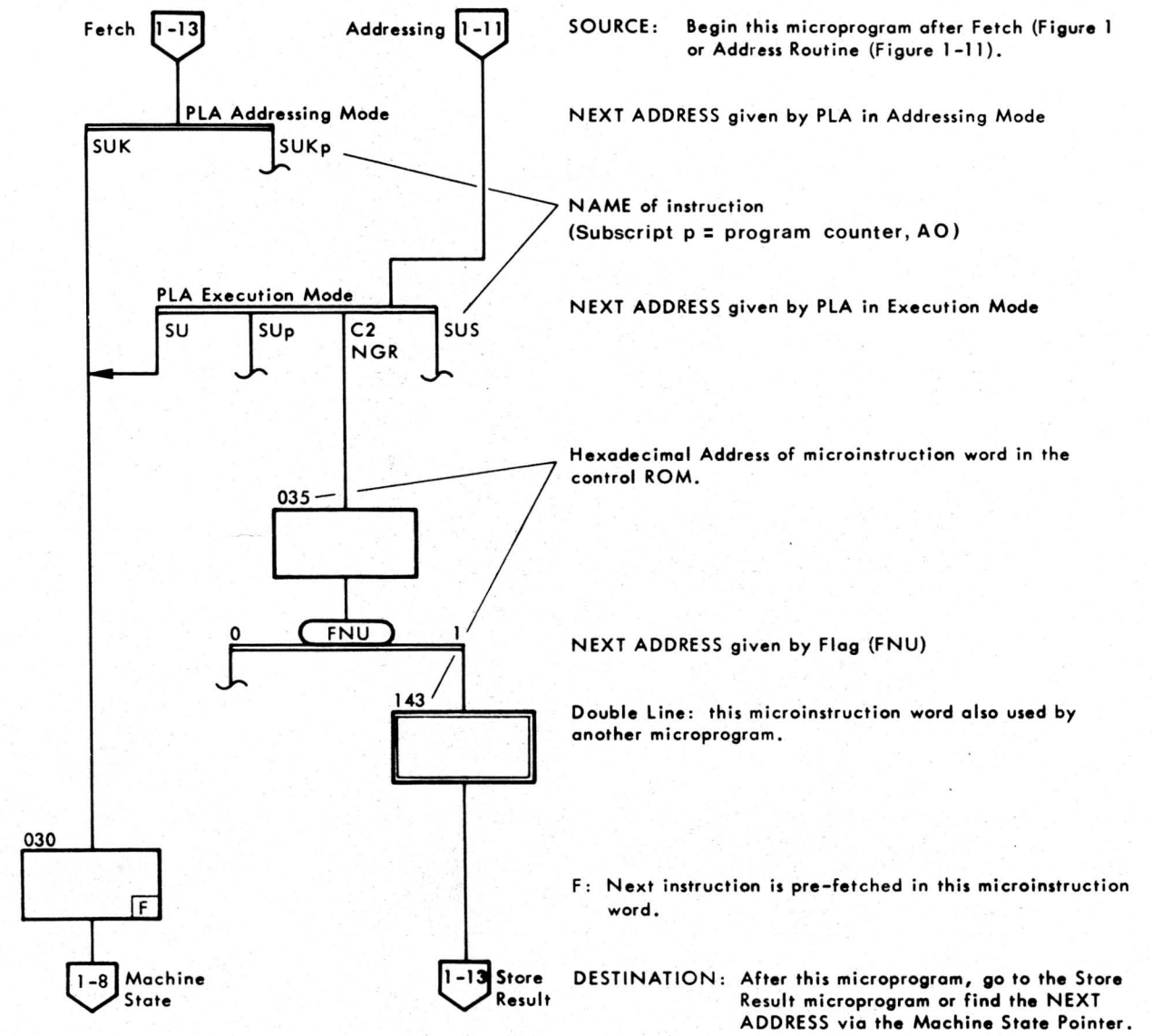
1.75 Repeated microinstructions of a routine are selected with explicit addressing (the microinstruction includes the address of the next microinstruction). Microprogram decision tests may modify the explicit addressing with a flag bit (SNA Flag mode). At the end of a microprogram routine, a Bus Request is made for the next instruction word and the Fetch routine is then used to load the instruction. If the machine state pointers allow (program still running, no control-panel operations or interrupts pending), the computer selects the next microinstruction address via the PLA addressing, and continues with the next microprogram.

1.76 Data paths are shown in Figures 2-1 and 2-4. The microinstructions within each microprogram or routine control the data paths and control the operations performed on the data. These controls are described (in Section II) for each of the logic blocks shown on the data-path diagrams. The data handling is also listed on the microprogram flow charts with terminology such as: $M + 2 \rightarrow M$, where the data in the M register loops through the data paths shown on Figures 2-1, 2-4 and back to M, with a left-shift being performed enroute to produce the +2.

Example of one instruction (SC-Store Character) using addressing mode T6 (indirect)



Part A



Address Type Designations	
T1 (2nd Operand in R2):	T2-7
RT1	RT2-7 Result → R1
RT1P - R2=0, thus operand in P.	RT2-7S - Store; Result → memory
RT1D - Double Length; (R2) → Q, (R2+1) → M	RT2-7C - Constant
RT1DP - Double Length and R2=0	T3A - R2 ≠ A15
	T3B - R2 = A15

Part B

Figure 1-7 Operation Terminology, Flow Diagram Key

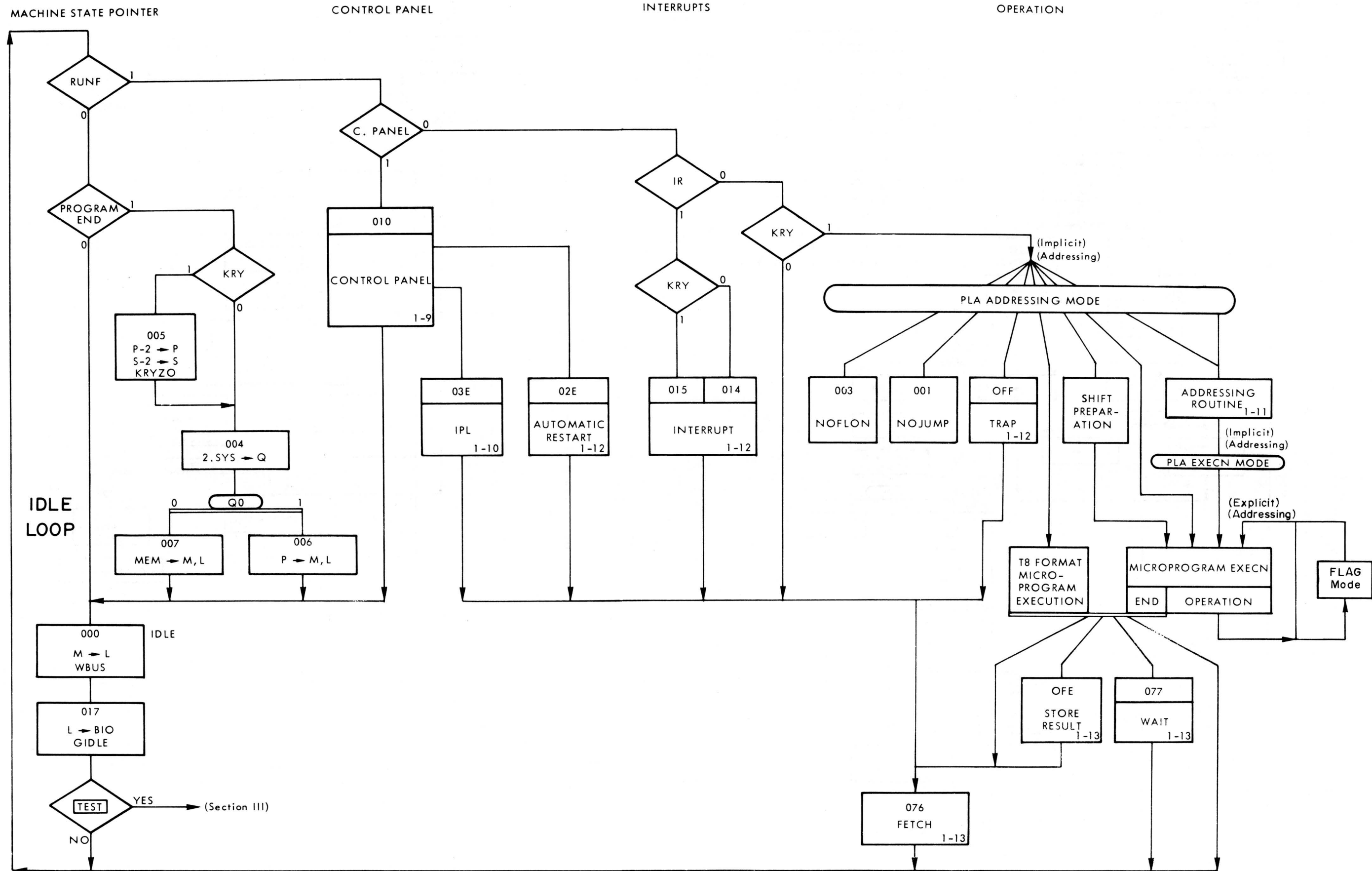


Figure 1-8 CPU Operational Flow and Machine State Pointer

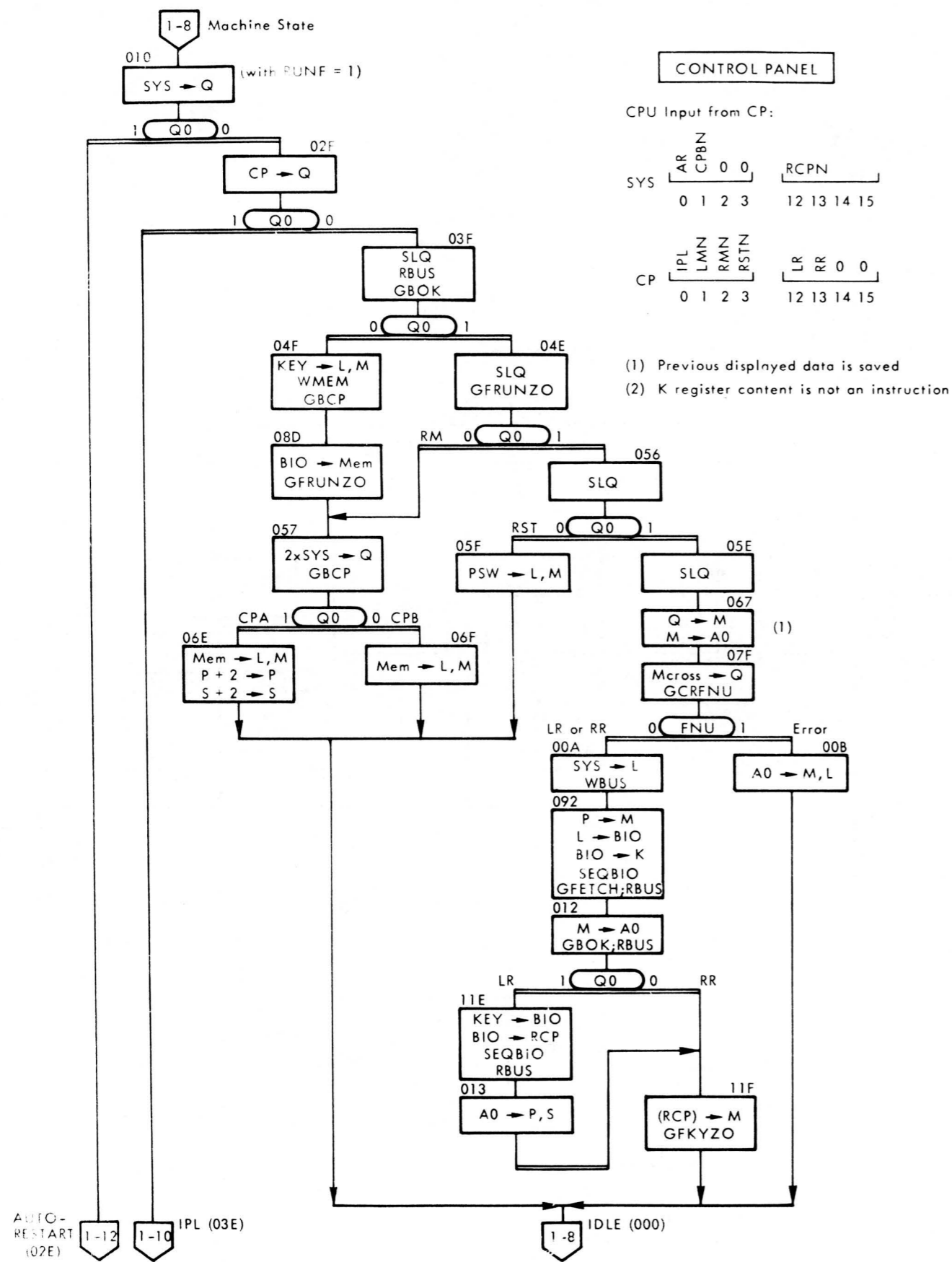


Figure 1-9 Control Panel

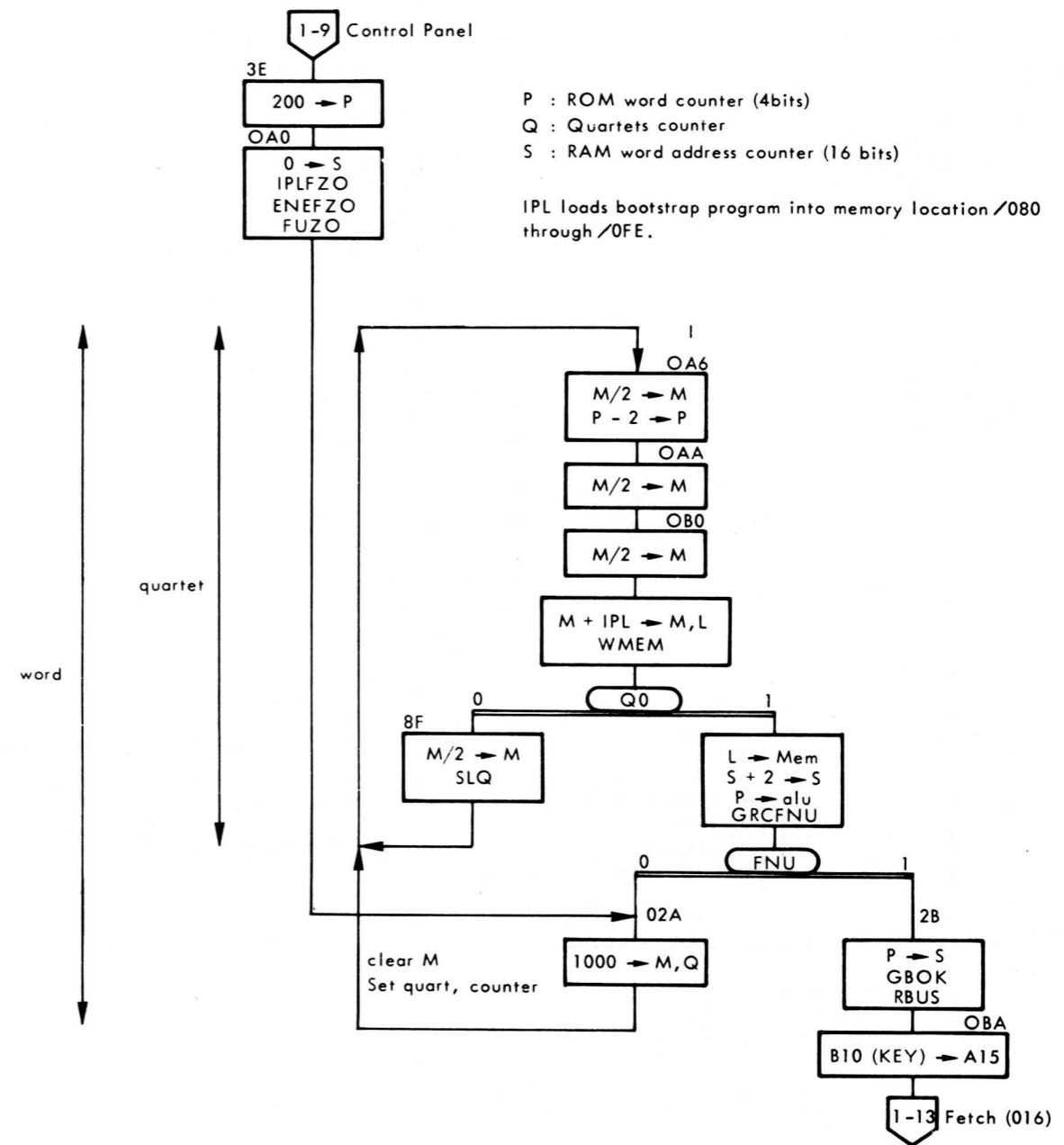


Figure 1-10 IPL

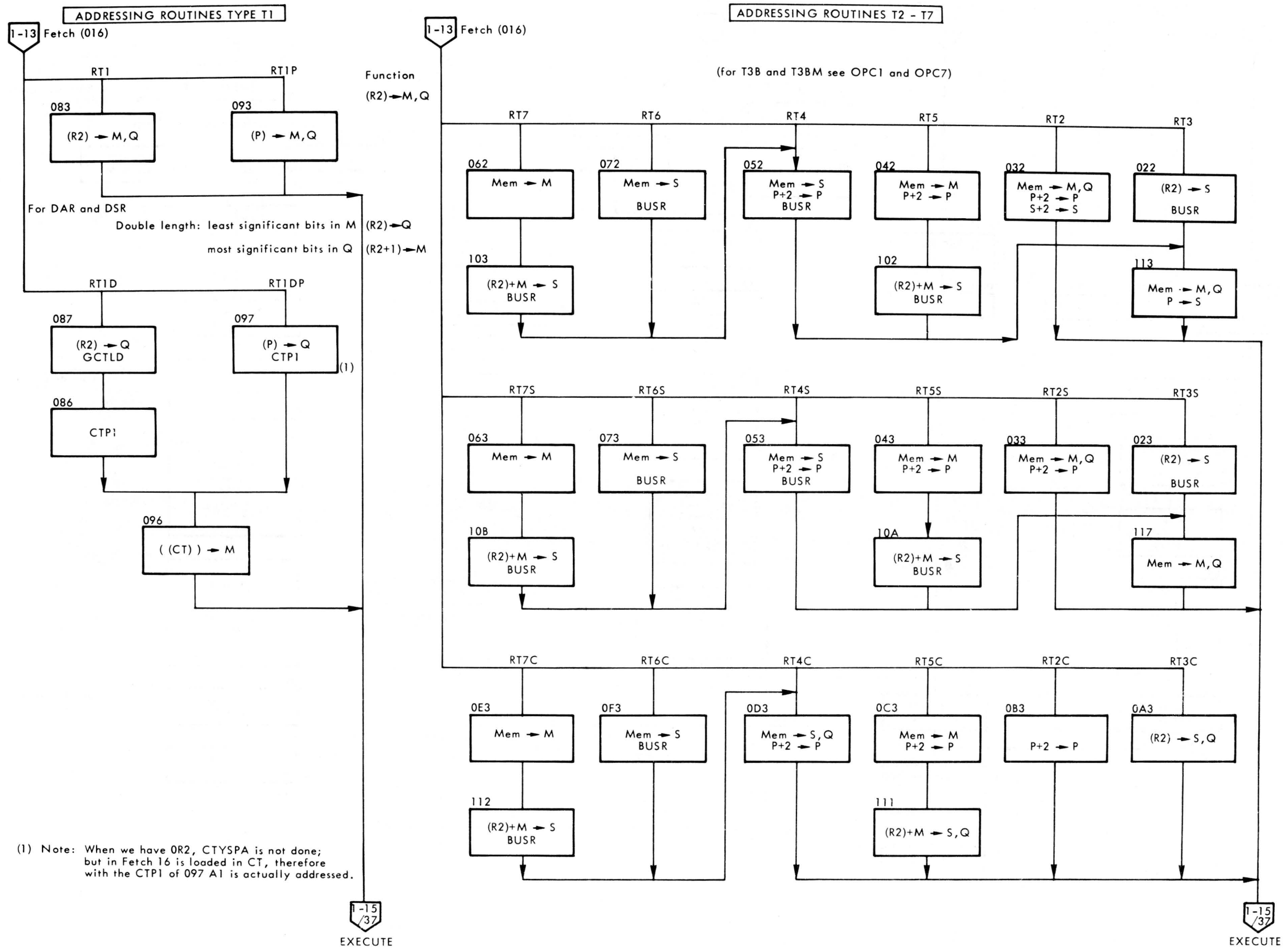


Figure 1-11 Addressing

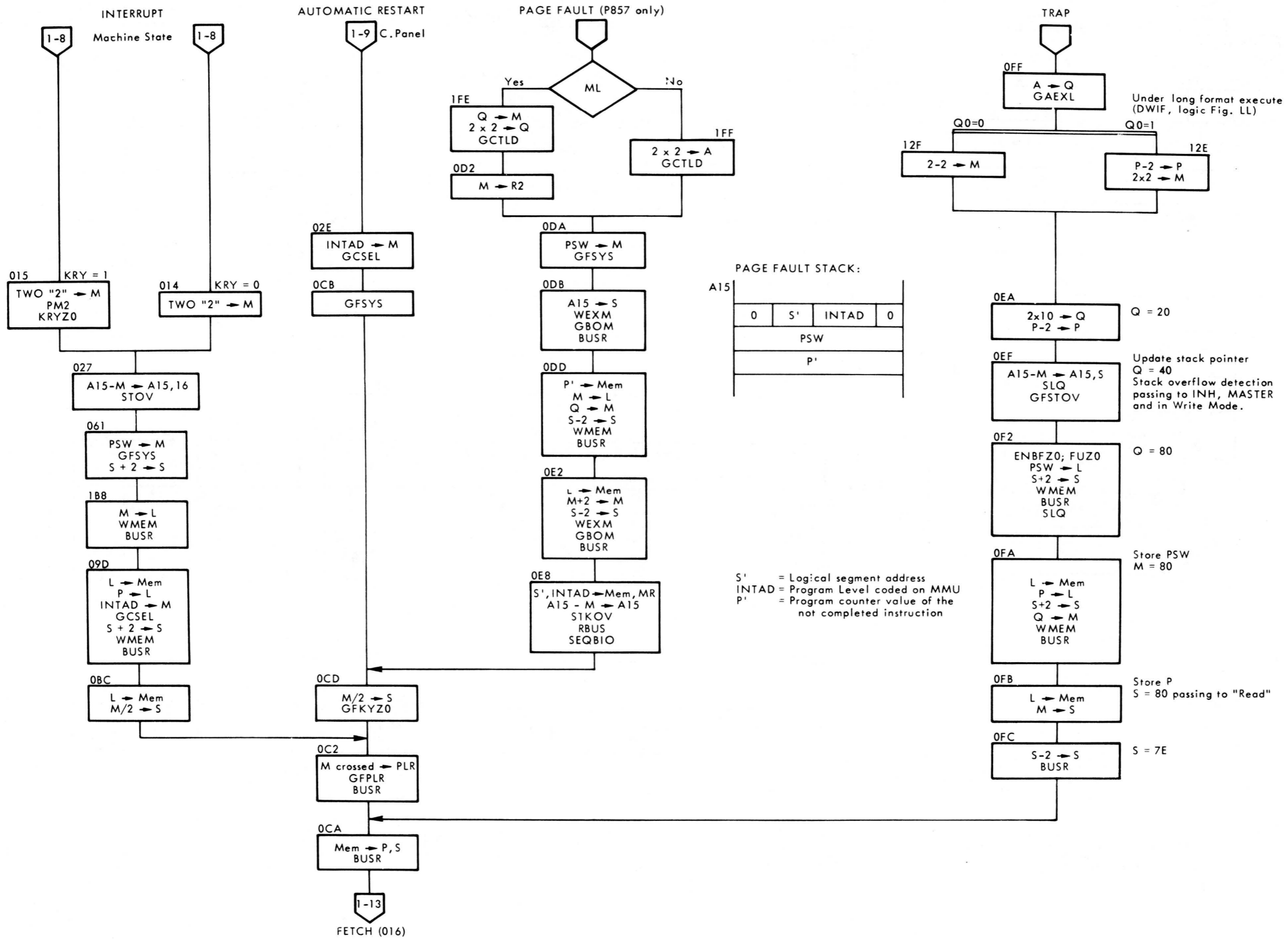


Figure 1-12 Interrupt, Restart, Fault, Trap

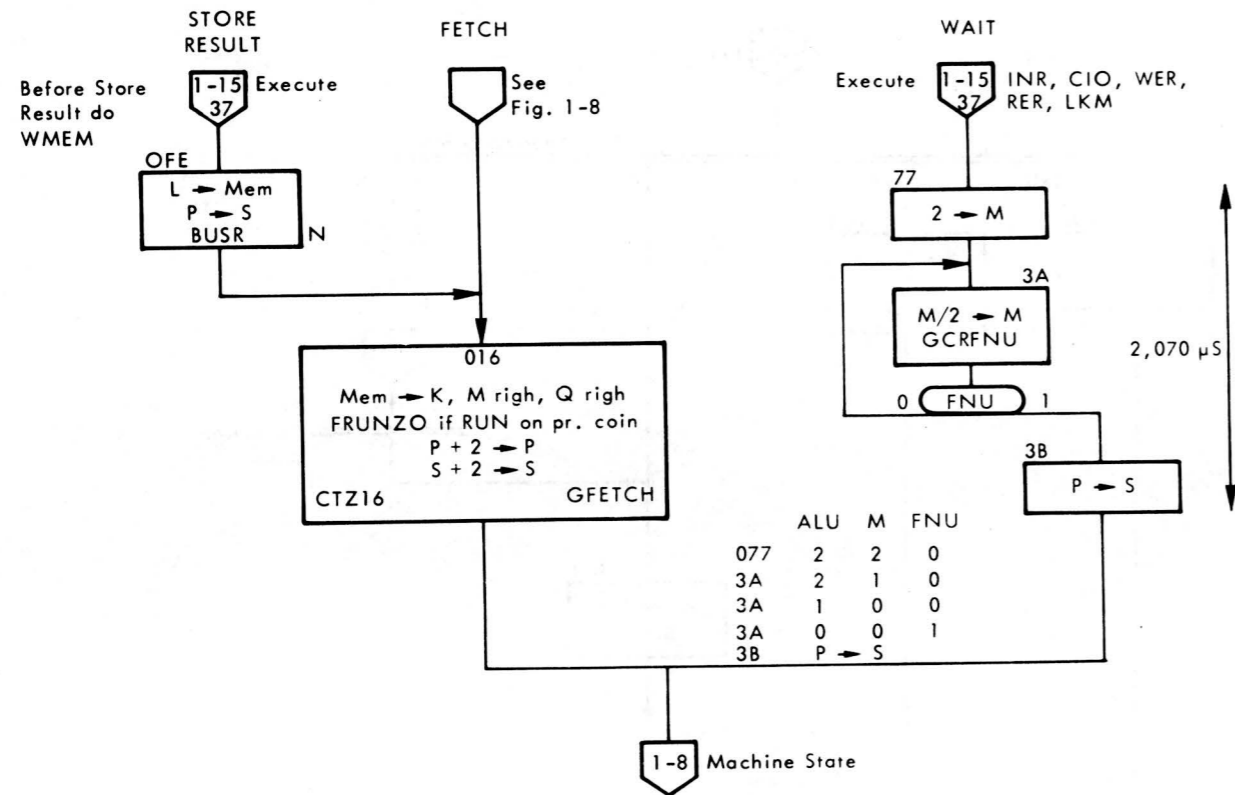


Figure 1-13 Store, Fetch, Wait

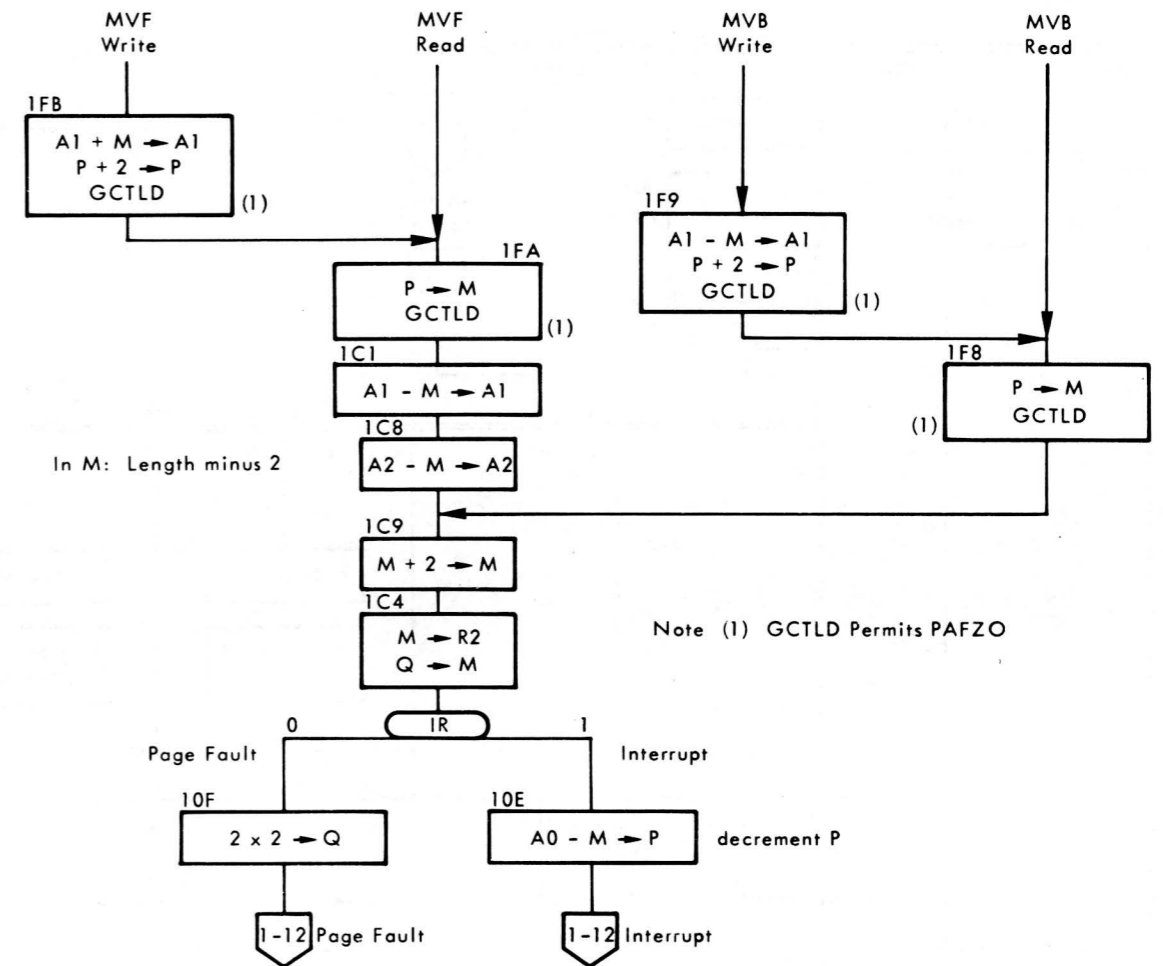


Figure 1-14 Shifting of Move Tables (P857 only)

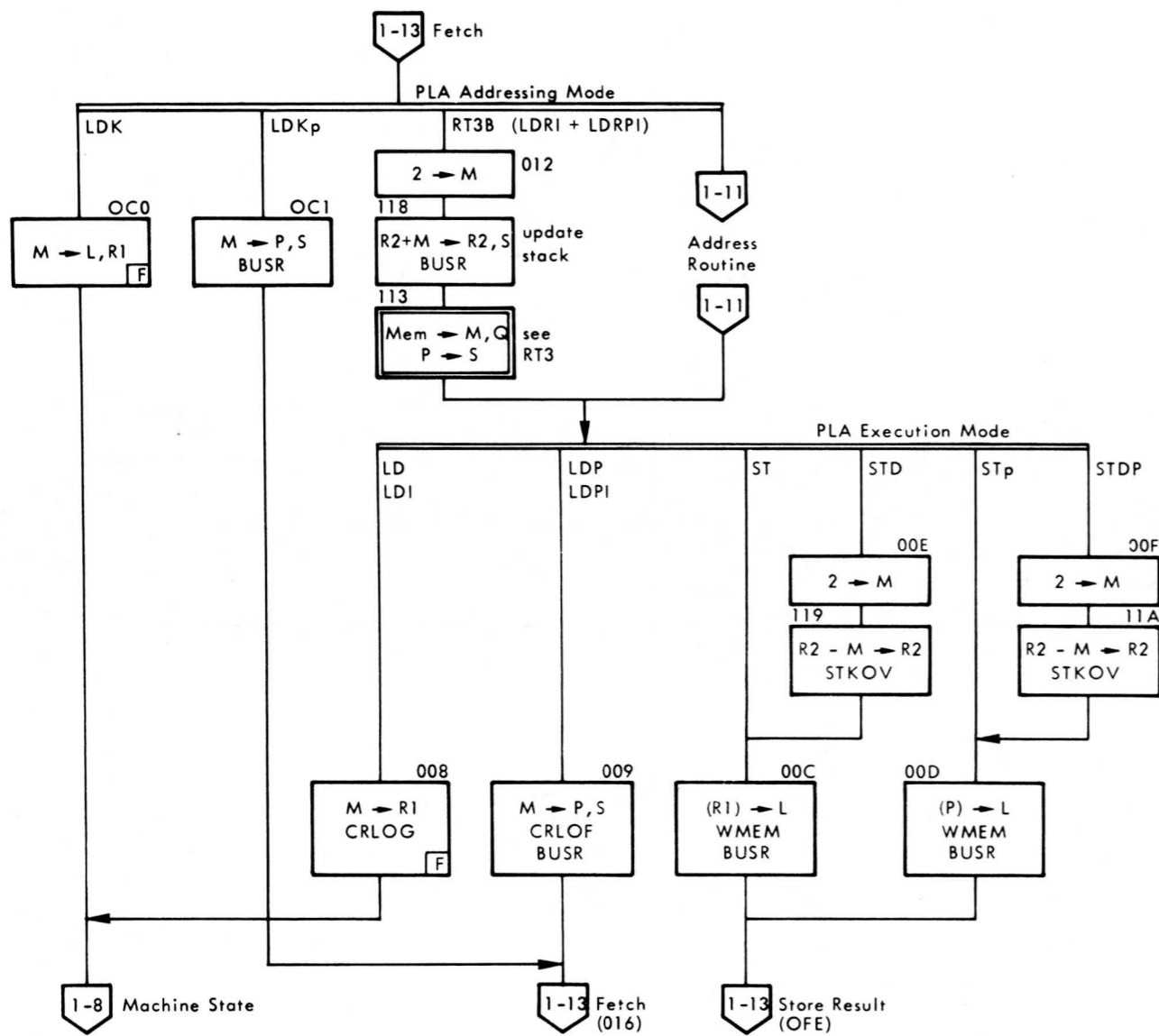


Figure 1-15 OPC 0 (LD, ST)

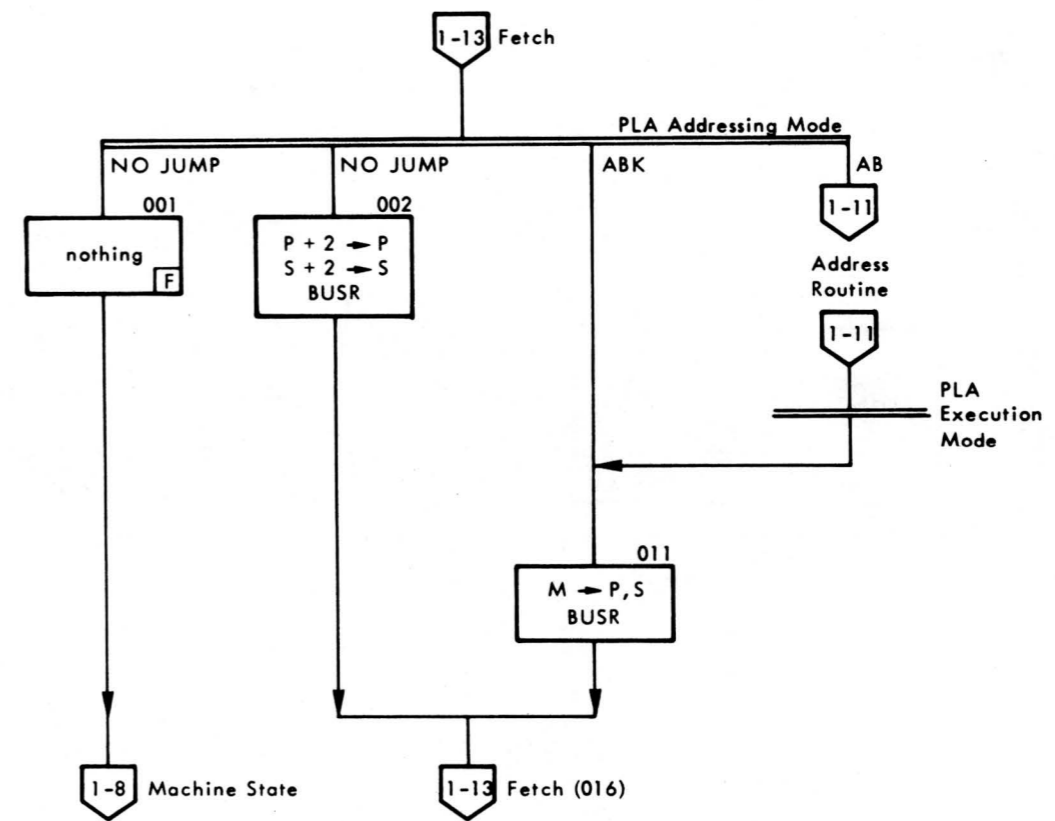


Figure 1-16 OPC 1 (AB)

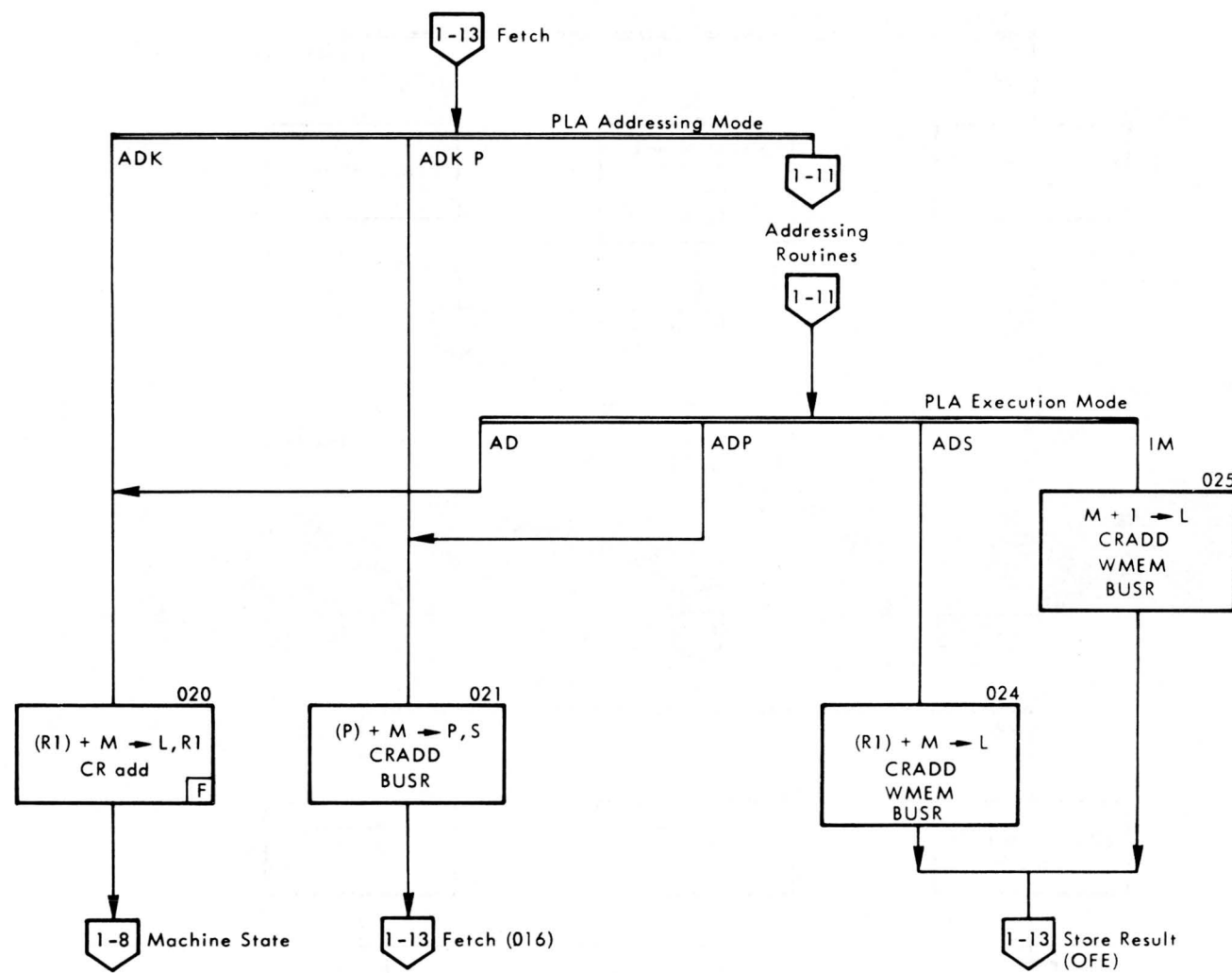


Figure 1-17 OPC 2 (AD, IM)

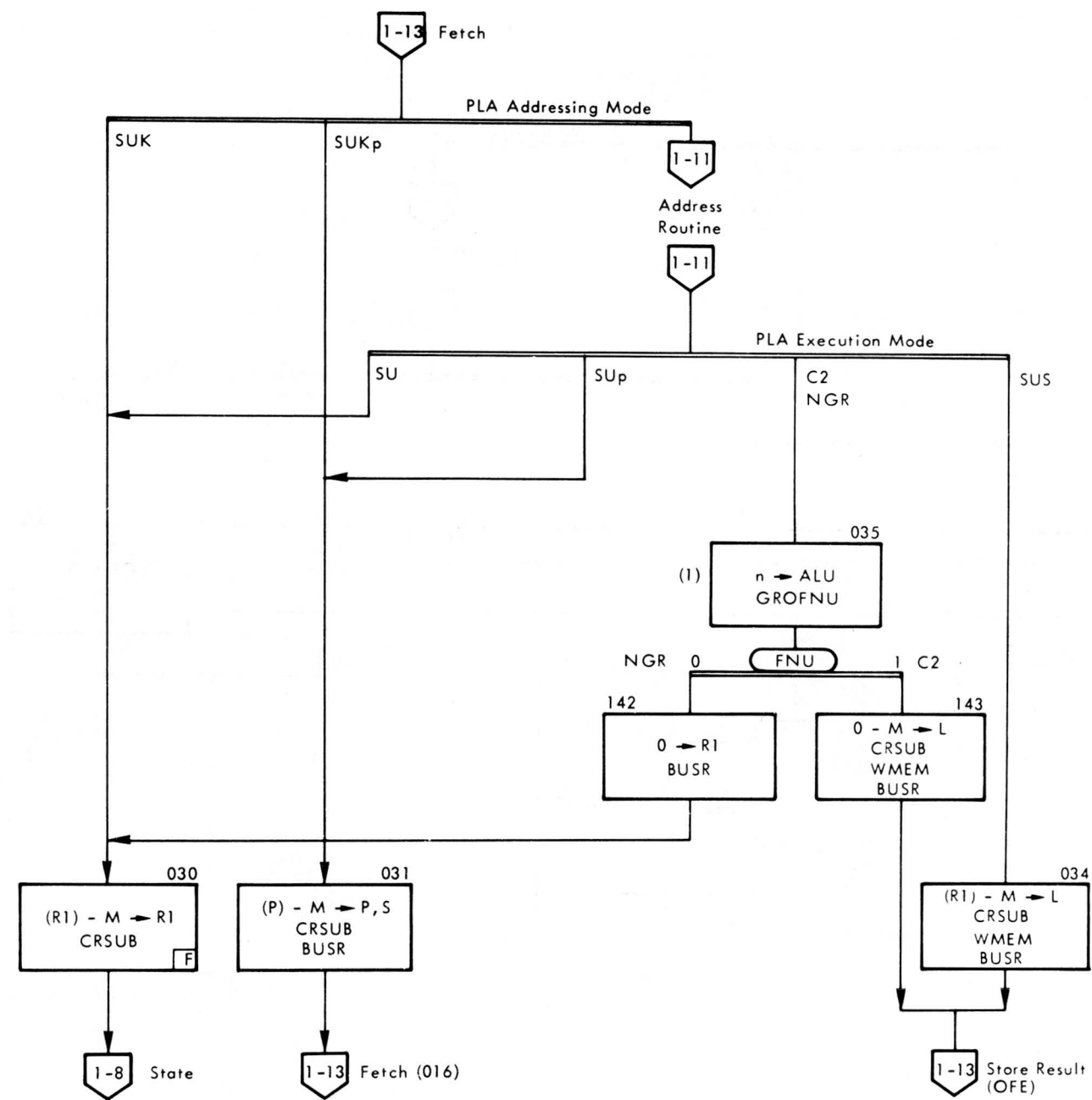
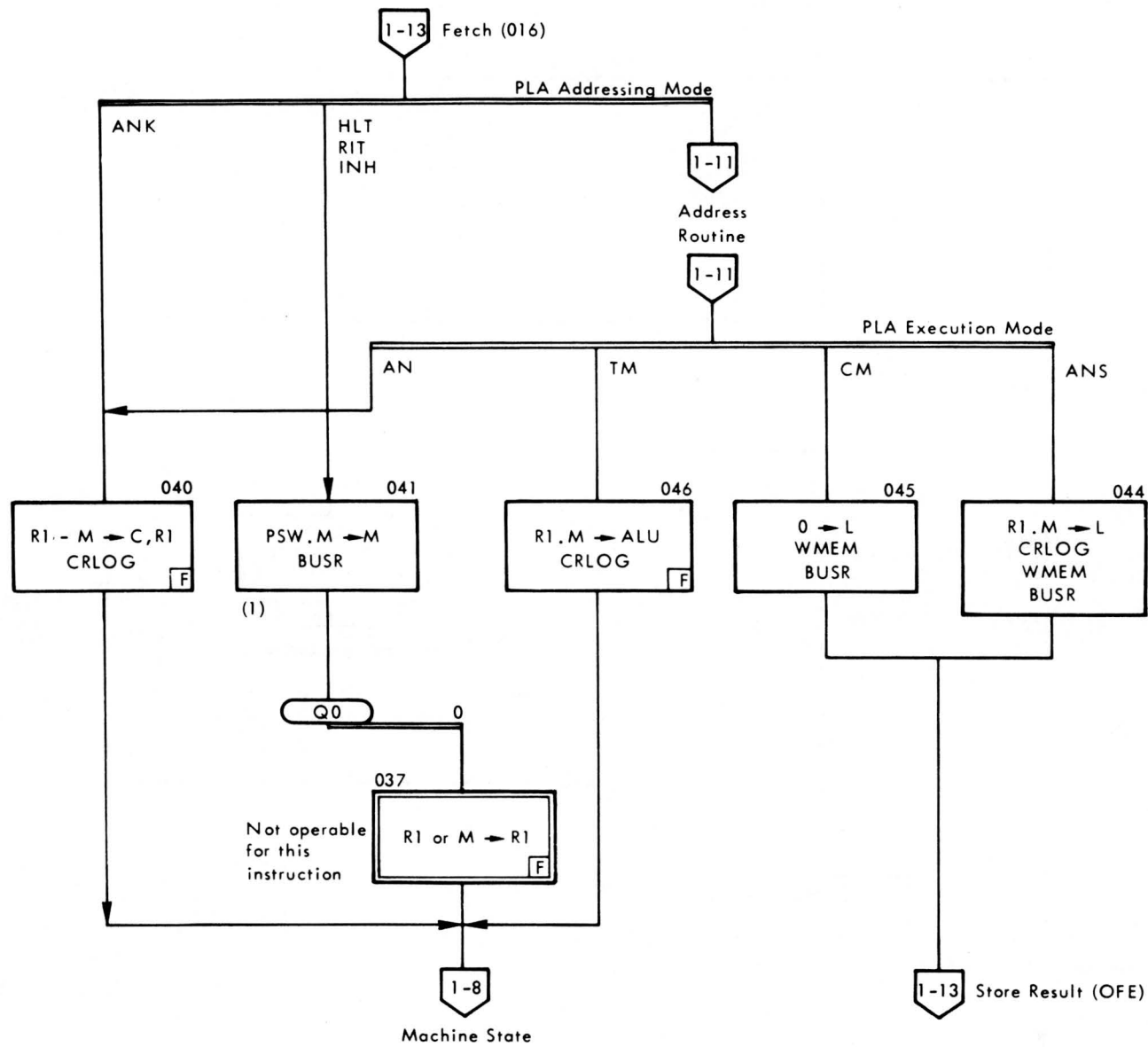
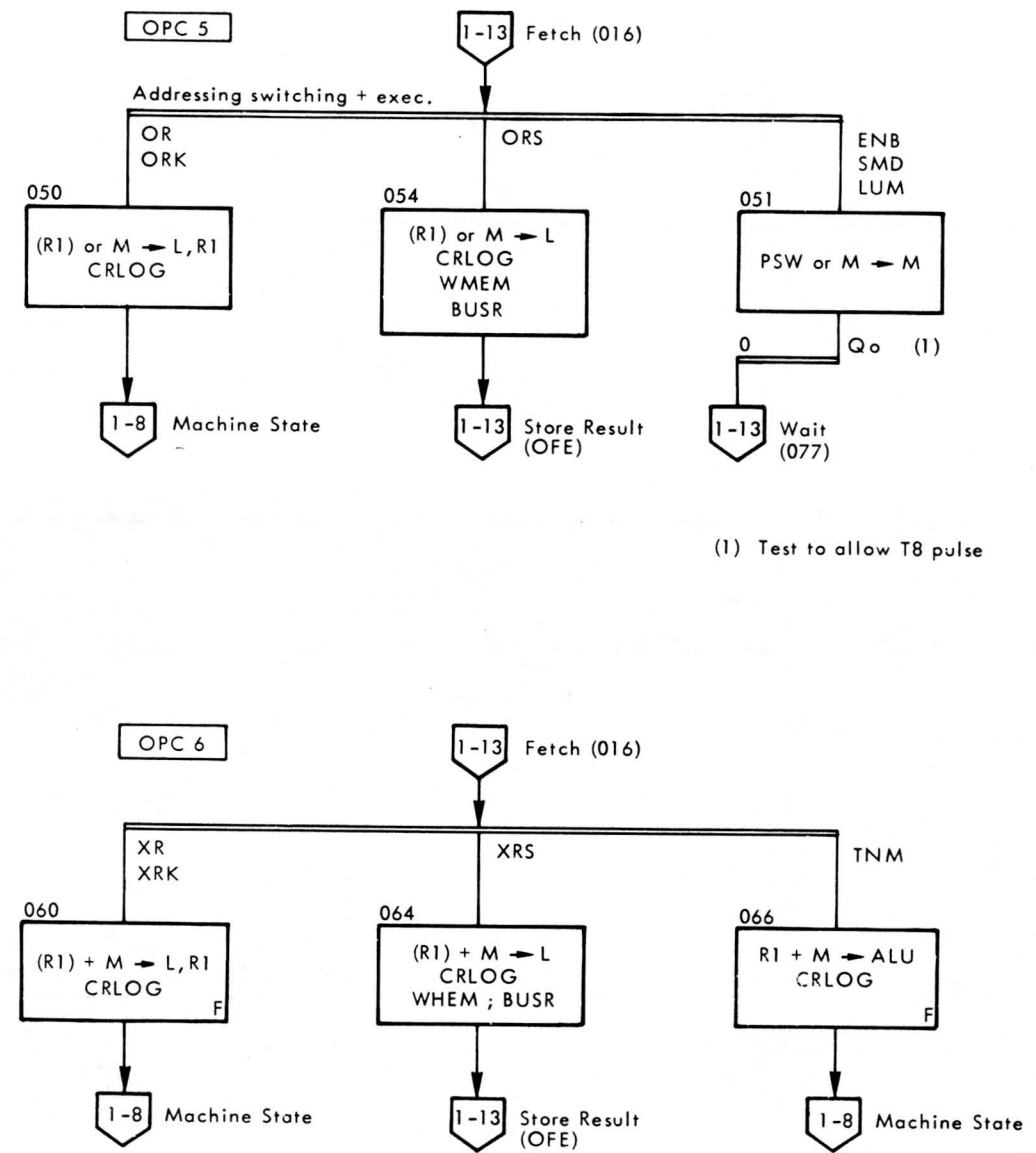


Figure 1-18 OPC 3 (SU, NG, C2)



(1) Note : - Test Q0 to have a T8 pulse CLG = REPSW, T8
 - M register is updated because PFF is reloaded by M11.

Figure 1-19 OPC 4 (AN, TM, CM, AC, HLT, INH, RIT)



(1) Test to allow T8 pulse

Figure 1-20 OPC 5, 6 (OR, ENB, LKM, SMD, XR, TNM)

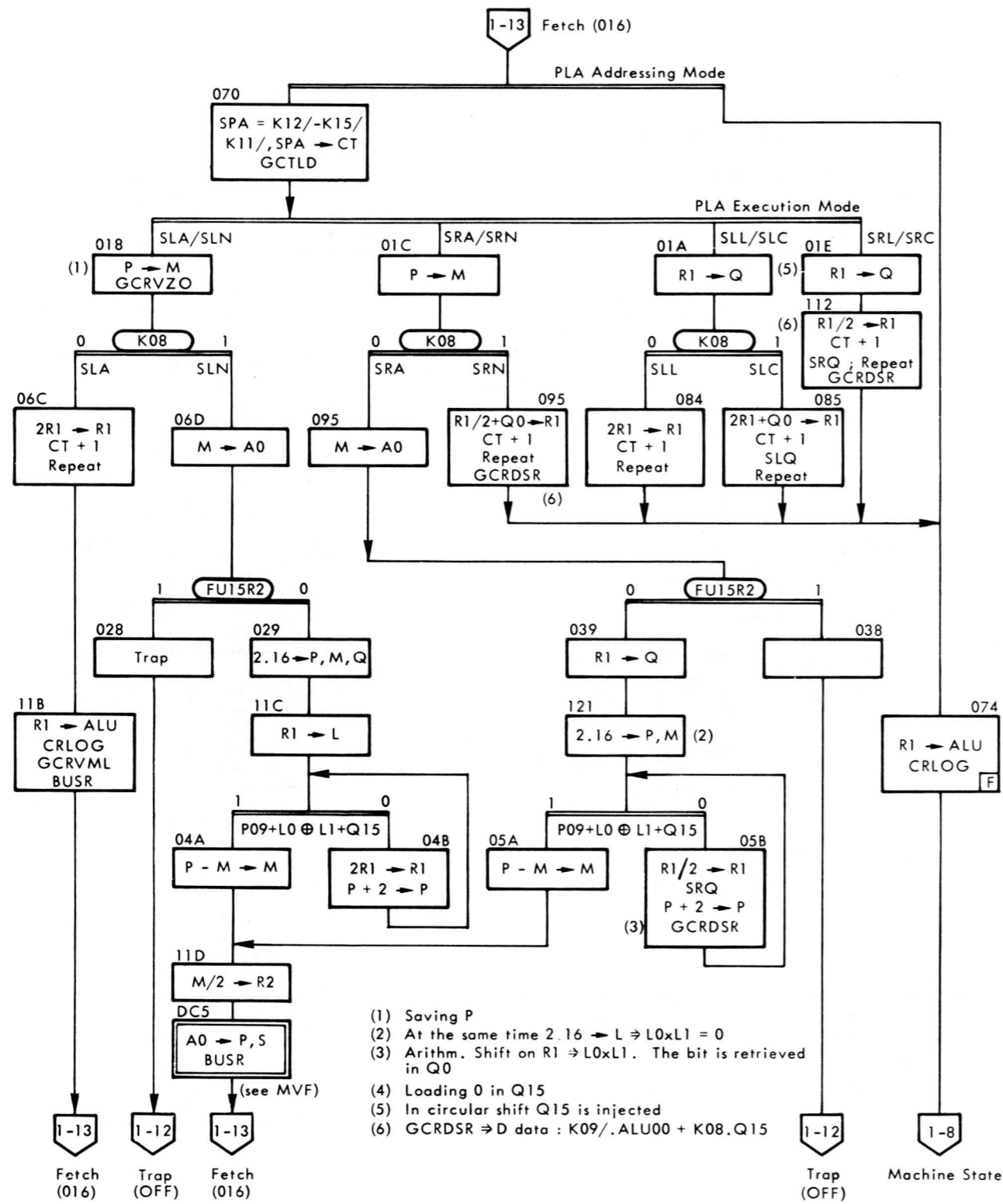


Figure 1-21 OPC 7 Single Shifts (SL, S R)

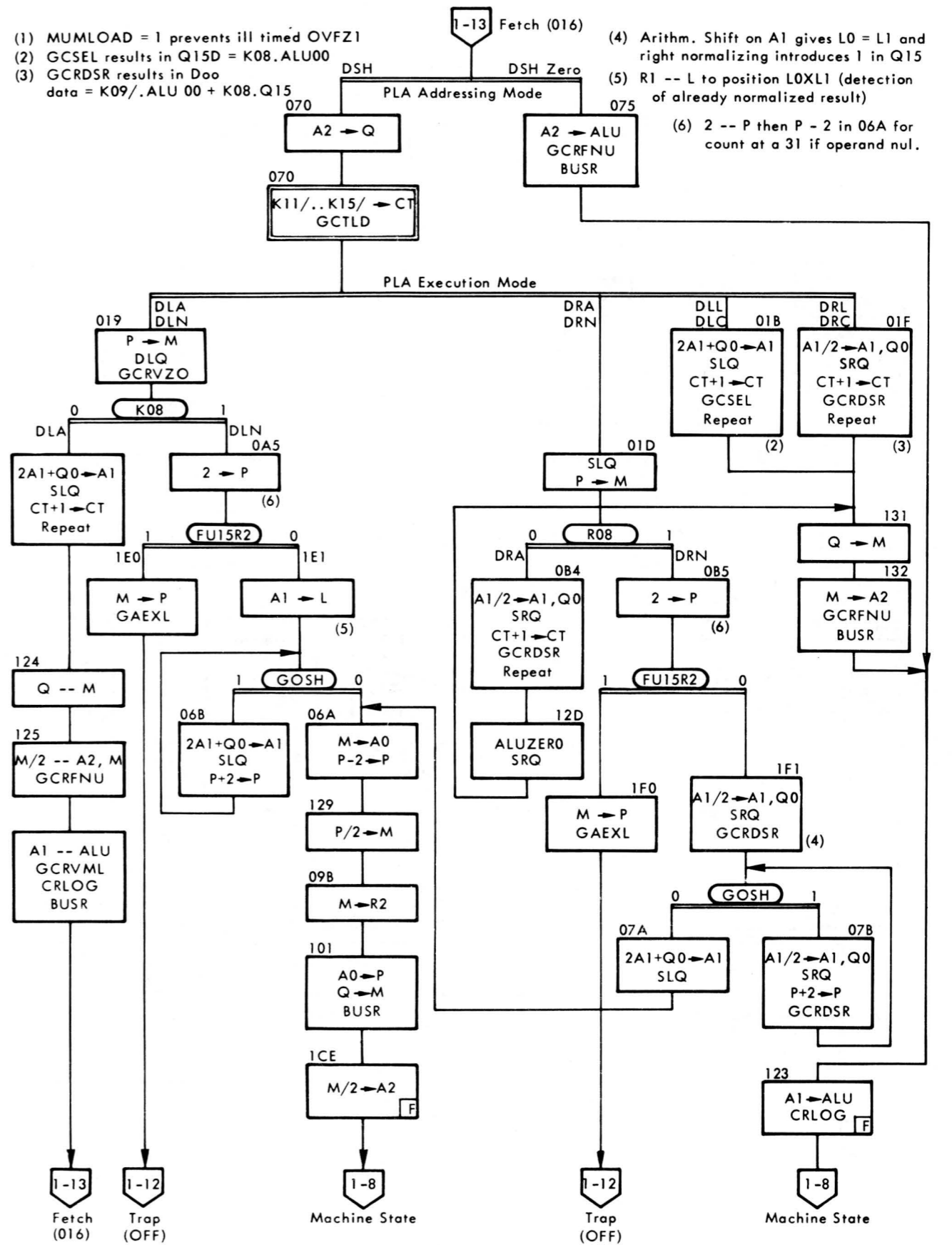


Figure 1-22 OPC 7 Double Shifts (DL, DR)

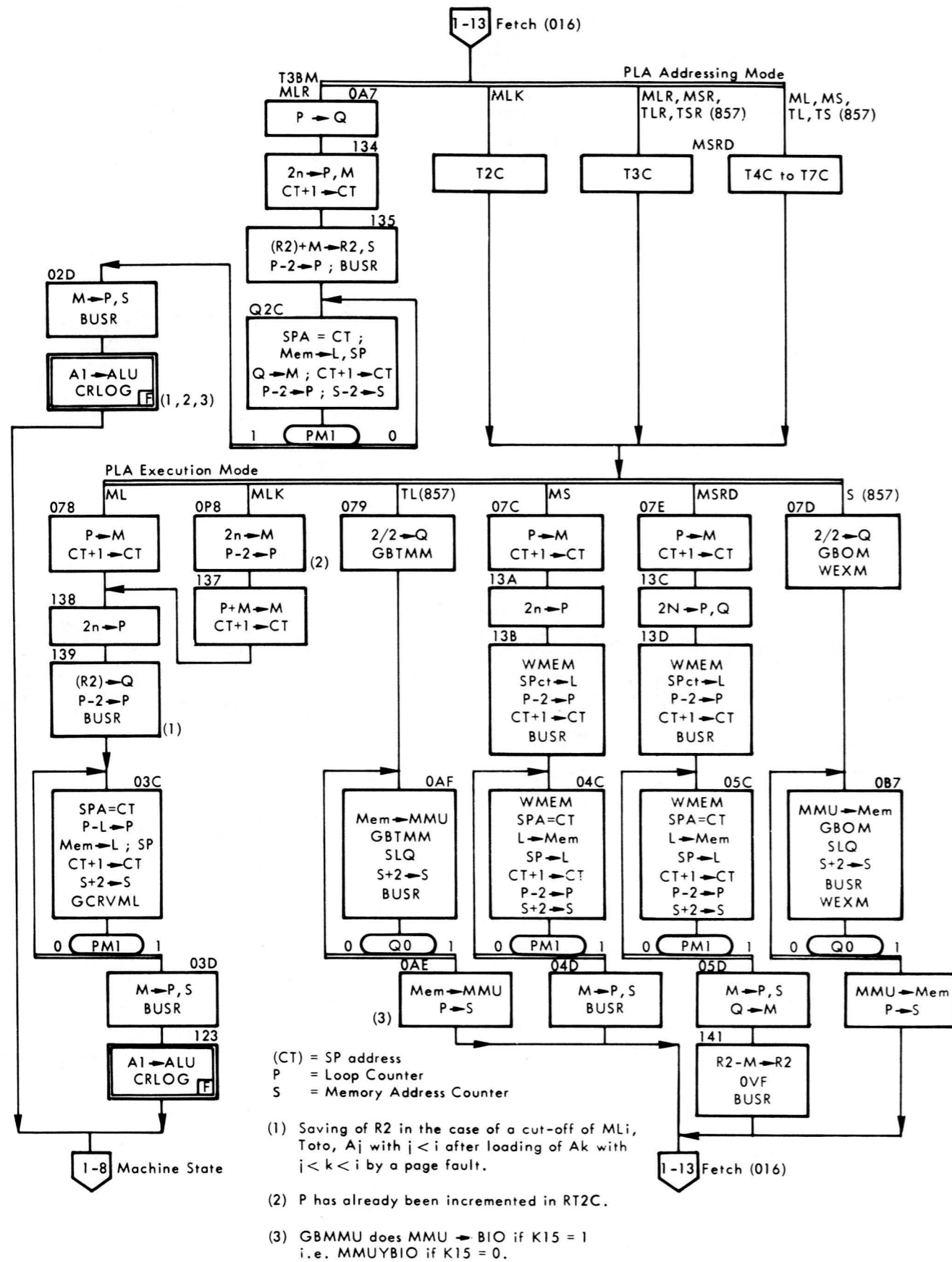
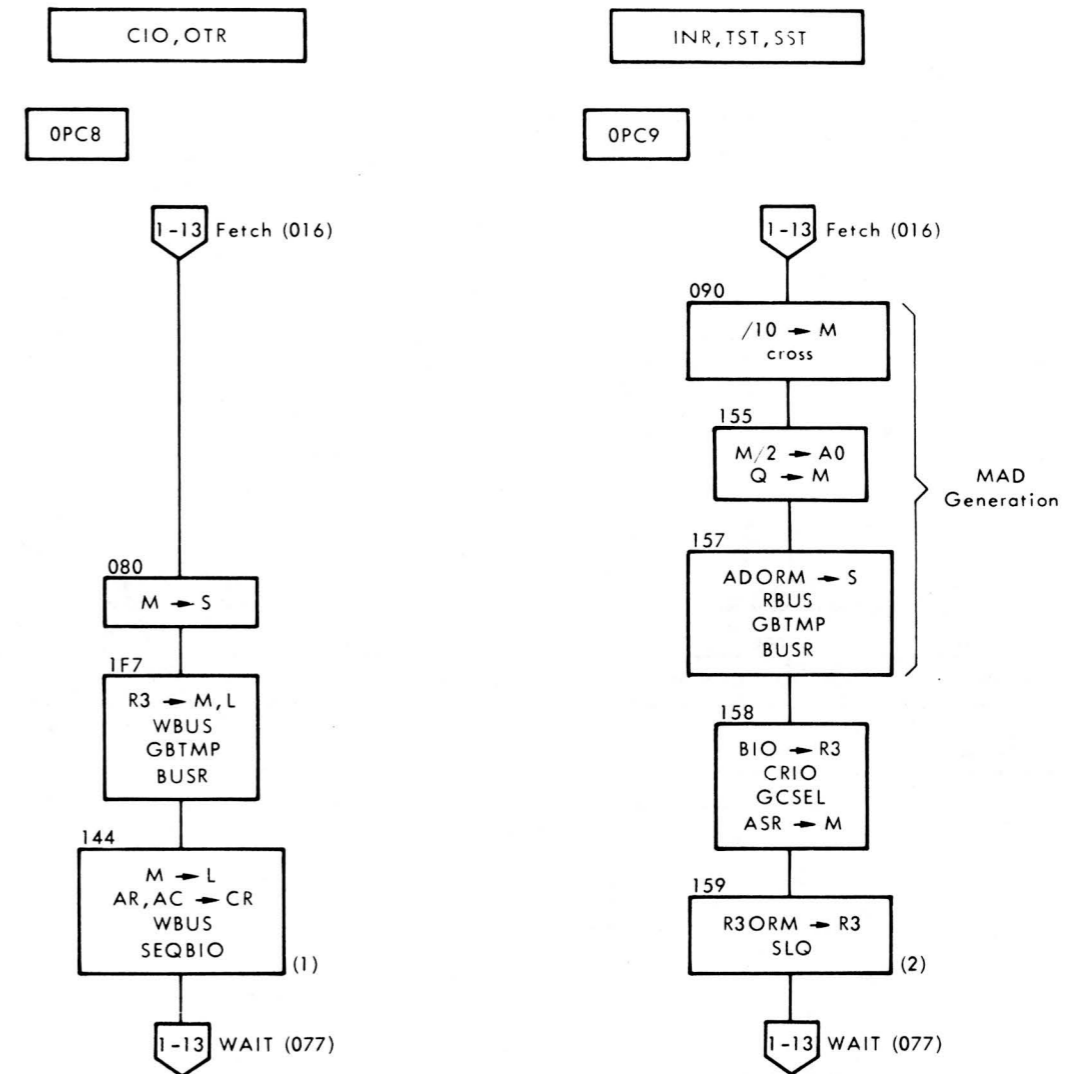


Figure 1-23 OPC 7 Multiple Load/Store, Table Load/Store (ML/MS, TL/TS)



- (1) - M \rightarrow L and WBUS maintain BIO for 120 ns after ending edge of TMR. In the same cycle SEQBIO inhibits the BSYCPU resetting on T6.
- WBUS sets WRITE to "one"; so in INR, SST, TST, WRITE = 0 and in CIO, OTR, WRITE = 1.
(2) SLQ for MUQ1 = 1 permits clock echo if FACINR.

Figure 1-24 OPC 8, 9 I/O Instructions

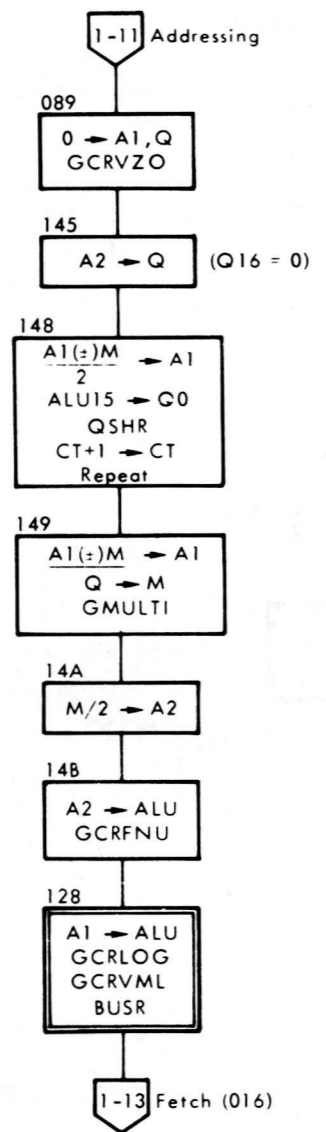
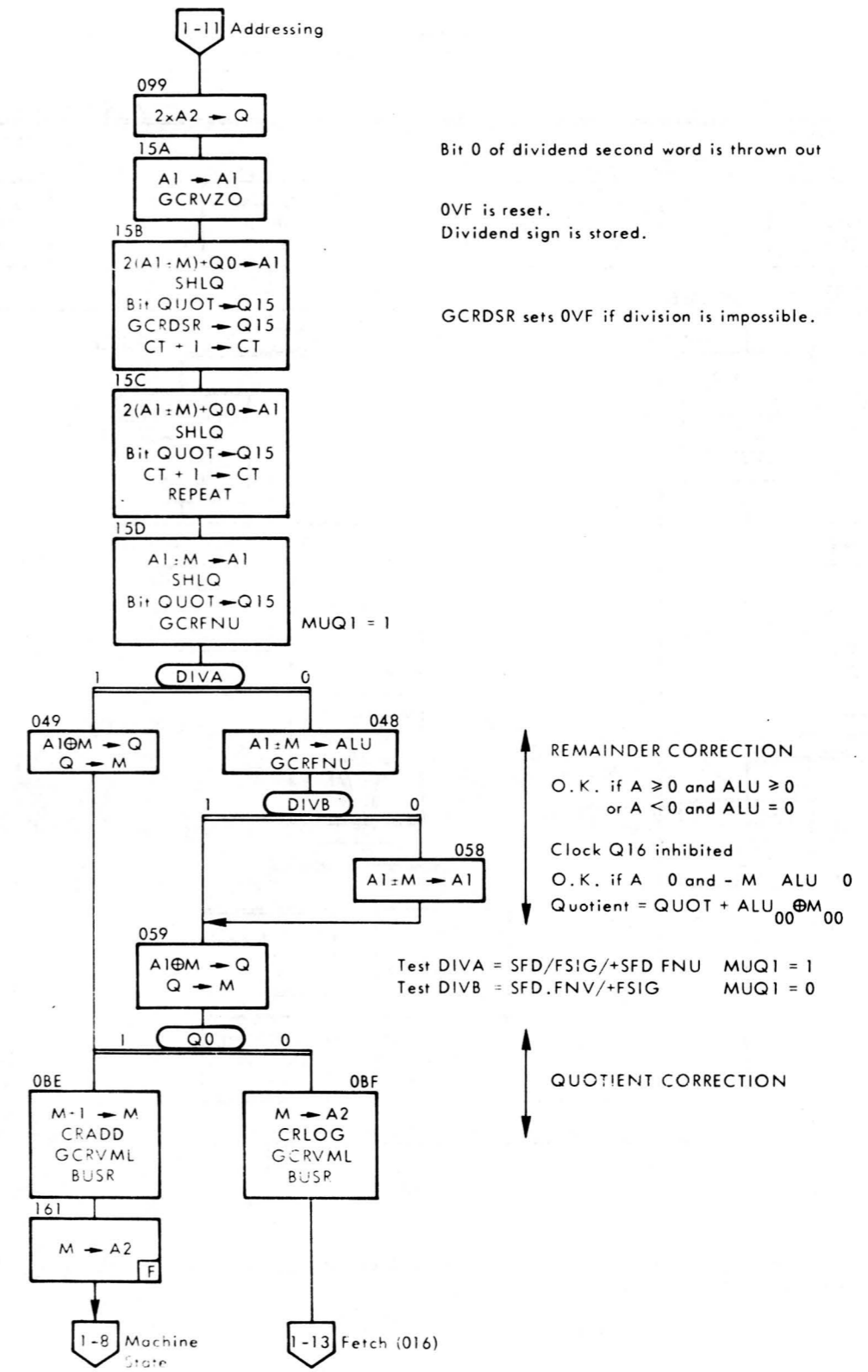
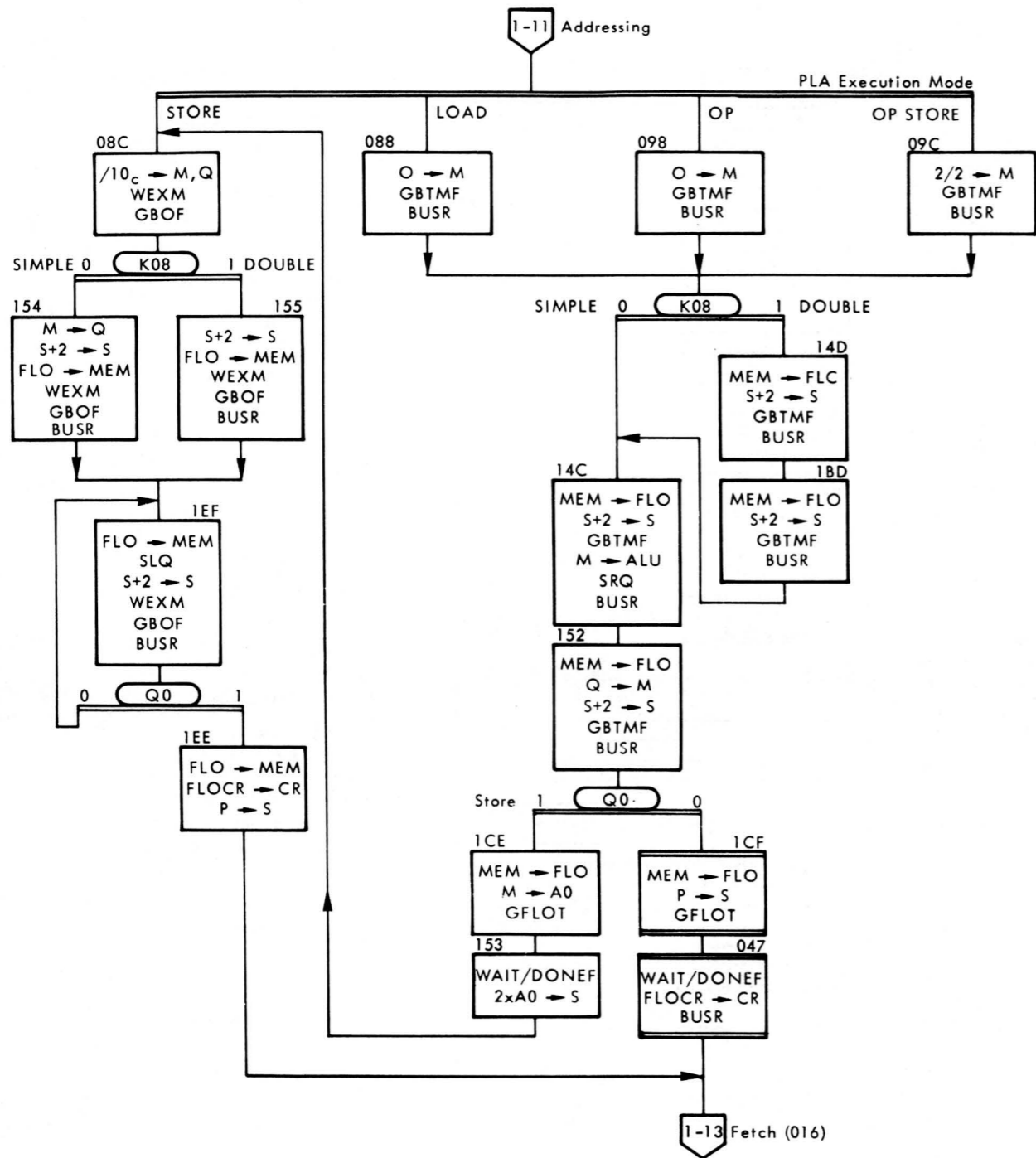


Figure 1-25 OPC 8 Multiplication (MU)



1-26 OPC 9 Division (DV)

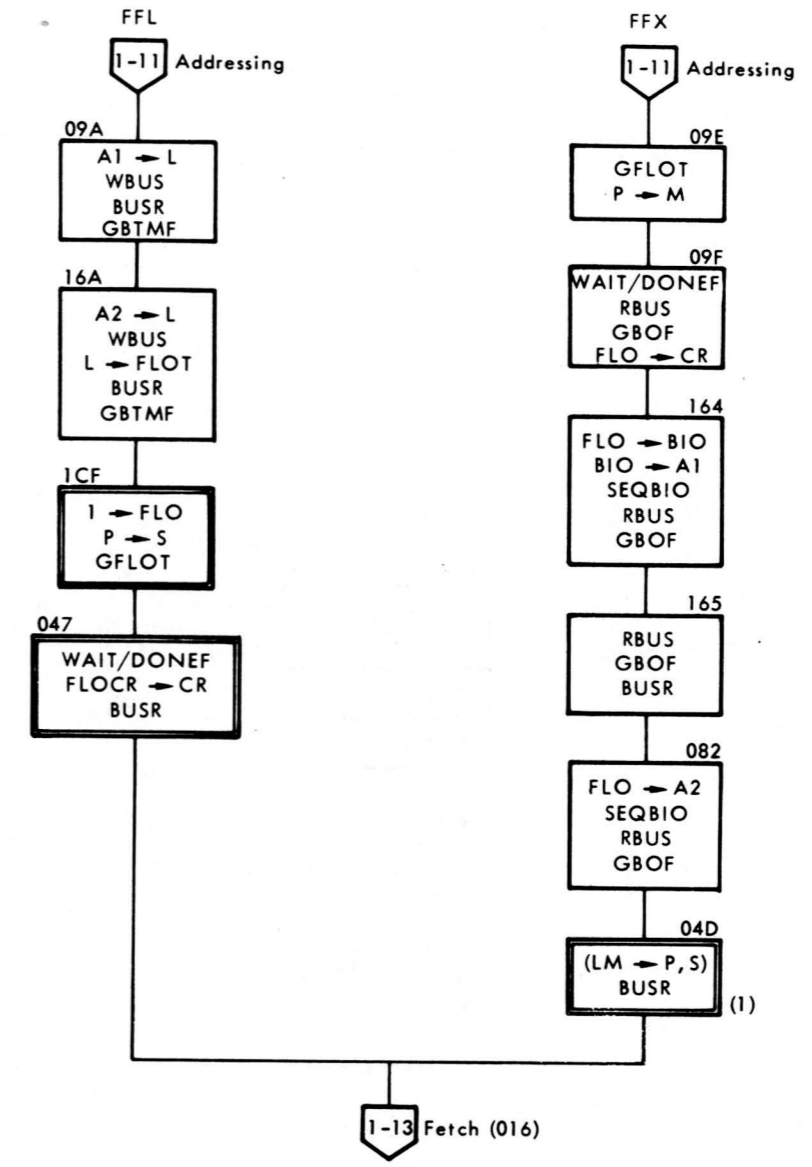


After the addressing routines, the effective address of the operand is in Q register (see routines type "C")
 (1) contents of Q are now:

S	AD
---	----

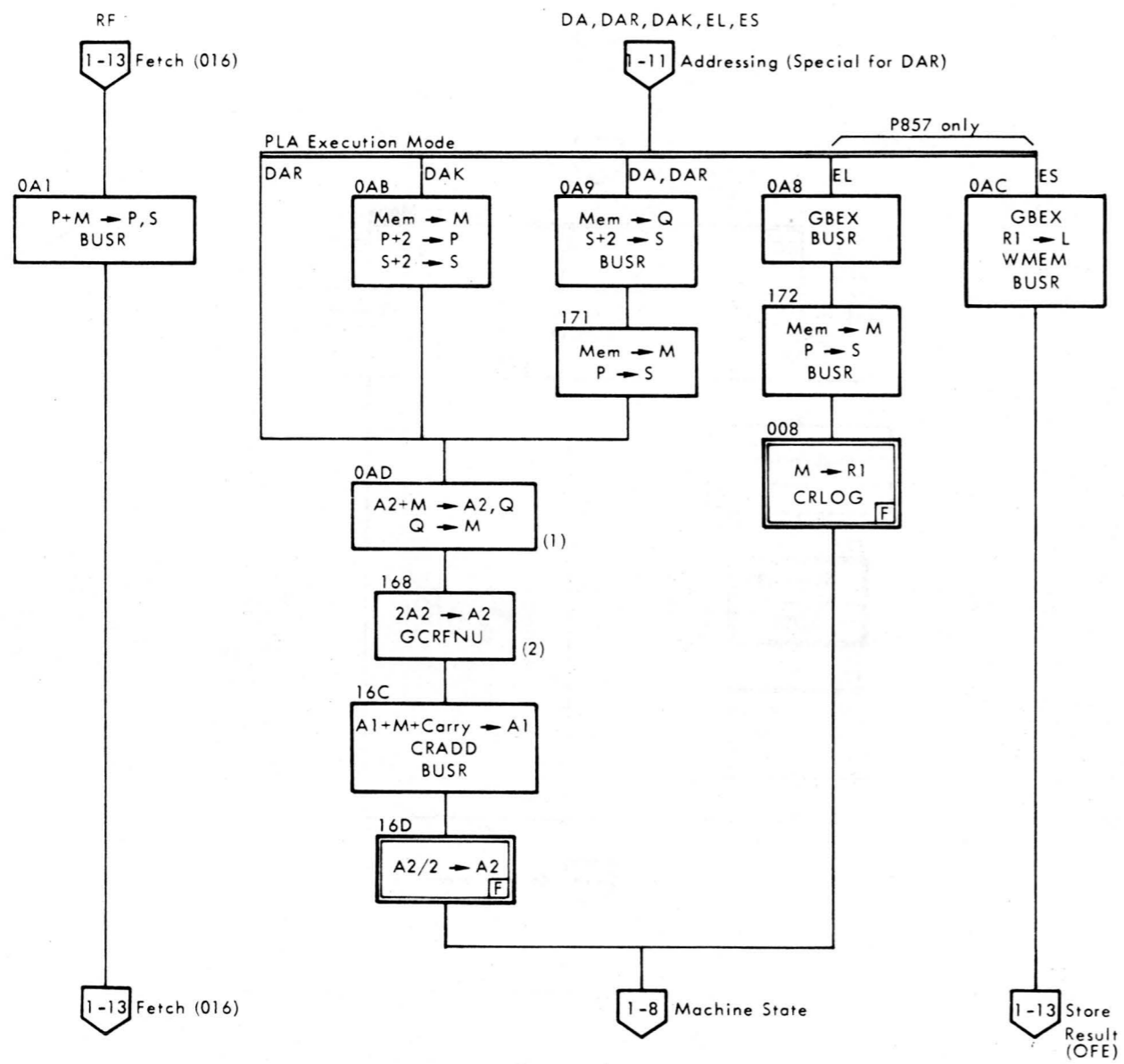
 S = Store flag
 AD = Address (right shifted)

Figure 1-27 OPC 8, 9 Floating Point OP, OP Store, Load Store (FA, FS, FM, FD, FSD, FST) P857 only



(1) This location is necessary to reset BSYCPU - M → P, S needs the transfer P → M in 09E

Figure 1-28 OPC 9 Floating Point Conversions (FFL, FFX), P857 only



- (1) Carry in Q00
- (2) FNU correctly positioned on 15 bits of the second word.

Figure 1-29 OPC10 (RF, DA, DAR, DAK, EL, ES)

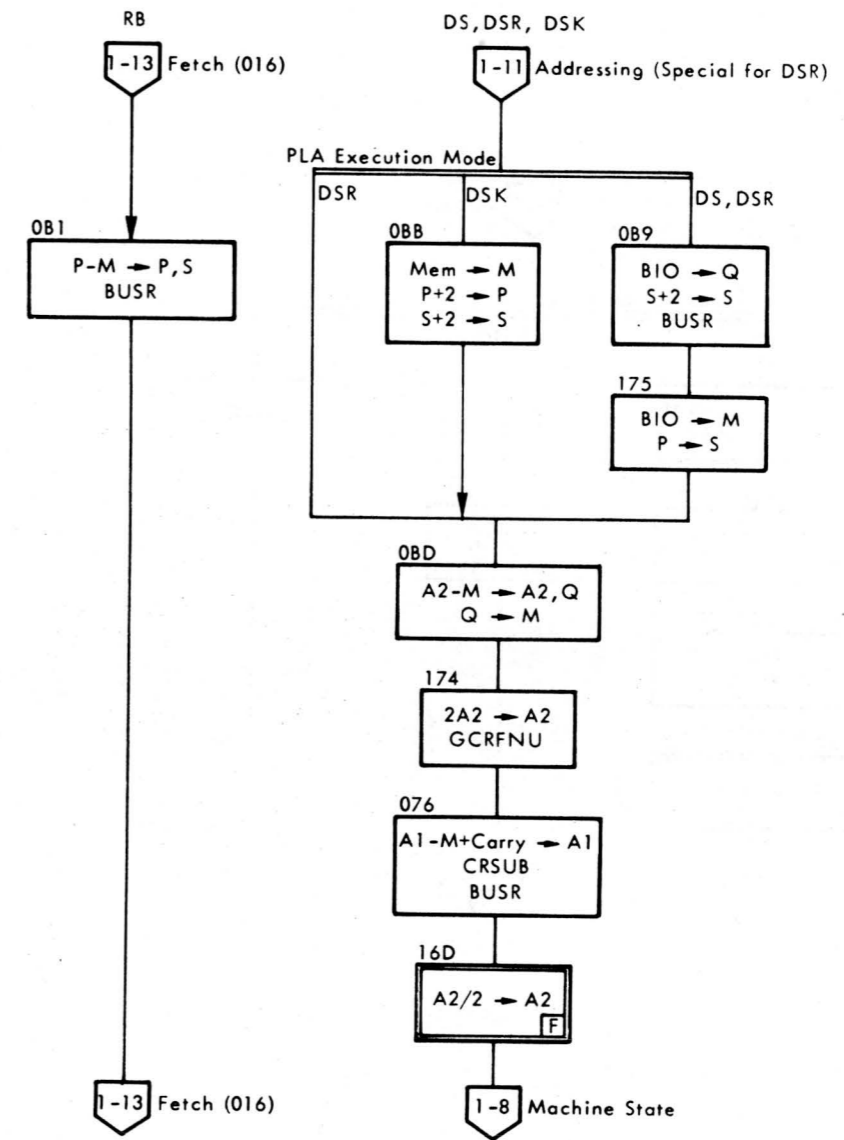


Figure 1-30 OPC 11 (RB, DS)

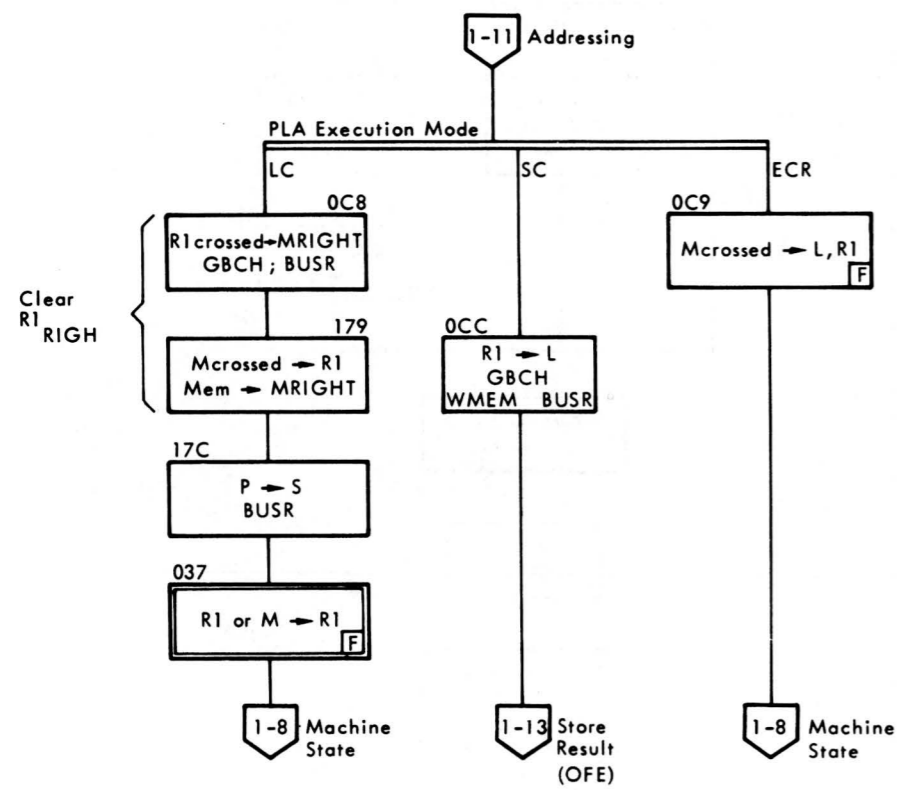


Figure 1-31 OPC 12 (LC, SC, FCR)

1/9/75

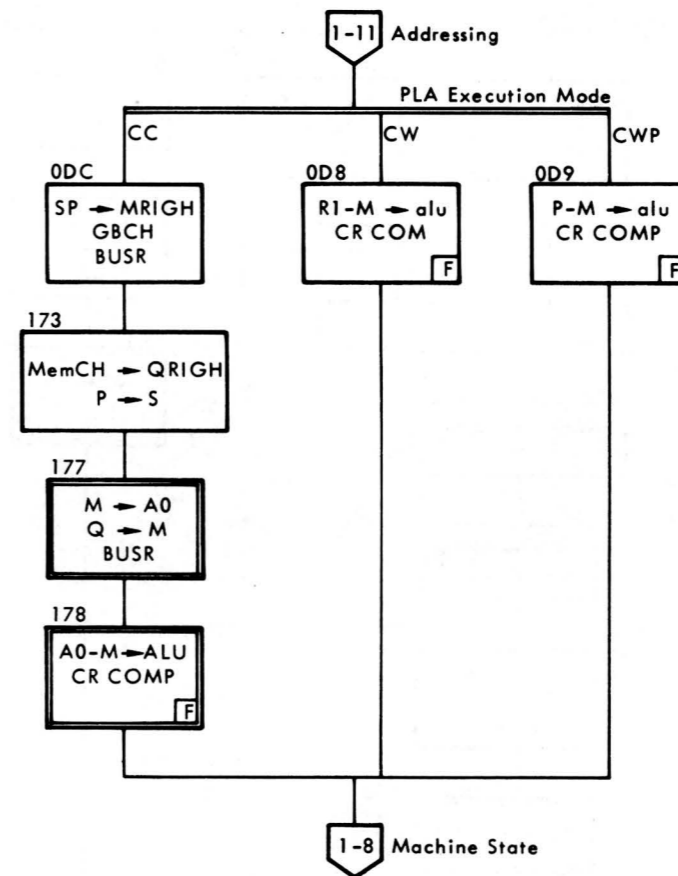


Figure 1-32 OPC 13 (CC, CW)

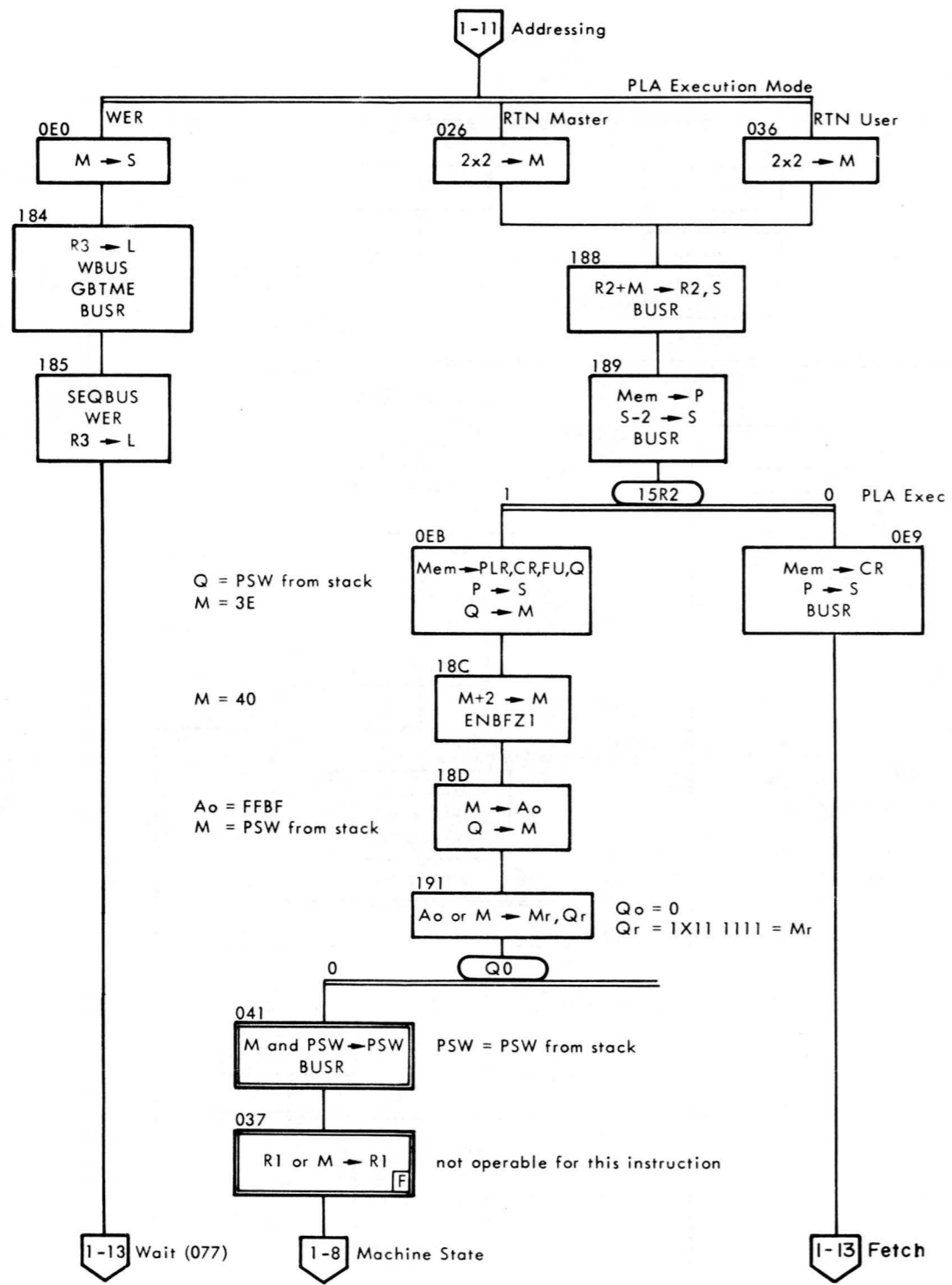
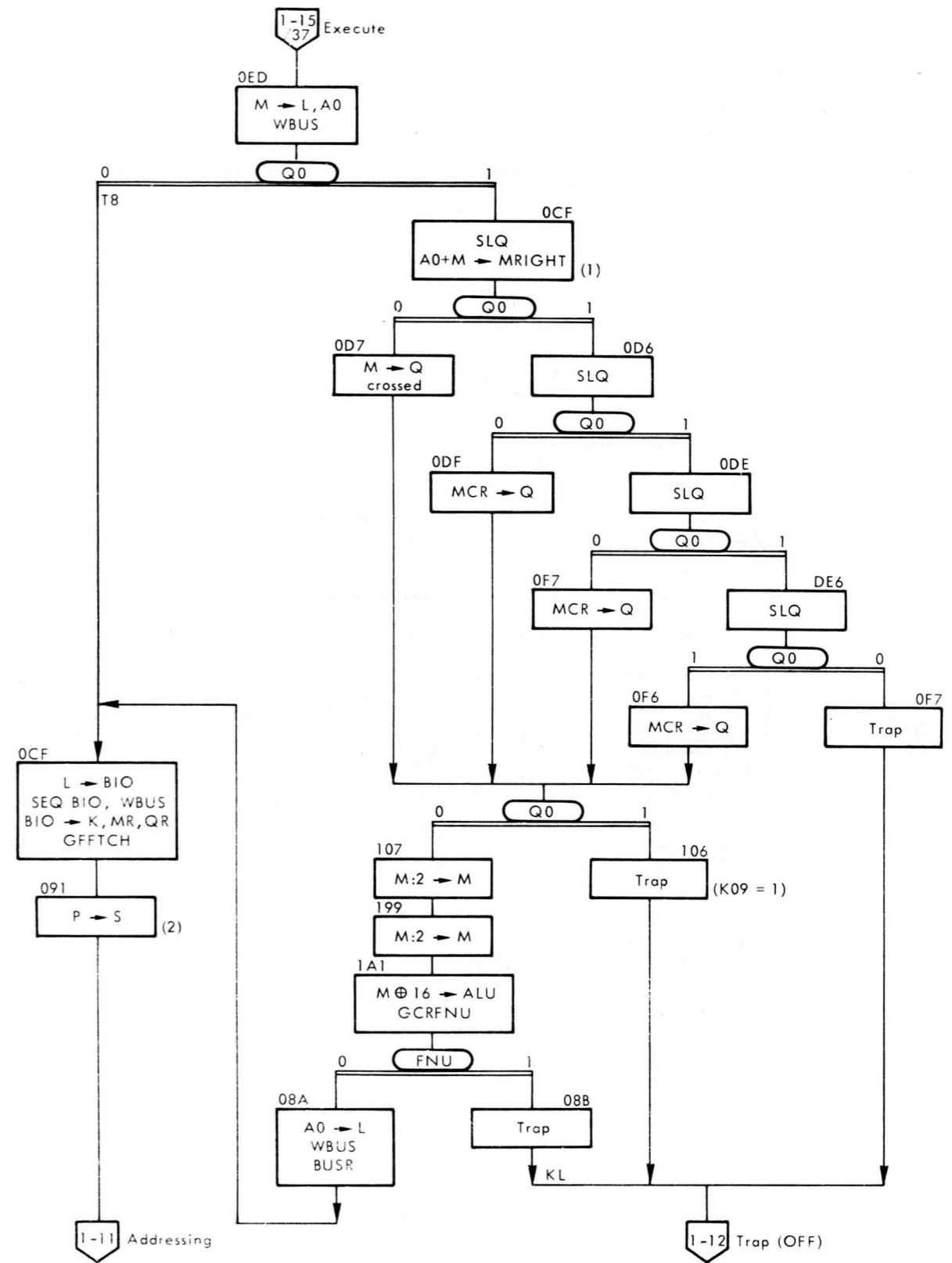


Figure 1-33 OPC 14 WER and RTN



(1) Contents of M are

0	k9	k15	0
---	----	-----	---

(2) This position permits the RSYZO which had been inhibited in 0CF by SERBIO so that the BIO's are not destroyed on T6. S is loaded because addressing routines type "S" (without P S) were used.

Figure 1-34 OPC 14 Execute (EX)

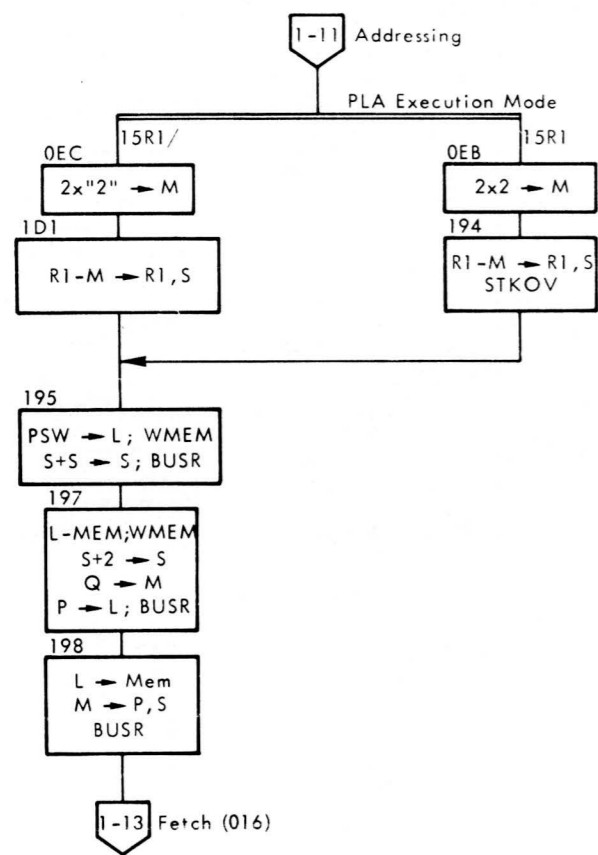


Figure 1-35 OPC 14 Call Function (CF)

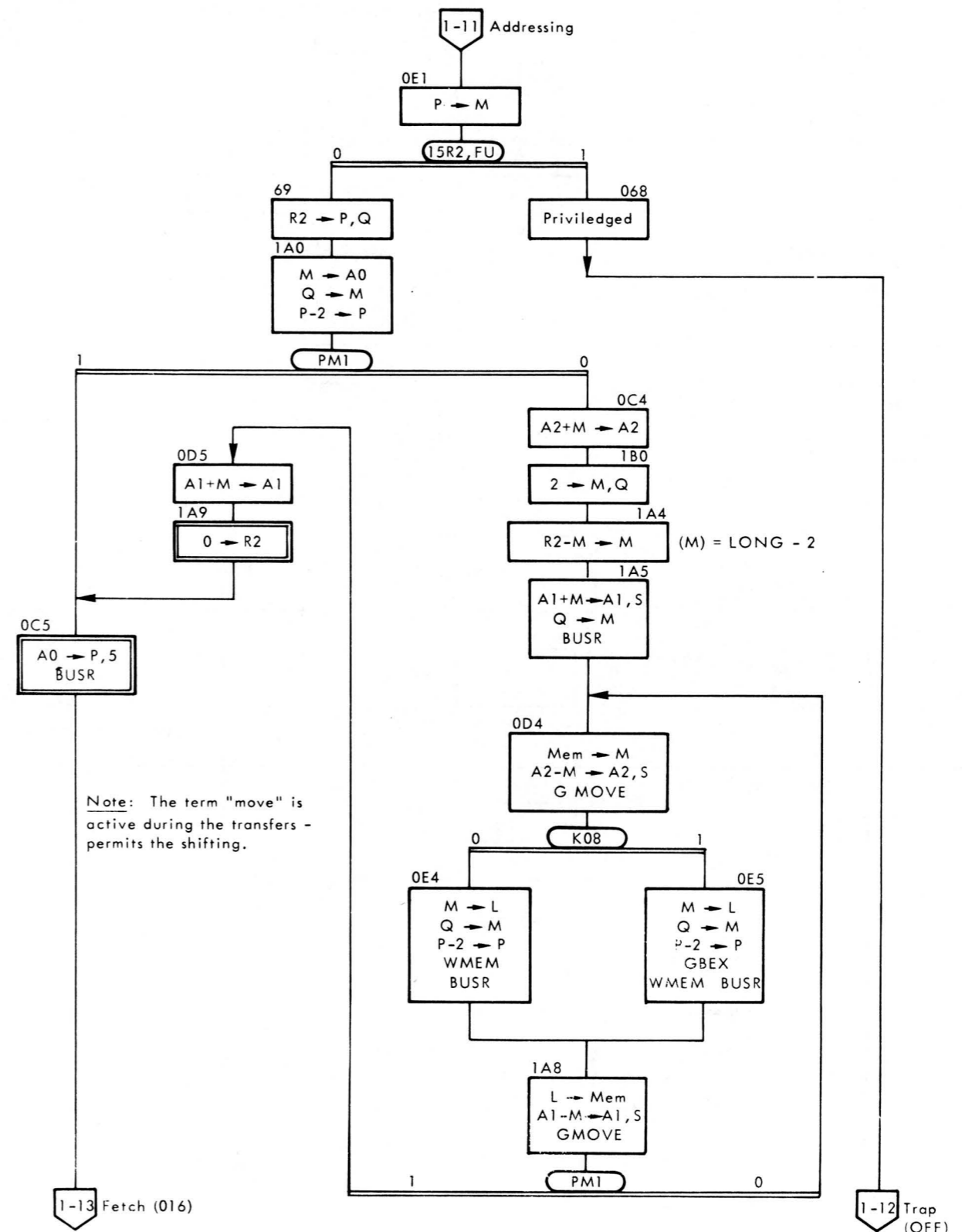


Figure 1-36 OPC 14 MVF and MVSU (P857 only)

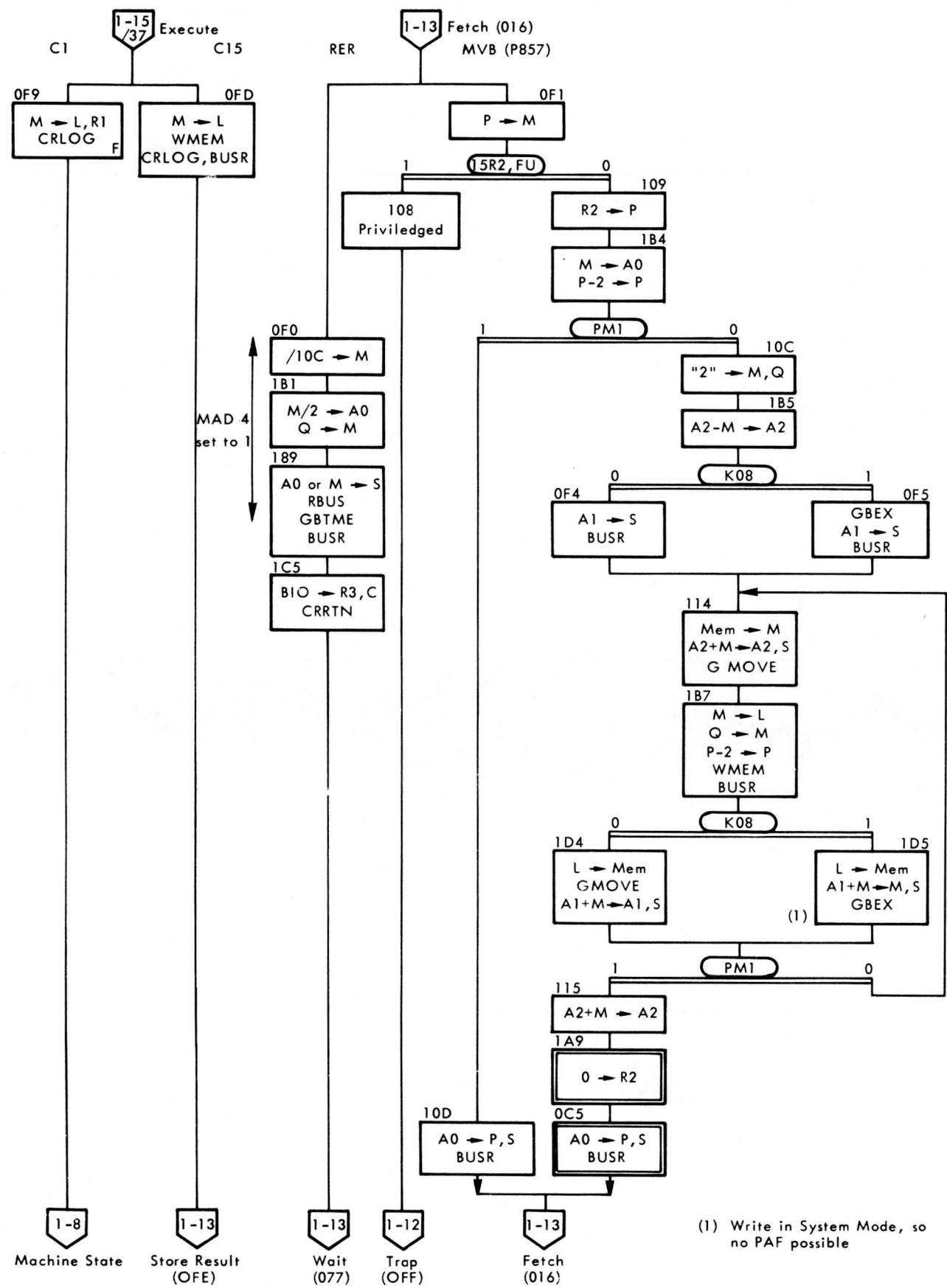


Figure 1-37 OPC 15 (CI, RER, MVB, MVUS)

SECTION II

CPU LOGIC

2.1 GENERAL

The CPU logic is divided into functional sections shown in the block diagram, Figure 2-1. Detailed logic diagrams and a list of all CPU signals are located at the end of this section. A guide to the integrated circuits is provided in Section IV. The CPU logic description is given in the following paragraphs:

2.4 GP Bus Lines	2.53 Data Handling Logic
2.10 Bus Controller	2.97 Interrupt Logic
2.24 Microprogram Control	2.104 Sequensor
2.48 Instruction Word Logic	2.116 Power-Fail/Restart/Resets

2.2 Signal Mnemonics

The signal names used are mnemonics for the function of the signals. The following letters have a special significance when used with the mnemonics:

- F indicates a flip-flop output.
- N suffix indicates an active-low signal (0V=1, 5V=0).
- Y indicates the copying of information.
- Z0, Z1 is used after a flip-flop mnemonic to indicate (respectively) setting to 0 or setting to 1 of the flip-flop.

A complete list of signals is provided at the end of this section (Table 2-14).

2.3 Logic Conventions

In the following logic descriptions, the suffix N indicates only the electrical level of the named signal (NAMEN is 0v when active and NAME is +5v when active). The logic state of the signal is indicated either by saying "active NAME," "inactive NAME," "reset NAME," etc., or by the bar (NAME, $\overline{\text{NAME}}$).

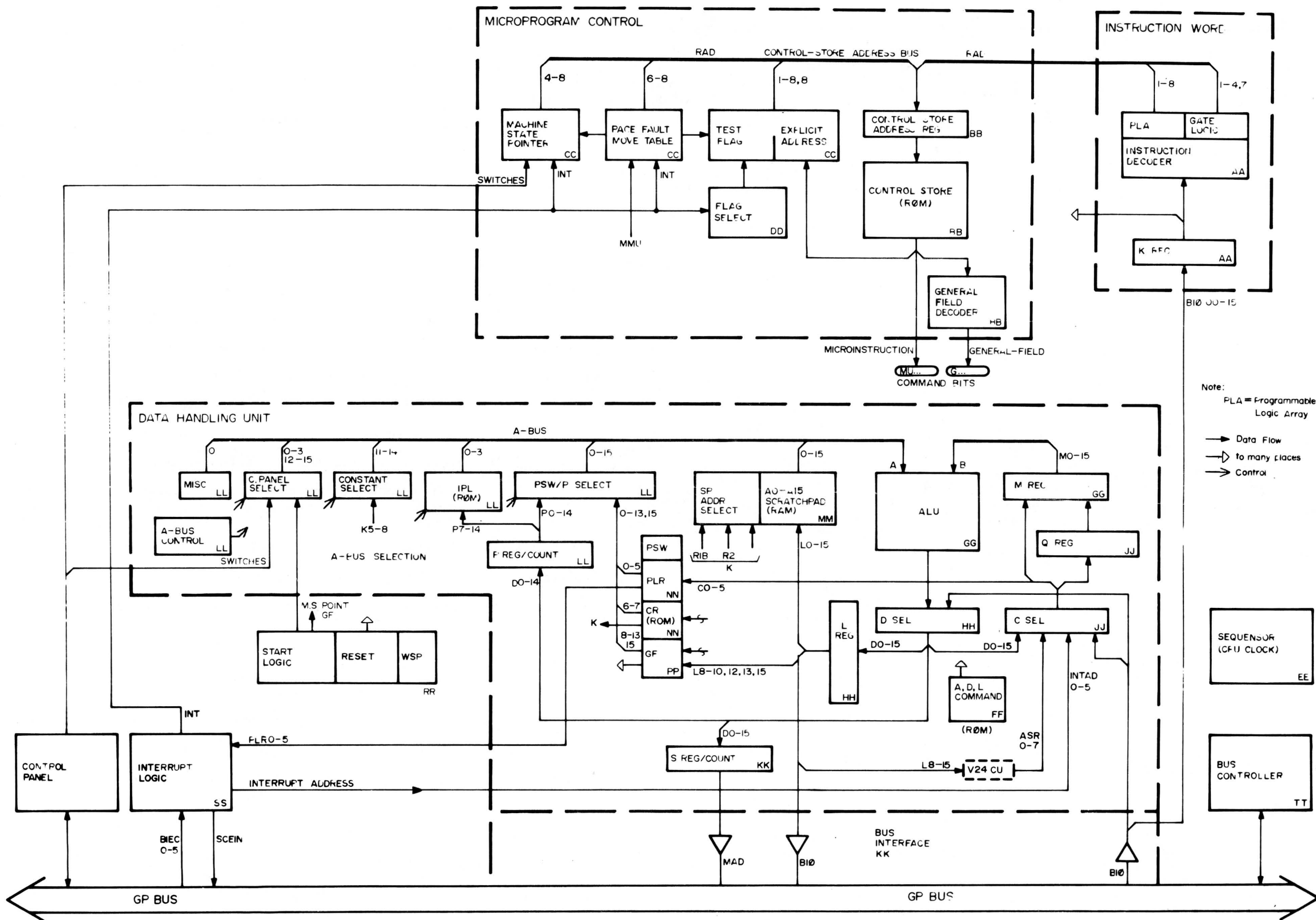
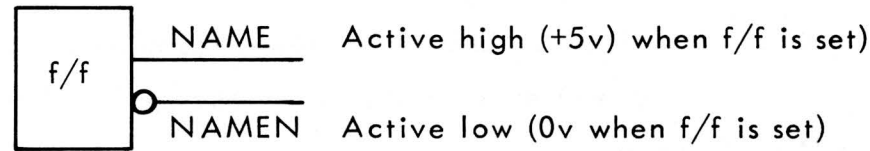


Figure 2-1 P856/857 CPU Block Diagram

Where command bits are used to form a control code, their state may be shown by the equation sign: $\mu\text{BIT}_{0,1,2} = 001$. A logic signal is in its True logic state when at the level specified by its name: NAME is True when high (+5v) and NAMEN is True when low (0v). Some examples are:



Signals in True logic state	Signals in False logic state
NAME, NAMEN	$\overline{\text{NAME}}, \overline{\text{NAMEN}}$
"active NAME," "active NAMEN"	"inactive NAME," "inactive NAMEN"
$\mu\text{BIT}_2 = 1, \mu\text{BIT}_{2N} = 1$	$\mu\text{BIT}_2 = 0, \mu\text{BIT}_{2N} = 0$

2.4 GP BUS

The 57-line General Purpose Bus comprises the following signals:

Control	BUSRN SPYC OKO/OKI MSN BSYN
Timing	TMRN TMPN TMEN TRMN TPMN
Data	BIO00-15N
Address	MAD00-15,64,128
Misc	ACN BIEC0-5 CHA CLEARN PWFN RSLN SCEIN WRITE

2.5 Control Lines

- BUSRN - Bus Request, from master to CPU (bus controller); remains active (low) while master is requesting control of the Bus.
- SPYC - Scan Priority Chain, CPU (bus controller) response to BUSRN; it warns the masters to prepare for the selection of a new master of the bus. SPYC is active low.
- OKO - from CPU (bus controller) to the master with the highest bus priority. If this master does not require the bus, it passes OKO on to the next lower-priority master. If a master requires the bus, it blocks OKO and generates MSN. (The order of priority of the masters is determined by hard wiring at installation time.)
- OKI - is the signal name of OKO at the input of each master.
- MSN - is the Master Selected, generated by the master which accepted OKO to take control of the Bus.
- BSYN - Bus Busy, from the CPU or any other master which has control of the Bus, when an exchange is in progress.

2.6 Timing Signals

- TMRN - from master to register or memory; validates the BIO and MAD lines and controls the exchange timing.
- TMPN - from master to peripheral CU; initializes the CU exchange and validates the CU address on the Bus.
- TMEN - from master to external register; validates the register addresses and the data, and controls the exchange timing.
- TRMN - from register or memory to master, in response to TMEN or TMRN when the slave is ready for the transfer; also terminates the exchange.
- TPMN - from peripheral CU to master, in response to TMPN to validate the response; also terminates the exchange.

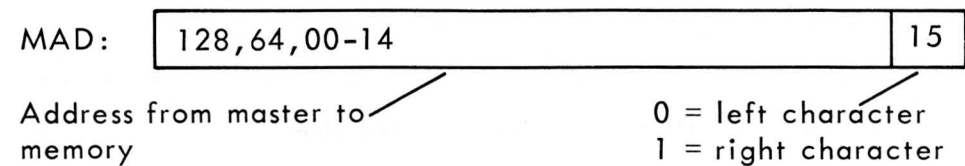
2.7 Data Lines

- BIO00-15 handle both input and output data between all system elements on the GP Bus.

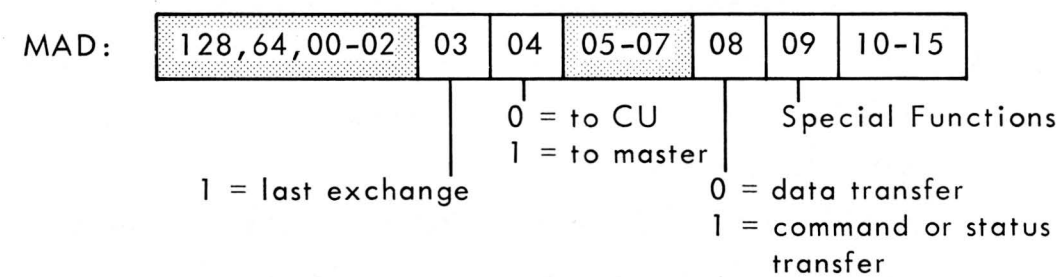
2.8 Address Lines

- MAD128,64,00-15 are the Bus address lines; 128 is the most-significant bit and 15 is the least-significant bit. The meaning of the MAD lines depends on the type of exchange, as follows:

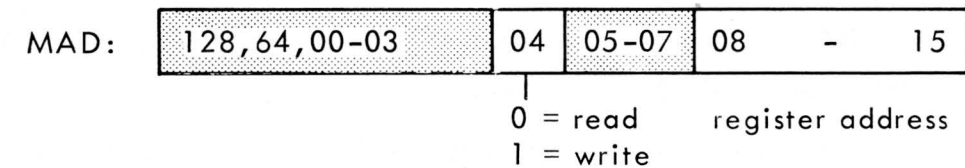
Master ↔ Memory (TMRN time)



Master ↔ CU (TMPN time)



Master ↔ External Register (TMEN time)



Note: Shaded parts not sent on Bus.

2.9 Miscellaneous Lines

- ACN - Accept, from the addressed CU. The CU accepts the command from the master.
- SCEIN - Scan External Interrupts from the CPU is sent at the end of every instruction; at SCEIN time, the CPU compares the BIEC0-5 code with the level of the running program to determine if a program interrupt is required.
- BIEC0-5 - Bus Interrupt Encoded is the priority level of the highest-priority interrupt request pending from an external rack, at SCEIN time.
- CHA - Character, from master to memory indicates exchange is by

character (CHA=1) or by word (CHA=0).

- CLEARN - General clear (reset) signal from CPU to all system elements, for initialisation.
- PWFN - Power Failure from the CPU sequences power off at power-failure and switching-off time without losing data, and controls power-on/auto-restart.
- RSLN - Reset Line from the CPU is the power restoration/validation signal.
- WRITE - From master to memory indicates exchange is write (line active) or read (line inactive).

2.10 BUS CONTROLLER

The Bus Controller logic (Figure 2-8TT) regulates access of all system masters to the GP Bus. When the Bus is free, the Controller scans the masters for a Bus-access request.

2.11 Bus Access

Any master requiring Bus access (Figure 2-2) may generate the Bus Request signal BUSRN if MSN is inactive, indicating no other master is being selected. The CPU Bus Controller logic receives BUSRN and, in response, generates Scan Priority Chain (SPYC), which is active low. If the Power Failure signal PWFN is inactive, BUSR sets the OKVAL flip-flop, and OKO is put onto the GP Bus along with SPYC. If PWFN is active, the CPU takes control of the Bus and does not generate OKO.

2.12 The highest-priority master (first master in the series) with a Bus Request blocks OKI and generates Master Selected (MSN). This highest-priority master which has just been selected may not have been the first to generate BUSRN. Active MSN blocks further generation of BUSRN by any master.

2.13 The Bus may be busy with an exchange (BSYN) while a new master is being selected. Once BSYN drops, the newly-selected master takes control of the Bus by generating BSYN and dropping MSN. With MSN inactive, any other master requiring Bus access may generate BUSRN and initiate a new selection sequence.

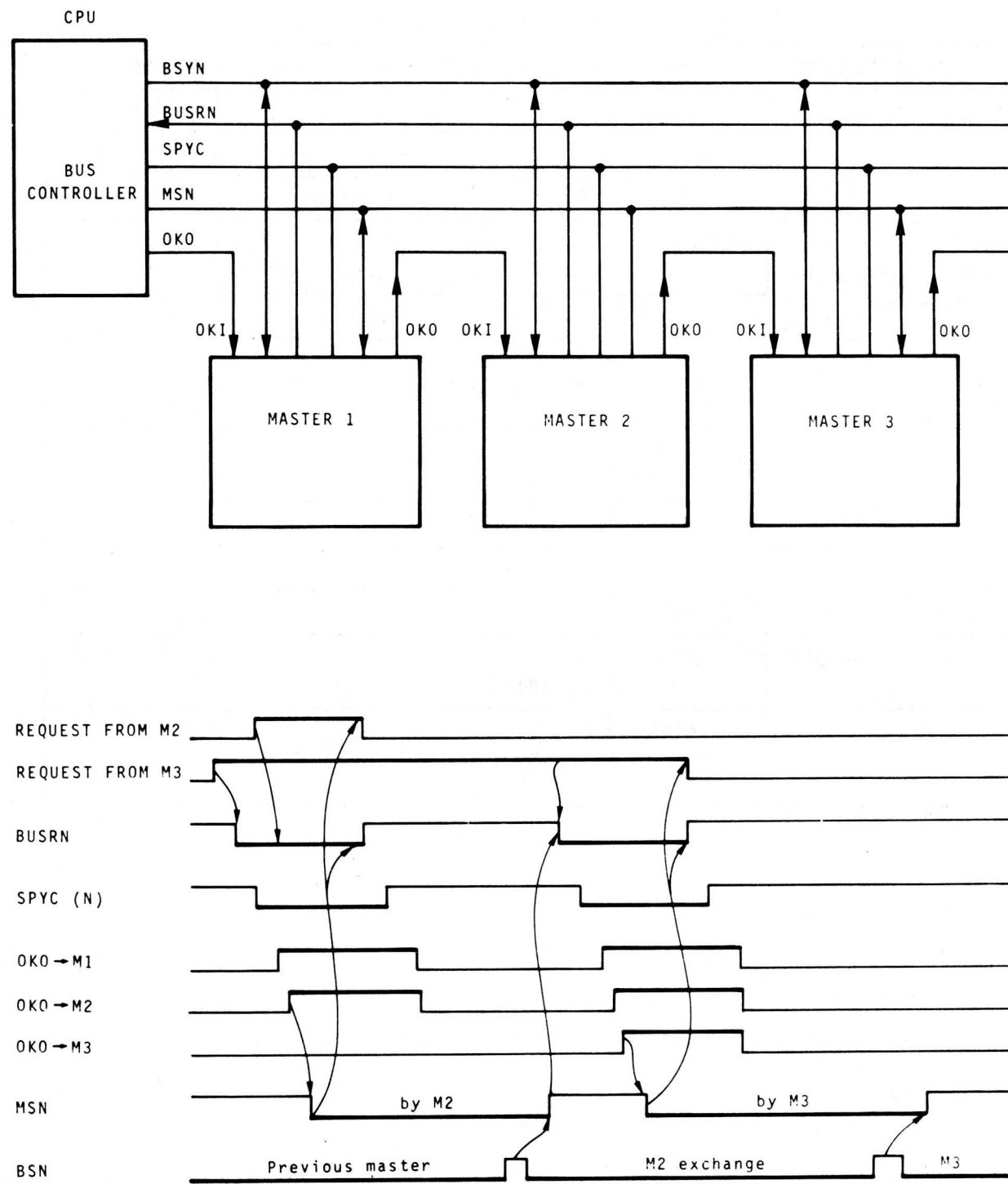


Figure 2-2 Bus Control Block/Timing Diagram

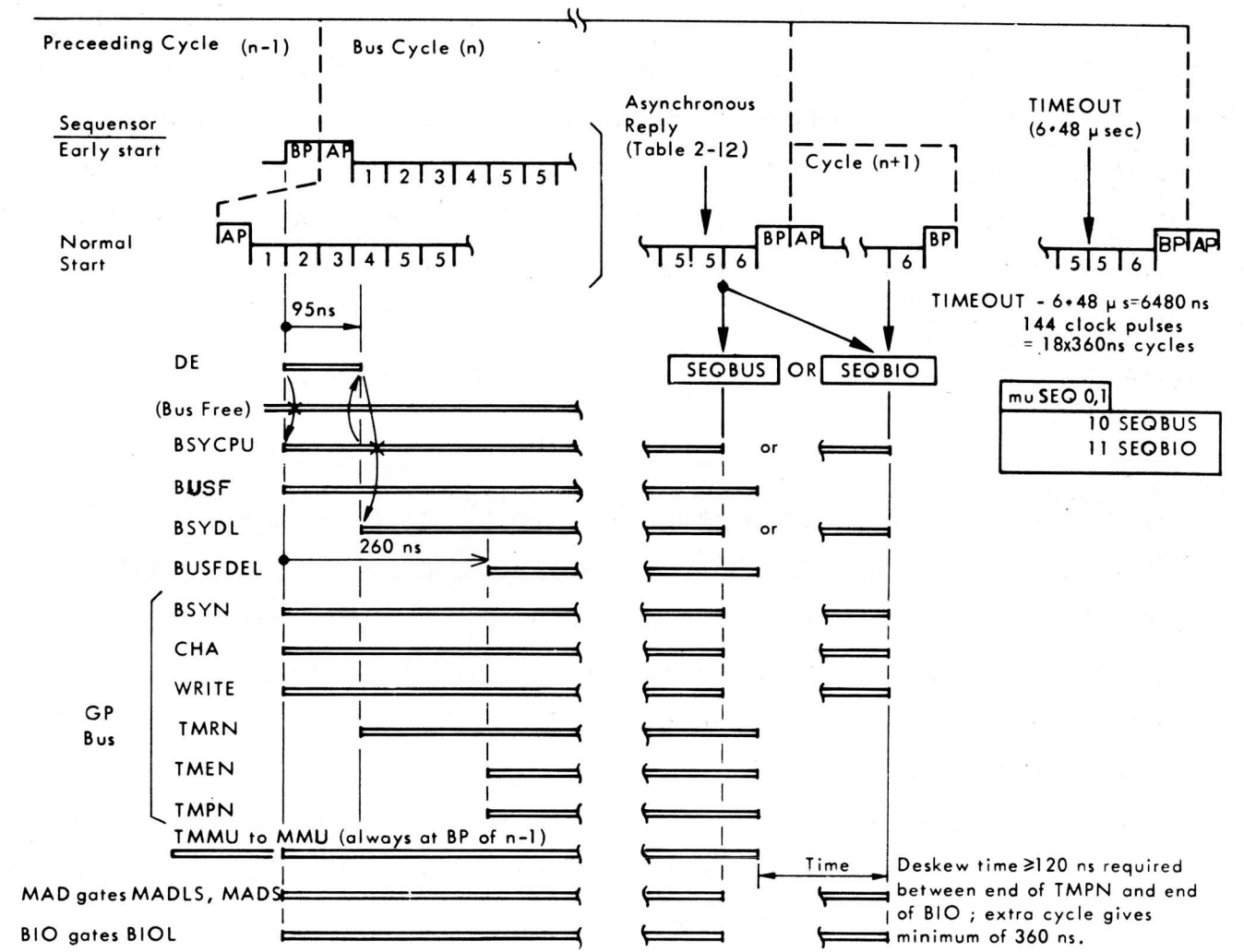


Figure 2-3 CPU Bus Control Timing

2.14 CPU Bus Control

The CPU has lowest priority for Bus access (except for power failure). The Bus-free condition, no other master requesting or using the Bus, is $\overline{\text{BUSRN}}$. $\overline{\text{MSN}}$. $\overline{\text{BSYN}}$. $\overline{\text{TRMN}}$. $\overline{\text{TPMN}}$. Any of these signals will inhibit BSYZ1N and prevent setting BSYCPU . The CPU takes control of the Bus under microprogram control when $\mu\text{SEQ0},1 = 1x$ (Sequensor cycles SEQBUS [10] or SEQBIO [11]).

2.15 (Figures 2-3, 2-8TT). If a specific microcycle (n-1) is always followed by a Bus microcycle (n), the Bus access is started early, at BP of the preceding microcycle (n-1). This reduces the Bus cycle by 135ns if the preceding cycle is a Sequensor logic cycle; additional time is saved following other sequensor cycles. The early Bus access is initiated by microinstruction-bit μBSR and BP setting the DE flip-flop (if there is no MMU page fault, $\text{MFAULTN}/\text{PAFN}$). When not started early, the Bus access is initiated at T2 of the Sequensor Bus cycle: T2 and μSEQ0 set the DE flip-flop if the early-start signal MUBURSRFN is inactive.

2.16 The DE output sets BSYCPU with signal BSYZ1N , if the Bus is free. BSYCPU in turn sets the BUSF flip-flop. If BSYCPU is set, DE resets itself after a 95ns delay. When DE drops, the BSYDL signal is activated; BSYDL and $\mu\text{SEQ0} = 1$ select the Sequensor operating cycles for Bus transfers (Table 2-12). BSYCPU also provides the following Bus gating:

- BYSN activated onto the Bus.
- Exchange-parameters CHA and WRITE gated onto the Bus: the CHA and WRITE data are always loaded into the Bus Controller from the microprogram control store at BP of the preceding cycle (n-1).
- Bus timing signal TMRN , TMEN , or TMPN gated onto the Bus: the conditions for these signals (Table 2-1) are set into the Bus Controller at BP of the preceding cycle.
- MADS , MADLS gated to the Bus interface logic for gating the MAD lines onto the Bus.
- BIOL gated to the Bus interface logic for gating the BIO lines onto the Bus.

Table 2-1 Bus Timing Signals

Operation	Microinst	Stored at BP	GP Bus Signal	Reply
CPU Exchange Ext. Reg/CU/Memory				
External Register	GBTMEN	TMEF	TMEN	TRMN
CU	GBTMPN	TMPF	TMPN	TPMN
Memory	μTMRN	TMRF	TMRN	TRMN
	$\overline{\text{GBCP}} + (\overline{\text{FU}} \cdot \overline{\text{GBEX}}) + \text{MMUABS}$			
Mem. via MMU translation	μTMRN	TMMU	TMRN from MMU	TRMN or MFAULTN
	$\overline{\text{GBCP}} \cdot (\overline{\text{FU}} + \overline{\text{GBEX}}) \cdot \overline{\text{MMUABS}}$			
MMU Operations				
Table Load	GBTMFN	TMMN		DONEMN
Table Store	GBOMN GCRFNUN	BOMFN		DONEMN
Load P'	GFETCH			
FPP Operations				
Load K'	GFETCH			
Store Operand	GBOFN GCRFNUN	BOFFN		
Load Operand	GBTMMN	TMFN		DONEFN
Process	GFLOT	FLOACT		DONEFN

P857 only

2.17 Bus Addressing (MAD Lines)

The control of the Bus MAD lines is determined by the MADLCPUN and MADCPUN signals which control MADLS and MADS to the MAD output gates. The gating signal to both MADLS and MADS is BSYCPU (preceding paragraph). The conditioning logic for setting MADLCPUN and MADCPUN produces the Boolean equations:

$$\text{for MADLS: } (\overline{\text{GBCP}} + \text{CPBABS}) \cdot \underbrace{(\overline{\text{GBCP}} + (\overline{\text{GBEX}} \cdot \overline{\text{FU}}) + \text{MMUABS})}_{\text{P857 only}} = \text{within page}$$

$$\text{for MADS: } (\overline{\text{GBCP}} + \text{CPBABS}) = \text{page address}$$

- If GBCP : the control panel validates the MAD lines; the CPU has

control only if control-panel B-half is absent (CPBABS).

- If \overline{GBCP} (P856): the CPU validates MAD128,64,00-15 with MADLS and MADS.
- If \overline{GBCP} (P857): the CPU validates MAD128,64,00-03 with MADS; the MMU validates the MAD04-15 lines if GBEX or FU, and the MMU option is present; the CPU has control only if the MMU is absent or if GBEX and FU.

The following diagram is a resume of the MAD line validation control for memory addressing:

MAD	128,64,00-03	04	15
$\overline{BSYCPU}.\overline{GBCP}.\overline{GBEX}.\overline{FU}$	CPU		
$\overline{BSYCPU}.\overline{GBCP}.\overline{CPBABS}$	CONTROL PANEL		
$\overline{BSYCPU}.\overline{GBCP}.\overline{CPBABS}$	CPU		
$\overline{BSYCPU}.\overline{GBCP}.\overline{(GBEX+FU)}.\overline{MMUABS}$	CPU	MMU] P857 only
$\overline{BSYCPU}.\overline{GBCP}.\overline{(GBEX+FU)}.\overline{MMUABS}$	CPU		

2.18 For memory addressing under CPU control, the S register provides the memory address to the Bus MAD lines, according to the preceding diagram. For CU or External Register addressing, the CPU logic provides the address via the D-selector and the S-register to the MAD lines.

2.19 Bus Data (BIO Lines)

The Bus data from the CPU are gated onto the BIO lines by the BIOL signal from the Bus Controller. BIOL is generated by microinstruction bit μBIOL , which is loaded at BP of the preceding microcycle (n-1) as BIOELN.

2.20 Bus Timing Signals

The Bus timing signals (Table 2-1) TMRN, TMEN, TMPN are generated by the master that has control of the Bus. For CPU Bus control, one of these signals is generated by the Bus Controller during the Sequensor Bus cycle (SEQBUS or SEQBIO). For MMU or FPP operation (P857), the Bus Controller sends a special timing signal directly to the MMU or FPP, and this external unit generates the Bus timing signal. The CPU waits at Sequensor clock-pulse T5 until the necessary

reply is received to end the Bus cycle. The reply (Table 2-5) is either the Bus response TRMN, TPMN to the CPU or the special reply signal directly from the MMU or FPP to the CPU.

2.21 When the required ending condition is satisfied, including reply received, the Sequensor steps to clock T6. For SEQBUS ($\mu\text{SEQ}0, \overline{1}$) operations, BSYCPU is reset by the T6 and $\mu\text{SEQ}1 = 0$. The inactive BSYCPU ends the Bus control and data lines BSYN, CHA, WRITE, MAD, and BIO. The timing signals end 45ns later, being reset directly by the BP clock. For SEQBIO ($\mu\text{SEQ}0, 1$) operations, the BIO lines must be prolonged at least 120ns to satisfy CIO timing requirements. In this case, $\mu\text{SEQ}1 = 1$ and BSYCPU is not reset. The Bus control and data lines, including BIO, are continued into the next Sequensor cycle (n+1) while the timing signal TMPN is reset normally by BP at the end of the BUS cycle. The microcycle after the Bus cycle (n+1) will have $\mu\text{SEQ}1 = 0$ so that BSYCPU is reset at T6 of that cycle.

2.22 Timeout

A timeout circuit (Figure 2-8TT) is used to release the Bus and set the condition register to 3 ($\text{CR} = 11_2$) if the addressed slave doesn't respond within 6.48usec. The basic clock for timeout is derived from the 45ns OSC signal (Figure 2-8EE) divided-by-2 by the OSC90 (90ns), and then divided-by-9 by the TC810 counter. The counter initially (following RSLBN -- 0) counts from 0 to 15; each TC810 output pulse (at count 15) then resets the counter to 7 so that the counting cycles continuously from 7 through 15. The resultant TC810 pulse (810ns period) drives the Timeout counter (Figure 2-8TT).

2.23 Timeout is held reset by the four inactive Bus-Control signals to the MR input. At the beginning of a bus cycle, (Figure 2-3), BSYCPUN or any master command goes active and the counter begins counting the TC810 input pulses. During the bus cycle, one of the TM... timing signals will also be activated. When the addressed slave responds, the bus cycle is terminated and the timing signal TM... and signal BSYCPUN are deactivated, and the Timeout counter is again reset. If there is no response by the time eight TC810 pulses have been counted

(6480ns), the counter produces TIMEOUT. TIMEOUT activates the required response signal on the Bus (TPMN or TRMN) to stop the exchange and unblock the system, generates sequensor pulse T6 to end the bus cycle, and sets CR bit 0.

2.24 MICROPROGRAM CONTROL

Each CPU instruction or operating sequence (paragraph 1.67) is controlled by a selection of microinstruction control words stored in a read only memory (Control ROM). The different microinstruction control words are accessed by the instructions or sequences (at each AP clock pulse time). The words are addressed partly by the instruction content (K register) and partly by the currently-accessed control word. Various CPU or system conditions may also modify or change the selection of the next microinstruction control word.

2.25 The microinstruction control words are used directly by the CPU logic as command bits ($\mu\dots$). Each 48-bit microinstruction is divided into 14 command fields (Table 2-2). The five-bit general-field section of the control word is decoded to produce 28 general field command bits for further microinstruction control. Both the micro-command bits ($\mu\dots$) and the general-field command bits (G...), listed in Table 2-3, are used by the CPU logic as direct command bits. These command bits are shown on the logic diagram (Figure 2-8BB) with their destinations listed; they are shown throughout the logic as command inputs, and identified by being enclosed in boxes.

Table 2-2 Microinstruction Command Bits

	Field	Bit	CPU section controlled by the command field
Next Word Control	μ SNA 0-1	0-1	Select Source of Next Address
	μ NA 0-8N	2-10	Next Address, Explicit
Data Path Control	μ A 0-4	11-15	A-Bus Selector
	μ ADL 0-4	16-20	ALU, D, L Command Decoder
	μ C 0-1	21-22	C Multiplexer
	μ MLOAD, MSEL	23-24	M Register
	μ Q 0-1	25-26	Q Register
	μ S 0-1	27-28	S Register
	μ P 0-1	29-30	P Register
	μ CT	31	CT Counter: loops, shifts
	μ SEQ 0-1	35-36	Sequensor (CPU clock). Bus Controller
	μ TMRN, BIOL, WRITE, BUSR	32-34 37	Bus Controller
	μ CR 0-2	43-45	Condition Register
	(not used)	46-47	
	μ GP 0-4	38-42	GENERAL FIELD SELECTION

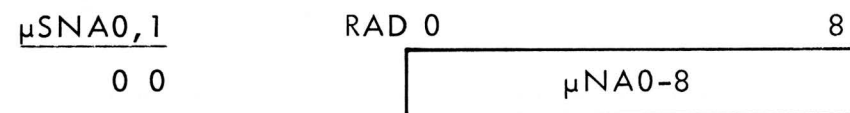
2.26 Microinstruction Addressing

The Next-Address mode for the microinstruction word determines which of four types of microprogram addressing is to be used. The mode is selected by bits μSNA0 and μSNA1 from the current microinstruction word, as follows:

μSNA0 0 1	Mode	Function
0 0	Explicit	Explicit address is $\mu\text{NA0-8}$.
0 1	Flag	Flag from execution-pointer tests (selected by $\mu\text{NA6-8}$) sets RAD8 ; $\mu\text{NA0-7}$ to RAD0-7 .
1 0	Instruction Word (PLA)	Instruction word (K-reg, decoder) provides address RAD1-8 ; μNA7 modifies the decoder for fetch or execution.
1 1	Machine-State	The machine-state pointer provides address RAD4-8 ; $\mu\text{NA0-3}$ to RAD0-3 .
x x	Move Table/Fault:	an interrupt or a page fault during a Move Table instruction force a special microinstruction address with bits RAD6-8 .

The Next-Address mode selects a set of open-collector gates onto the control-store address bus (Figures 2-8AA, CC). This control-store address is loaded into the RA register by the AP clock pulse.

2.27 Explicit Address. The next address is explicit in the current microinstruction $\mu\text{NA0-8}$ field.



The explicit-address gates (Figure 2-8CC) invert the $\mu\text{NA0N-8N}$ onto the RAD0-8 lines. All eight gates (and the flag gate) are blocked (forced high) by NARDT during a Move Table/fault. Other gates are blocked selectively by next-address modes which use some of the μNA bits.

2.28 Flag. The next address is explicit in the current microinstruction $\mu\text{NA0-7}$ field, with the least significant bit being selected by the Flag test. Any one of eight different execution-pointer tests can be performed during Flag mode, as selected by the bits $\mu\text{NA6-8}$.



$\mu\text{NA6,7,8N}$	Flag	Use
0 0 0	Q00	Display, C.Panel, IPL, Trap, End of TL-TS (P857), Execute, DIV correction.
0 0 1	IR	Move Jump.
0 1 0	PM1	End of ML, MS, Move
0 1 1	K08	Shifts, Move
1 0 0	FNU	C.Panel test, IPL, Wait-C2, Trap in EX-T2.
1 0 1	NORM	End of normalized shifts.
1 1 0	DIV	Remainder correction.
1 1 1	FU15R2	Privileged test; Trap for SLN, SRN; Move.

The Flag mode uses explicit-address gates 0 to 7 (Figure 2-8CC) onto the RAD0-7 lines. The eighth explicit gate is blocked by NAEXPL while the flag gate is enabled by NAFLG . The signal FLAGN sets bit RAD8 . FLAGN is selected by a type 74151A eight-input multiplexer (Figure 2-8DD) which is controlled by $\mu\text{NA6N,7N,8N}$. The eight tests selected for the FLAGN bit are described in the following paragraphs.

2.29 Flag Q00N is used as a general test condition for various commands and instruction.

2.30 Flag IR is the interrupt request. During a page fault or an interruption of a Move-instruction/execution, flag IR enables the separation of the exit subroutine. If the Move instruction was interrupted, the last exchange was executed. These subroutines save the parameters necessary for resuming the Move execution.

2.31 Flag PM1 is used to detect the last word transfer during a Multiple Load or Store (ML, MS) or Move instruction. PM1 is taken from the un-used (most-significant) bit position of the P counter when it is used as an auxiliary

counter. The P counter is normally used as a 14-bit counter for program addresses. When used as an auxiliary counter, PM1 is set by the first decrementing clock after P0-14 has counted down to zero.

2.32 Flag K08 is taken directly from the instruction word to differentiate between different types of shift and Move instructions, as follows:

Shifts: K08-0 = SLA, SLL, SRA, DLA, DRA
 K08-1 = SLN, SLC, SRN, DLN, DRN

Move: K08-0 = MVF, MVB
 K08-1 = MVSU, MVUS

2.33 Flag FNU stores the zero contents of the ALU. FNU is stored in one bit of a four-bit register (74175). The other three bits of this register have no relationship with FNU.

2.34 Flag NORM defines the end of a shift normalize execution. The end-of-shift signal is GOSH (Figure 2-8DD) which is controlled by a 74151A chip. The inputs to the multiplexer chip are held at 1 or 0 (+5v or 0v) so that the selection and enable inputs provide direct control of GOSH as follows:

$$Q15+(L00 \oplus L01)+P09 = \overline{GOSH} = NORM$$

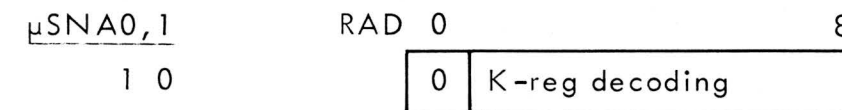
- Shift Right Normalize ends when Q15 sets. The active Q15N selects the low 74151A inputs 10-14 to generate a low GOSH. The P counter is used as an auxiliary counter to count 32 shifts (the program address is saved). If P09 sets before Q15, it means the constant to shift is equal to zero and GOSH goes low. L0⊕L1 never sets.
- Shift Left Normalize ends when L00⊕L01 is true. Since Q15N is not used (held high), L00,01 select 74151A inputs 14-17 as an EXclusive-OR, and GOSH is low when L00⊕L01 is true. The P counter is used in the same manner as in Shift Right Normalize.
- Shift without normalizing. The number of shifts to be executed is contained in K11 - K15. This number is loaded in complement into the Scratchpad CT counter. The counter is incremented to 31 in a single microinstruction (CT+1, Repeat), and the shifting is ended when the microinstruction is finished.

2.35 Flag DIV enables the remainder correction according to test DIVA or DIVB, selected by μQ1. Test DIVA (μQ1) is true if the remainder sign is opposite to the dividend sign. Test DIVB (μQ1N) is true if both the remainder and dividend are <0, but the absolute remainder value is greater than the absolute divisor value. The select inputs and data inputs of a 74151A multiplexer chip are used together to generate the FLAGDIV signal when either test is true.

- DIVA: μQ1 FSIGDIV FSIG (input 14)
 FSIGDIV FNU (input 16,17)
- DIVB: μQ1 FSIGDIV FSIG FNUN (input 13)

2.36 Flag FU15R2 tests for System-Mode reserved instructions which can modify the stack pointer contents. When the stack pointer is indicated (R2E15) and User Mode is set (FU), the active FU15R2N signal is inverted to a high FLAGN, and the bit RAD8 = 0 to cause a trap.

2.37 Instruction Word. The next address is decoded directly from the instruction word.



This is used as the first address of an instruction microprogram. The instruction-word addressing mode operates in one of four sub-modes, selected by the signals PLA0, PLA1 :

PLA0,1	Sub-Mode	Purpose
0 0	Inhibit	Instruction-word outputs held inactive (high).
1 0	Add Master	Fetch operand for types T-T7; execute for type T8; Trap if illegal.
1 1	Add User	Fetch operand for types T1-T7; execute for type T8; Trap if privileged.
0 1	Execution	Used for first microinstruction word after the fetch-operand routine.

The PLA selection logic is shown on Figure 2-8AA. The sub-mode inhibit is set by the inactive validate signal (PLAVALN high). PLAVALN is held high

by $\mu\text{SNA}0 = 0$ (Explicit mode or Flag mode) or an active PAFN. PLAVALN is activated by $\mu\text{SNA}0 = 1$ (Instruction-Word or Machine-State modes). The conditioning for PLA is as follows:

	μSNA 0,1	PLAVALN	$\mu\text{NA}7$	PLA 0,1	Sub-Mode
	0 0, 0 1	h		0 0	Inhibit
Explicit Implicit	1 0 1 1]RADETPLN]	L	0 0 1 0 - 1	1 0 1 1 0 1	Add Master Add User Execution

Microprogram bit $\mu\text{NA}7$ selects a fetch sub-mode (Add Master or User) or the execution sub-mode. If fetch is selected, the FU signal indicates either User mode ($\text{FU} = 1$) or Master mode ($\text{FU} = 0$). For all three of the active sub-modes, the instruction-word addressing may be explicit ($\mu\text{SNA}1 = 0$) or implicit ($\mu\text{SNA}1 = 1$). The explicit addressing is used for instruction execution and the execution pointers. The implicit addressing is part of the machine-state-pointer control of the microprogram.

2.38 Instruction-word addressing of the microinstruction store is controlled by the instruction-word decoder (Figure 2-8AA). The decoding is done mainly by a programmable logic array (PLA) which provides address codes for 96 different input combinations. In many cases, the output address codes are the same for various different input combinations. The PLA decoding is shown in Table 2-6, Instruction Decoder. Some instruction-word decoding is done by logic gates operating in parallel with the PLA.

2.39 The external instruction-word logic gates are used for an ineffective branch (NOJUMP) or, with P857, a floating-point instruction without an attached FPP (NOFLO). For both instructions, PLA0 is high, and the external gates for RAD1-4 are enabled. The NOFLON signal generates NACND which activates the four gates to force RAD1 through 4 low, thus selecting address 001. The (P857) NOJUMP signal generates NACND and NACNDAD. These two signals activate the gates for RAD1 through 4, and RAD7, thus selecting address 003.

No Jump	RAD	0	1	2	3	4	5	6	7	8	
		0	0	0	0	0	0	0	0	1	/001
No FFP	RAD	0	1	2	3	4	5	6	7	8	
		0	0	0	0	0	0	0	1	1	/003

2.40 Machine-State Pointer. The Next Address is determined by machine-state pointers, such as RUNF and control-panel switches. (See also Figure 1-9 for general flow). The machine-state pointer logic is shown on Figure 2-8CC. The instruction-word decoder (Figure 2-8AA) is also used for one sub-mode of the machine-state pointer microprogram control. There are four sub-modes of machine-state pointer address selection, as follows:

<u>Instruction</u>	$\text{RUNF} \cdot \overline{\text{IR}} \cdot \overline{\text{PUP}} \cdot \text{KRY}$	RAD	0	1	2	3	4	5	6	7	8	
			0									PLA Decoder
<u>Interrupt</u>	$\overline{\text{RUNF}} \cdot \text{IR}$	RAD	0	1	2	3	4	5	6	7	8	
($\text{RUNF} \mu\text{NA}8$ = RADET6)	KRY $\overline{\text{KRY}}$		0	0	0	0	1	0	1	0	1	(/015)
			0	0	0	0	1	0	1	0	0	(/014)
<u>C. Panel</u>	$\text{RUNF} \cdot \text{PUP}$	RAD	0	1	2	3	4	5	6	7	8	
			0	0	0	0	1	0	0	0	0	(/010)
<u>Program End</u>	$\text{RUNF} \cdot \overline{\text{IR}} \cdot \overline{\text{PUP}}$	RAD	0	1	2	3	4	5	6	7	8	
($\text{RUNF} \text{PUP}$ = RADET6)	KRY $\overline{\text{KRY}}$		0	0	0	0	0	0	1	0	1	(/005)
			0	0	0	0	0	0	1	0	0	(/004)

In all cases, the control-panel TEST key will set RAD bit 0 to 1, via ETATEST and VALNA0 on Figure 2-8CC. The Instruction sub-mode generates the signal RADETPLN (Figure 2-8CC) which activates the instruction-word decoder PLA via signal PLAVALN on sheet AA. Refer to paragraph 2.37. The Interrupt sub-mode selects address /014 or /015 for the interrupt routines. The Control-Panel sub-mode selects address /010 if any of the control-panel command signals are active (PUP signal active). The control-panel command switches operate through the A-bus of the Data Handling logic. The Program-End sub-mode selects a branch to address /004 or /005.

2.41 Move Table/Fault (P857). An interrupt or a page fault during a Move Table instruction force the RAD bits to a special control-store address, as follows:

	Address	RAD	0	1	2	3	4	5	6	7	8
Interrupt, MVB read	1F8		1	1	1	1	1	1	0	0	0
MVB write	1F9		1	1	1	1	1	1	0	0	1
MVF read	1FA		1	1	1	1	1	1	0	1	0
MVF write	1FB		1	1	1	1	1	1	0	1	1
Page Fault	1FF		1	1	1	1	1	1	1	1	1

The move table/fault logic is shown on Figure 2-8CC. Either the page fault (PAFN) or the interrupt during Move Table (MOVEIRN) cause the signal NARDT. NARDT blocks all the control-store addressing gates on the machine-state-pointer, explicit-address, and test-flag logic, which puts the address bits high. The move table/fault logic then forces bits RAD6-8 low according to its control inputs to obtain the address code shown in the above table.

2.42 RA -- Microinstruction Address Register

The 9-bit RA register holds the address of a single 48-bit microinstruction word. This is actually a 10-bit buffer register comprising one 6-bit 74s174 and one 4-bit 74s175. The tenth bit position, however, is used to store the command-bit-derived signal FETCH, which is re-generated as FTDEL.

2.43 The 9-bit address code RAD0-8 is loaded from the control-store address bus at the leading-edge of clock pulse AP. The address code is produced by the Instruction Word logic or the Microprogram Control logic; the source is selected by gating (from the source selector) within that logic. The reset signal RSLBN resets the RA register to all zeros so that control-store address 000 is selected by the RA output. The signal RSLBN is derived from the GP-Bus reset-line signal RSLN, via RSLF.

2.44 Microinstruction Store (Control ROM)

This section is composed of six type 8205 read only memory (ROM) ICs. Each

IC contains 512 eight-bit words which are accessed by nine addressing inputs. Addressing from the RA register is applied in parallel to the six ICs to obtain 512 48-bit words of store.

2.45 The enabling inputs of the control ROM are tied high; the outputs thus display continuously the content of whatever word is addressed by the RA register. When RA is reset by RSLN, the control-ROM address zero is selected (Idle state). A listing of all control-ROM microinstructions is provided in Table 2-4B. The binary contents of the control ROM are provided in Table 2-5.

2.46 Microinstruction Decoding

Two groups of command bits (Table 2-2) are used by the CPU logic as direct command bits. The micro-command bits ($\mu...$) are the actual bit contents of the currently-addressed micro-instruction word. The general-field section of these bits (μ GP0-4) are decoded to produce a further 28 command bits (G...).

2.47 The general-field command bits (Figure 2-8BB) are decoded by four type 74s138 ICs. The signals μ GP4,3,2 are connected in parallel to the A, B, C inputs. The gating inputs G1, G2A, G2B are connected as follows:

Bits:	1-7	9-15	16-23	24-29
G1	+5V	μ GP1	μ GP0	μ GP0
G2A/	μ GP0	μ GP0	0V	$\overline{\mu$ GP1
G2B/	μ GP1	BPN	μ GP1	0V

The three gate inputs must all be active (G1-high; G2A, B-low) to enable the three A, B, C inputs. The resultant output for the different input combinations are listed in Table 2-3. Seven of the general-field commands are valid during BP for use as set/reset commands. Ten other general-field commands are stored at BP time and updated on the following BP; these are used as parameters for next-cycle exchange.

Table 2-3 General Field Command Bit Codes

μ GP 01234	Command Bit	Clockea by BP	Stored on RP	Function	Application
00000	---				
00001	GIDLEN	--	--	Resynchro. Sequensor	Display, Tests.
00010	GCTLDN	--	--	Load Counter	Shift, DAR, DSR.
00011	GCRDSRN	--	--	Set OVF if Divide Error and DOOD	Shift right, Test Divide.
00100	GCRFNUN	--	--	Store OALU in FNU	IPL, PUP, Arith. Oper.
00101	GCRVMLN	--	--	Set CR if OVF; Branch for ML.	ML, Divide correction, Shift LA.
00110	GCSELN	--	--	ASR or INTAD on sel.C; data Q15	DLL, DLC, INR, Interrupt.
00111	GMOVEN	--	--	Branch during Move	Move read/write cycles.
01000	---				
01001	GFSYSN	Ys	--	Reset ENBF, ARF, PAF, FU.	System operations: INT, IPL, PAF, etc.
01010	GFENBN	Ys	--	Set ENBF	RTN with R2=15.
01011	GFSTOV	Ys	--	Allows set PIF if Stack OVF.	Update A15.
01100	GFPLR/ CLPLR	Ys	--	Loads PLR, FU by C	INT, AR, RTN Master Control-panel operations
01101	GFRZO	Ys	--	Reset RUNF	INT, PAF, AR, LR, RR
01110	GFKYZON	Ys	--	Reset K Ready	
01111	GCRVZON	Ys	--	Reset OVF; Divided Sign Store	Shift LA, DIV.
10000	GBCHN	--	Ys	Character Mode	Character instructions.
10001	GBCPN	--	Ys	C. Panel-B memory address	LM, RM.
10010	GBEXN	--	--	MMU translation to System Mode	EL, ES, MVUS, MVSU.
10011	GBOKN	--	Ys	(keys) on BIO	IPL, C. Panel, Tests.
10100	GBOFN	--	Ys	(FPP) on BIO	FPP store, FIX.
10101	GBOMN	--	Ys	(MMU) on BIO	TS, PAF.
10110	GBTMEN	--	Ys	Ext. Register Cmnd.	RER, WER.
10111	GBTMPN	--	Ys	Prog. Channel Cmnd.	I/O instructions.
11000	GBTMMN	--	Ys	MMU command	TL
11001	GBTMFN	--	Ys	FPP command	FL, FO, FOS.
11010	GFLOTN	--	Ys	FPP activation	Wait for FPP execution.
11011	GAEXLN	--	--	Execute double word	Trap for EX, MV, 15R2, OPC=15.
11100	GFETCHN	--	--	Load K; set CT to 16	Fetch, EX, LR, RR, Tests.
11101	GMULTIN	--	--	D selection; OVF set for Multi	Last cycle of Multi.
11110	---				
11111	---				

Ys = Yes

Table 2-4A Control-ROM Microinstruction Listing P857

ADD.		ADD.		ADD.	
000	ROM /1E8.ARI5.ALUB.CALU.MYC.QYC,,,WBUS,,,0	IDLE	040	FTA AWR1.AANDB.CRLOG	AN
001	FTA ,0	NO JUMP	041	ROM FLAG./1C8.APSW.AANDB.CALU.MYC,,,,,BUSR,,0	HLT.RIT.INH.RTA15
002	ROM /1E9,,,,,SP2.PP2,,,BUSR,,0	INCR P	042	ROM /DFD,,,CBIO.MYC,,,PP2,,,SEQBUS,,,0	RT5
003	ROM FLAG./000.AZ.ALUA.CALU,,QYC,,,,,GAEXL,0	NO FLOT PROC	043	ROM /DF5,,,CBIO.MYC,,,PP2,,,SEQBUS,,,0	RT5S
004	ROM FLAG./1F8.ASYS.TWOA.CALU,,QYC,,,,,N	TEST DISPLAY	044	ROM /101.ARR1.AANDB,,,,,WMEM,,BUSR,,CRLOG	AN S
005	ROM /1FB,,,,,SM2.PM2,,,,,GFKEYZ0,0	DECR BEFORE VISU	045	ROM /101.ZERO,,,,,WMEM,,BUSR,,0	CM
006	ROM /1FF.AEP.ALUA.CALU.MYC,,,,,0	VISUA	046	FTA ARR1.AANDB.CRLOG	TM
007	ROM /1FF,,,DBIO.CBIO.MYC,,,,,SEQBUS,,,0	VISUB	047	ROM /1E9,,,,,BUSR,,CRFLO	FPP WAIT EXEC.
008	FTA AWR1.ALUB.CRLOG	LD	048	ROM FLAG./1A6.ARAL.DIVALU,,,,,GCRFNU,0	DIV TEST COR.REM.
009	ROM /1E9,.ALUB,,,SYD.PYD,,,BUSR,,CRLOG	LD P	049	ROM FLAG./140.ARAL.AXB.CALU.MYQ.QYC,,,,,0	DIV TEST QUOT.
00A	ROM /16D.ASYS.ALUA,,,,,WBUS,,,0	RCP IN L REG. LR OR RR	04A	ROM /DE2.AEP.AMB.CALU.MYC,,,,,0	SLN
00B	ROM /1FF.ARAO.ALUA,,MYC,,,,,0	ERRONEOUS START	04B	ROM FLAG./1B5.AWR1.TWOA,,,,,PP2,,,0	SLN LOOP
00C	ROM /101.ARR1.ALUA,,,,,WMEM,,BUSR,,0	ST	04C	ROM FLAG./1B2.ARCT.ALUA,,SP2.PM2.CTPI,WMEM,SEQBUS,,,0	MS LOOP
00D	ROM /101.AEP.ALUA,,WMEM,,BUSR,,0	ST P	04D	ROM /1E9,.ALUB,,,SYD.PYD,,,BUSR,,0	MS END
00E	ROM /OE6.ATWO.ALUA.CALU.MYC,,,,,0	STD	04E	ROM FLAG./1A8,,,,,SLQ,,,GFYZ0,0	TEST READM
00F	ROM /OE5.ATWO.ALUA.CALU.MYC,,,,,0	STD P	04F	ROM /172,,,DBIO.CBIO.MYC,,,,,WMEM,SEQBUS,,,GBCP,0	LOAD M
010	ROM FLAG./100.ASYS.ALUA.CALU,,QYC,,,,,0	TEST AUTOREST	050	FTA AWR1.AOB.CRLOG	OR
011	ROM /1E9,.ALUB,,,SYD.PYD,,,BUSR,,0	AB	051	ROM FLAG./1B8.APSW.AOB.CALU.MYC,,,,,0	ENB SMD LKM
012	ROM FLAG./DE0.AWAD.ALUB,,,,,RBUS,,,GBOK,0	SAVE P TEST IF L/R R	052	ROM /DEC,,,DBIO,,,SYD.PP2,,,SEQBUS,BUSR,,0	RT4
013	ROM /DE0.ARAO.ALUA,,SYD.PYD,,,0	LOAD P AND S LR	053	ROM /DE8,,,DBIO,,,SYD.PP2,,,SEQBUS,BUSR,,0	RT4S
014	ROM /108.ATWO.TWOA.CALU.MYC,,,,,0	INT	054	ROM /101.ARR1.AOB,,,,,WMEM,,BUSR,,CRLOG	OR S
015	ROM /108.ATWO.TWOA.CALU.MYC,,,PM2,,,GFKEYZ0,0	INT DECR	055	ROM /OAB.AWAD.BSHR,,MYQ,,,,,0	INR
016	FTA ,0	FETCH	056	ROM FLAG./1A0,,,,,SLQ,,,,,0	TEST READST
017	ROM ETAT./1FE,,,,,SEQBUS,,GIDLE,0	VISU IDLE	057	ROM FLAG./190.ASYS.TWOA.CALU,,QYC,,,,,GBCP,0	READ M
018	ROM FLAG./193.AEP.ALUA.CALU.MYC,,,,,GCRVZO,0	SLASLN SAVE P	058	ROM /1A6.AWAL.DIVALU,,,,,0	DIV CORR.REM.
019	ROM FLAG./15B.AFP.ALUA.CALU.MYC.SLQ,,,,,ACRVZO,0	DLADLN SAVE P	059	ROM FLAG./140.ARAL.AXB.CALU.MYQ.QYC,,,,,0	DIV TEST QUOT.
01A	ROM FLAG./17B.ARR1.ALUA.CALU,,QYC,,,,,0	SLL SLC	05A	ROM /DE2.AEP.AMB.CALU.MYC,,,,,0	SRN
01B	ROM /OCE.AWAL.ASHL,,SLQ,,,CTPI,,REPEAT,,GCSEL,0	DLL DLC	05B	ROM FLAG./1A5.AWR1.ASHR,,SRQ,PP2,,,GCRDSR,0	CRN LOOP
01C	ROM FLAG./16B.AEP.ALUA.CALU.MYC,,,,,0	SRA SRN	05C	ROM FLAG./1A2.ARCT.ALUA,,SM2.PM2.CTPI,WMEM,SEQBUS,,,0	MSRD LOOP
01D	ROM FLAG./14B.AEP.ALUA.CALU.MYC.SLQ,,,,,0	DRA DRN	05D	ROM /DBE,.ALUB,,MYQ,,SYD.PYD,,,0	MSRD END
01E	ROM /ODD.ARR1.ALUA.CALU,,QYC,,,,,0	SRL SRC	05E	ROM /198,,,,,SLQ,,,,,0	PUP
01F	ROM /OCE.AWAL.ASHR,,SRQ,,,CTPI,,REPEAT,,GCRDSR,0	DRL DRC	05F	ROM /1FF.APSW.ALUA.CALU.MYC,,,,,0	READ STATUS
020	FTA AWR1.APB.CRADD	AD	060	FTA AWR1.AXB.CRLOG	XR
021	ROM /1E9.AEP.APB,,,SYD.PYD,,,BUSR,,CRADD	AD P	061	ROM /O47.APSW.ALUA.CALU.MYC,,,SP2,,,GFSYS,0	INT
022	ROM /DEC.ARR2.ALUA,,SYD,,,BUSR,,0	RT3	062	ROM /DFC,,,CBIO.MYC,,,,,SEQBUS,,,0	RT7
023	ROM /DE8.ARR2.ALUA,,SYD,,,BUSR,,0	RT3S	063	ROM /DF4,,,CBIO.MYC,,,,,SEQBUS,,,0	RT7S
024	ROM /101.ARR1.APB,,,,,WMEM,,BUSR,,CRADD	AD S	064	ROM /101.ARR1.AXB,,,,,WMEM,,BUSR,,CRLOG	XR S
025	ROM /101.AZ.APLB1,,,,,WMEM,,BUSR,,CRADD	IM	065	FTA AWA2.BSHR,0	DLN END
026	ROM /O77.ATWO.TWOA.CALU.MYC,,,,,0	RTN MASTER	066	FTA ARR1.AXB.CRLOG	TNM
027	ROM /19E.AW15.AMB,,SYD,,,GFSTOV,0	INT UPDATE A15	067	ROM /180.AWAD.ALUB,,MYQ,,,,,0	PUP
028	ROM FLAG./OD0.A7.ALUA.CALU,,QYC,,,,,GAEXI,0	TRAP SLN	068	ROM FLAG./ODD.AZ.ALUA.CALU,,QYC,,,,,GAEXL,0	MVF 15R2 TRAP
029	ROM /DE3.ATEN.TWOA.CALU.MYC.QYC,,,PYD,,,0	SLN	069	ROM /O5F.ARR2.ALUA.CALU,,QYC,,,PYD,,,0	MVF 15R2N
02A	ROM /159.ATEN.ACR.CALU.MYC.QYC,,,,,0	IPL SET	06A	ROM /OD6.AWAD.ALUB,,,PM2,,,0	DLNDRN END
02B	ROM /145.AEP.ALUA,,SYD,,,RBUS,,BUSR,GBOK,0	IPL END	06B	ROM FLAG./195.AWAL.ASHL,,SLQ,PP2,,,0	DLN LOOP
02C	ROM FLAG./102.AWCT,DBIO,,MYQ,SM2,PM2,CTPI,SEQBUS,,,0	MLRI	06C	ROM /OE4.AWR1.TWOA,,,,,CTPI,,REPEAT,,0	SLA
02D	ROM /ODC.ALUA,,SYD.PYD,,,BUSR,,0	MLRI	06D	ROM FLAG./107.AWAD.ALUB,,,,,0	SLN
02E	ROM /134,,CLUR.MYC,,,,,GCSEL,0	AUTO RESTART	06E	ROM /1FF,,,DBIO.CBIO.MYC,,,SP2,PP2,,,SEQBUS,,,0	READ M PUP A
02F	ROM FLAG./1CO.APUP.ALUA.CALU,,QYC,,,,,0	TEST IPL	06F	ROM /1FF,,,DBIO.CBIO.MYC,,,SEQBUS,,,0	READM PUP B
030	FTA AWR1.AMB.CRSUB	SU	070	ROM SNPLA./100.ARI215,,,,,GCTLD,0	SH DSH
031	ROM /1E9.AEP.AMB,,,SYD.PYD,,,BUSR,,CRSUB	SU P	071	ROM /18F.ARA2.ALUA.CALU,,QYC,,,,,GFKEYZ0,0	DSH
032	ROM SNPLA./100,,,CBIO.MYC.QYC,PP2,PP2,,,SEQBUS,,,0	RT2	072	ROM /1AD,,,DBIO,,,SYD,,,SEQBUS,BUSR,,0	RT6
033	ROM SNPLA./100,,,CBIO.MYC.QYC,PP2,PP2,,,SEQBUS,,,0	RT2S	073	ROM /1AC,,,DBIO,,,SYD,,,SEQBUS,BUSR,,0	RT6S
034	ROM /101.ARR1.AMB,,,,,WMEM,,BUSR,,CRSUB	SU S	074	FTA ARR1.ALUA.CRLOG	SHZERO
035	ROM FLAG./OBC.AEN.ALUA,,,,,GCRFNU,0	C2 OR NGR	075	ROM /ODC.ARA2.ALUA,,,,,BUSR,GCRFNU,0	DSHZERO
036	ROM /O77.ATWO.TWOA.CALU.MYC,,,,,0	RTN USER	076	ROM /O92.AWAL.DSUB,,,,,BUSR,,CRSUB	OS
037	FTA AWR1.AOB,0	INH.RTN A15 AND LC END	077	ROM /1C5.ATWO.ALUA.CALU.MYC,,,,,0	WAIT
038	ROM FLAG./ODD.AZ.ALUA.CALU,,QYC,,,,,GAEXL,0	TRAP SRN	078	ROM /OC7.AEP.ALUA.CALU.MYC,,,CTPI,,,0	ML
039	ROM /ODE.ARR1.ALUA.CALU,,QYC,,,,,0	SRN	079	ROM /150.ATWO.ASHR.CALU,,QYC,,,,,GBTMM,0	TL
03A	ROM FLAG./1C4,,,BSHR.CALU.MYC,,,,,GCRFNU,0	WAIT IO	07A	ROM /195.AWAL.ASHL,,SLQ,,,0	DRN
03B	ROM ETAT./1FF.AEP.ALUA,,SYD,,,0	WAIT IO	07B	ROM FLAG./185.AWAL.ASHR,,SRQ,PP2,,,GCRDSR,0	DRN LOOP
03C	ROM FLAG./1C2.AWCT,DBIO,,,SP2,PM2,CTPI,,SEQBUS,,GCRVML,0	ML LOOP	07C	ROM /OC5.AEP.ALUA.CALU.MYC,,,CTPI,,,0	MS
03D	ROM /ODC.ALUB,,,SYD.PYD,,,BUSR,,0	ML END	07D	ROM /148.ATWO.ASHR.CALU,,QYC,,,,,WEXM,,GBOM,0	TS
03E	ROM /15F.ATWO.ACR,,,PYD,,,0	IPL	07E	ROM /OC3.AEP.ALUA.CALU.MYC,,,CTPI,,,0	MSRD
03F	ROM FLAG./1B0,,,,,SLQ,,,RBUS,,GBOK,0	TEST LOADM	07F	ROM FLAG./1F4,,BCR.CALU,,QYC,,,,,GCRFNU	PUP
080	ROM /008,.ALUB,,,SYD,,,,,0	CIO QTR	080	ROM /008,.ALUB,,,SYD,,,,,0	CIO QTR
081	ROM /100,,,,,0	GO TO TRAP (WMP)	081	ROM /100,,,,,0	GO TO TRAP (WMP)
082	ROM /1B2.AWA2.DBIO,,,,,RBUS,SEQBIO,,,0	FFX LD A2	082	ROM /1B2.AWA2.DBIO,,,,,RBUS,SEQBIO,,,0	FFX LD A2
083	ROM SNPLA./100.ARA2.ALUA.CALU.MYC.QYC,,,,,0	RT1	083	ROM SNPLA./100.ARA2.ALUA.CALU.MYC.QYC,,,,,0	RT1
084	ROM /18B.AWR1.TWOA,,,,,CTPI,,REPEAT,,,0	SLL	084	ROM /18B.AWR1.TWOA,,,,,CTPI,,REPEAT,,,0	SLL
085	ROM /18B.AWR1.ASHL,,SLQ,,,CTPI,,REPEAT,,GCSEL,0	SLC	085	ROM /18B.AWR1.ASHL,,SLQ,,,CTPI,,REPEAT,,GCSEL,0	SLC
086	ROM /169,,,,,CTPI,,,0	RT1D	086	ROM /169,,,,,CTPI,,,0	RT1D
087	ROM /179.ARR2.ALUA.CALU,,QYC,,,,,GCTLD,0	RT1D	087	ROM /179.ARR2.ALUA.CALU,,QYC,,,,,GCTLD,0	RT1D
088	ROM FLAG./OB3,,,ZFR0.CALU.MYC,,,,,BUSR,GBTMF,0	FPP LOAD	088	ROM FLAG./OB3,,,ZFR0.CALU.MYC,,,,,BUSR,GBTMF,0	FPP LOAD
089	ROM /OBA.AWAL.ZERO.CALU,,QYC,,,,,GCRVZO,0	MUL-RESET PART.PROD	089	ROM /OBA.AWAL.ZERO.CALU,,QYC,,,,,GCRVZO,0	MUL-RESET PART.PROD
08A	ROM /130.ARAO.ALUA,,,,,WBUS,,BUSR,,0	EX T1 OR T3	08A	ROM /130.ARAO.ALUA,,,,,WBUS,,BUSR,,0	EX T1 OR T3
08B	ROM FLAG./OD0.AZ.ALUA.CALU,,QYC,,,,,GAEXL,0	EX T2 TRAP	08B	ROM FLAG./OD0.AZ.ALUA.CALU,,QYC,,,,,GAEXL,0	EX T2 TRAP
08C	ROM FLAG./OAB.ATEN.ACR.CALU.MYC.QYC,,,,,WEXM,,GBOF,0	FPP STORE	08C	ROM FLAG./OAB.ATEN.ACR.CALU.MYC.QYC,,,,,WEXM,,GBOF,0	FPP STORE
08D	ROM /1A8,,,,,SEQBUS,,GFRZO,0	LOAD M	08D	ROM /1A8,,,,,SEQBUS,,GFRZO,0	LOAD M
08E	ROM FLAG./104.AEP.ALUA,,SP2,,,SEQBUS,,GCRFNU,0	IPL LOOP	08E	ROM FLAG./104.AEP.ALUA,,SP2,,,SEQBUS,,GCRFNU,0	IPL LOOP
08F	ROM /159,,BSHR.CALU.MYC.SLQ,,,,,0	IPL LOOP	08F	ROM /159,,BSHR.CALU.MYC.SLQ,,,,,0	IPL LOOP
090	ROM /1AA.ATEN.ACR.CALU.MYC,,,,,0	INR	090	ROM /1AA.ATEN.ACR.CALU.MYC,,,,,0	INR
091	ROM SNPLA./103.AEP.ALUA,,SYD,,,,,0	EX BSYZO.RESTORE S	091	ROM SNPLA./103.AEP.ALUA,,SYD,,,,,0	EX BSYZO.RESTORE S
092	ROM /1ED.AEP.ALUA.CALU.MYC,,,,,RBUS,SEQBIO,,GFETCH,0	K-RCF L/RR	092	ROM /1ED.AEP.ALUA.CALU.MYC,,,,,RBUS,SEQBIO,,GFETCH,0	K-RCF L/RR
093	ROM SNPLA./100.AEP.ALUA.CALU.MYC.QYC,,,,,0	RT1P	093	ROM SNPLA./100.AEP.ALUA.CALU.MYC.QYC,,,,,0	RT1P
094	ROM /18B.AWR1.ASHR,,CTPI,,REPEAT,,GCRDSR,0	SRA	094	ROM /18B.AWR1.ASHR,,CTPI,,REPEAT,,GCRDSR,0	SRA
095	ROM FLAG./1C7.AWAD.ALUB,,,,,0	SRN	095	ROM FLAG./1C7.AWAD.ALUB,,,,,0	SRN
096	ROM SNPLA./100.ARCT.ALUA.CALU.MYC,,,,,0	RT1D END	096	ROM SNPLA./100.ARCT.ALUA.CALU.MYC,,,,,0	RT1D END
097	ROM /169.AEP.ALUA.CALU,,QYC,,,CTPI,,,0	RT1DP	097	ROM /169.AEP.ALUA.CALU,,QYC,,,CTPI,,,0	RT1DP
098	ROM FLAG./OB3,,,ZERO.CALU.MYC,,,,,BUSR,GBTMF,0	FPP OPER	098	ROM FLAG./OB3,,,ZERO.CALU.MYC,,,,,BUSR,GBTMF,0	FPP OPER
099	ROM /OAS.ARA2.TWOA.CALU,,QYC,,,,,0	DIV	099	ROM /OAS.ARA2.TWOA.CALU,,QYC,,,,,0	DIV
09A	ROM /O95.ARAL.ALUA,,,,,WBUS,,BUSR,GBTMF,0	FFL	09A	ROM /O95.ARAL.ALUA,,,,,WBUS,,BUSR,GBTMF,0	FFL
09B	ROM /OFE.AWR2.ALUB,,,,,0	DLN,DRN END	09B	ROM /OFE.AWR2.ALUB,,,,,0	DLN,DRN END
09C	ROM FLAG./OB3.ATWO.ASHR.CALU.MYC,,,,,BUSR,GBTMF,0	FPP OP/S	09C	ROM FLAG./OB3.ATWO.ASHR.CALU.MYC,,,,,BUSR,GBTMF,0	FPP OP/S
09D	ROM /143.AEP.ALUA,CLUR.MYC,SP2,,WMEM,SEQBUS,BUSR,GCSEL,0	INT	09D	ROM /143.AEP.ALUA,CLUR.MYC,SP2,,WMEM,SEQBUS,BUSR,GCSEL,0	INT
09E	ROM /160.AEP.ALUA.CALU.MYC,,,,,GFL0T,0	FFX	09E	ROM /160.AEP.ALUA.CALU.MYC,,,,,GFL0T,0	FFX
09F	ROM /O9B,,,,,RBUS,,GBOF,CRFLO	FFX UPD CR	09F	ROM /O9B,,,,,RBUS,,GBOF,CRFLO	FFX UPD CR
0A0	ROM /105,,,ZERO,,,SYD,,,GFSYS,0	IPL	0A0	ROM /105,,,ZERO,,,SYD,,,GFSYS,0	IPL
0A1	ROM /1E9.AEP.APB,,,SYD.PYD,,,BUSR,,0	RF	0A1	ROM /1E9.AEP.APB,,,SYD.PYD,,,BUSR,,0	RF
0A2	ROM /OE7.ATWO.ALUA.CALU.MYC,,,,,0	RT3B	0A2	ROM /OE7.ATWO.ALUA.CALU.MYC,,,,,0	RT3B
0A3	ROM SNPLA./100.ARR2.ALUA.CALU,,QYC.SYD,,,,,0	RT3C	0A3	ROM SNPLA./100.ARR2.ALUA.CALU,,QYC.SYD,,,,,0	RT3C
0A4	ROM /OD8.AWAL.ASHL,,SLQ,,,CTPI,,REPEAT,,,0	DLA	0A4	ROM /OD8.AWAL.ASHL,,SLQ,,,CTPI,,REPEAT,,,0	DLA
0A5	ROM FLAG./O1F.ATWO.ALUA,,PYD,,,,,0	DLN	0A5	ROM FLAG./O1F.ATWO.ALUA,,PYD,,,,,0	DLN
0A6	ROM /155,,BSHR.CALU.MYC,,,PM2,,,0	IPL LOOP	0A6	ROM /155,,BSHR.CALU.MYC,,,PM2,,,0	IPL LOOP
0A7	ROM /OCB.AEP.ALUA.CALU,,QYC,,,,,0	RT3BM	0A7	ROM /OCB.AEP.ALUA.CALU,,QYC,,,,,0	RT3BM
0A8	ROM /O80,,,,,BUSR,GBEX,0	EL	0A8	ROM /O80,,,,,BUSR,GBEX,0	EL
0A9	ROM /O8E,,,CBIO,,QYC,SP2,,,SEQBUS,BUSR,,0	DA	0A9	ROM /O8E,,,CBIO,,QYC,SP2,,,SEQBUS,BUSR,,0	DA
0AA	ROM /14F,,,BSHR.CALU.MYC,,,,,0	IPL LOOP	0AA	ROM /14F,,,BSHR.CALU.MYC,,,,,0	IPL LOOP
0AB	ROM /152,,,CBIO.MYC,SP2,PP2,,,SEQBUS,,,0	DAK	0AB	ROM /152,,,CBIO.MYC,SP2,PP2,,,SEQBUS,,,0	DAK
0AC	ROM /101.ARR1.ALUA,,,,,WMEM,,BUSR,GBEX,0	ES	0AC	ROM /101.ARR1.ALUA,,,,,WMEM,,BUSR,GBEX,0	ES
0AD	ROM /O94.AWA2.APB.CALU.MYQ.QYC,,,,,0	DA	0AD	ROM /O94.AWA2.APB.CALU.MYQ.QYC,,,,,0	DA
0AE	ROM /1E9.AEP.ALUA,,SYD,,,SEQBUS,,,0	TL END	0AE	ROM /1E9.AEP.ALUA,,SYD,,,SEQBUS,,,0	TL END
0AF	ROM FLAG./150,,,,,SLQ,SP2,,,SEQBUS,BUSR,GBTMM,0	TL LOOP	0AF	ROM FLAG./150,,,,,SLQ,SP2,,,SEQBUS,BUSR,GBTMM,0	TL LOOP
0B0	ROM /140,,BSHR.CALU.MYC,,,,,0	IPL LOOP	0B0	ROM /140,,BSHR.CALU.MYC,,,,,0	IPL LOOP
0B1	ROM /1E9.AEP.AMB,,,SYD.PYD,,,BUSR,,0	RB	0B1	ROM /1E9.AEP.AMB,,,SYD.PYD,,,BUSR,,0	RB
0B2	ROM FLAG./170.AIPL.APB.CALU.MYC,,,,,WMEM,,,0	IPL NEW QUART	0B2	ROM FLAG./170.AIPL.APB.CALU.MYC,,,,,WMEM,,,0	IPL NEW QUART
0B3	ROM SNPLA./100,,,,,PP2,,,0	RT2C	0B3	ROM SNPLA./100,,,,,PP2,,,0	RT2C
0B4	ROM /OD2.AWAL.ASHR,,SRQ,,,CTPI,,REPEAT,,GCRDSR,0	DRA	0B4	ROM /OD2.AWAL.ASHR,,SRQ,,,CTPI,,REPEAT,,GCRDSR,0	DRA
0B5	ROM FLAG./ODF.ATWO.ALUA,,PYD,,,,,0	DRN	0B5	ROM FLAG./ODF.ATWO.ALUA,,PYD,,,,,0	DRN
0B6	ROM /1E9.AEP.ALUA,,SYD,,,SEQBUS,,0	TS END	0B6	ROM /1E9.AEP.ALUA,,SYD,,,SEQBUS,,0	TS END
0B7	ROM FLAG./148,,,,,SLQ,SP2,,,WEXM,SEQBUS,BUSR,GBOM,0	TS LOOP	0B7	ROM FLAG./148,,,,,SLQ,SP2,,,WEXM,SEQBUS,BUSR,GBOM,0	TS LOOP
0B8	ROM FLAG./ODD.AZ.ALUA.CALU,,QYC,,,,,GAEXL,0	TRAP	0B8	ROM FLAG./ODD.AZ.ALUA.CALU,,QYC,,,,,GAEXL,0	TRAP
0B9	ROM /O8A,,,CBIO,,QYC,SP2,,,SEQBUS,BUSR,,0	DS	0B9	ROM /O8A,,,CBIO,,QYC,SP2,,,SEQBUS,BUSR,,0	DS
0BA	ROM /1E9.AW15.DBIO,,,,,SEQBUS,,,0	IPL END	0BA	ROM /1E9.AW15.DBIO,,,,,SEQBUS,,,0	IPL END
0BB	ROM /142,,,CBIO.MYC,SP2,PP2,,,SEQBUS,,,0	DSK	0BB	ROM /142,,,CBIO.MYC,SP2,PP2,,,SEQBUS,,,0	DSK
0BC	ROM /13D,,BSHR,,,SYD,,,SEQBUS,,,0	INT	0BC	ROM /13D,,BSHR,,,SYD,,,SEQBUS,,,0	INT
0BD	ROM /O8B.AWA2.AMB.CALU.MYQ.QYC,,,,,0	DS	0BD	ROM /O8B.AWA2.AMB.CALU.MYQ.QYC,,,,,0	DS
0BE	ROM /O9E.AZ.APLB1.CALU.MYC,,,,,BUSR,GCRVML,CRADD	DIV COR.QUOT	0BE	ROM /O9E.AZ.APLB1.CALU.MYC,,,,,BUSR,GCRVML,CRADD	DIV COR.QUOT
0BF	ROM /1E9.AWA2.ALUB,,,,,BUSR,GCRVML,CRLOG	DIV QUOT OK	0BF	ROM /1E9.AWA2.ALUB,,,,,BUSR,GCRVML,CRLOG	DIV QUOT OK

Table 2-4A contd.

ADD.		ADD.		ADD.	
0C0	FTA AWR1,ALUB,0 LDK	100	ROM /03F,.....,GFSYS,0 VERIF MAN. PUPIT	140	ROM /07E,.....,RBUS,,GBOK,0 TEST MAN REG M
0C1	ROM /1E9,ALUB,.....,SYD,PYD,.....,BUSR,,0 LDK P	101	ROM /19A,ARAO,ALUA,MYQ,.....,PYD,.....,BUSR,,0 DLN,DRN END	141	ROM /1E9,AWR2,AMB,.....,BUSR,GFSTOV,0 UPD STK MSRD
0C2	ROM /135,BCR,.....,BUSR,GFPLR,0 AR PAF INT	102	ROM /0EC,ARR2,APB,.....,SYD,.....,BUSR,,0 RT5	142	ROM /1CF,AWR1,ZERO,.....,BUSR,,0
0C3	ROM /0EE,.....,CBIO,MYC,.....,PP2,.....,SEQBUS,.....,0 RT5C	103	ROM /1AD,ARR2,APB,.....,SYD,.....,BUSR,,0 RT7	143	ROM /101,AZ,AMB,.....,WMEM,.....,BUSR,CRSUB
0C4	ROM /04F,AWA2,APB,.....,0 MVF DEST IN A2	104	ROM /0D3,.....,CBIO,MYC,.....,RBUS,SEQBIO,.....,0 TMP TEST RD KEYS	144	ROM /188,ALUR,.....,WBUS,SEQBIO,.....,CRIO CIO QTR
0C5	ROM /1E9,ARAO,ALUA,.....,SYD,PYD,.....,BUSR,,0 MOVE END SLN REST.P	105	ROM FLAG,0D4,ARAO,ALUA,.....,GCRFNU,0 TEST RETURN	145	ROM /0B7,ARAZ,ALUA,CALU,MYC,.....,0 MU- MUL.TOR IN Q REG.
0C6	ROM /088,ARR1,ALUA,CALU,MYC,.....,0 CA	106	ROM FLAG,0DD, AZ,ALUA,CALU,MYC,.....,GAEXL,0 EX OPER T4T7 TRAP	146	ROM FLAG,01C,ATWO,APB,MYC,.....,GCRFNU,0 TEST NO 1
0C7	FTA ARR1,AMB,CRCMP CA	107	ROM /066,BSHR,CALU,MYC,.....,0 EX OPER T1,T2,T3	147	ROM /018,.....,SLQ,.....,0 TEST NO 1
0C8	ROM /086,ARR1,ACR,CLUR,MYC,.....,BUSR,GBCH,0 LC	108	ROM FLAG,0DD, AZ,ALUA,CALU,MYC,.....,GAEXL,0 MVB 15R2FU TRAP	148	ROM /086,AWA1,MULTI,SRQ,CTP1,REPEAT,.....,0 MU- ALGORITHM
0C9	FTA AWR1,BCR,0 ECR	109	ROM /048,ARR2,ALUA,.....,PYD,.....,0 MVB 15R2FUN	149	ROM /085,AWA1,MULTI,MYQ,.....,GMULTI,0 MULTI 16TH PASS
0CA	ROM /1E9,.....,DBIO,.....,SYD,PYD,.....,SEQBUS,BUSR,,0 AR PAF INT TRAP	10A	ROM /0E8,ARR2,APB,.....,SYD,.....,BUSR,,0 RT55	14A	ROM /084,AWA2,BSHR,.....,0 MULTI STORE LSB OF PROD.
0CB	ROM /132,.....,GFSYS,0 AR END	10B	ROM /1AC,ARR2,APB,.....,SYD,.....,BUSR,,0 RT7S	14B	ROM /0D7,ARAZ,ALUA,.....,GCRFNU,0 MULTI UPDATE FNU
0CC	ROM /101,ARR1,ALUA,.....,WMEM,.....,BUSR,GBCH,0	10C	ROM /04A,ATWO,ALUA,CALU,MYC,MYC,.....,0 MVB LENGTH>0	14C	ROM /0AD,ALUB,SRQ,SP2,SEQBUS,BUSR,GBTMF,0 FPP SINGLE RD MA
0CD	ROM /13D,BSHR,.....,SYD,.....,GFKYZO,0 PAGE FAULT AR	10D	ROM /1E9,ARAO,ALUA,.....,SYD,PYD,.....,BUSR,,0 MVB LENGTH=0	14D	ROM /042,.....,SP2,.....,SEQBUS,BUSR,GBTMF,0 FPP OP/S DOUBLE READ M1
0CE	ROM FLAG,128,ARAO,APB,CLUR,MYC,SLQ,.....,0 EX T&N TEST K1	10E	ROM /1EB,ARAO,AMB,.....,PYD,.....,0 INT MOVE	14E	ROM /053,ATWO,ASHR,MYC,MYC,SM2,PM2,WMEM,.....,0 T. MEMOIRE NO1
0CF	ROM /16E,.....,CIOR,MYC,MYC,.....,WBUS,SEQBIO,GFETCH,0 EX LOAD K	10F	ROM /125,ATWO,TWOA,CALU,MYC,.....,0 PAF MOVE	14F	ROM /06C,AWA1,ALUB,.....,SM2,PM2,.....,0 TEST MEMOIRE NO1
0D0	ROM FLAG,0DD, AZ,ALUA,CALU,MYC,.....,GAEXL,0 TRAP	110	ROM /0DF,.....,GFRZO,0 VERIF MAN. REG L	150	ROM /09F,.....,GFRZO,0 TEST Q,CHARG.NO
0D1	ROM FLAG,0DD, AZ,ALUA,CALU,MYC,.....,GAEXL,0 TRAP	111	ROM SNPLA,100,ARR2,APB,CALU,MYC,SYD,.....,0 RT5C	151	ROM /0A1,AWA1,BCR,MYC,MYC,.....,0 TEST DLA
0D2	ROM /125,AWR2,ALUB,.....,0 PAGE FAULT ML	112	ROM /12C,ARR2,APB,.....,SYD,.....,BUSR,,0 RT7C	152	ROM FLAG,030,.....,MYQ,SP2,SEQBUS,BUSR,GBTMF,0 FPP READ LAST MA
0D3	ROM SNPLA,100,.....,DBIO,.....,CBIO,MYC,SYD,PP2,SEQBUS,.....,0 RT4C	113	ROM SNPLA,100, AEP,ALUA,CBIO,MYC,MYC,SYD,SEQBUS,.....,0 RT3	153	ROM /173,ARAO,TWOA,.....,SYD,.....,0 FPP OPS WAIT.LD S WITH 1ST AD.
0D4	ROM FLAG,118,AWA2,AMB,CBIO,MYC,SYD,SEQBUS,GMOVE,0 MVB READ	114	ROM /048,AWA2,APB,CBIO,MYC,SYD,SEQBUS,GMOVE,0 MVB CORRECT A2	154	ROM /010,.....,BINV,CALU,MYC,SP2,.....,WEXM,SEQBUS,BUSR,GBOF,0 FPP/S SI.
0D5	ROM /056,AWA1,APB,.....,0 MVF REST A1	115	ROM /056,AWA2,APB,.....,0	155	ROM /010,.....,SP2,.....,WEXM,SEQBUS,BUSR,GBOF,0 FPP/S DOUBLE ST M1
0D6	ROM FLAG,120,.....,SLQ,.....,0 EX K1 TEST K2	116	ROM /0AE,ATEN,ALUA,MYC,.....,GFRZO,0 TEST DLA	156	ROM /0E9,.....,0
0D7	ROM FLAG,0F8,BCR,CALU,MYC,.....,0 EX K1N TEST MD	117	ROM SNPLA,100,.....,CBIO,MYC,MYC,.....,SEQBUS,.....,0 RT3S	157	ROM /0A7,ARAO,AOB,.....,SYD,.....,RBUS,.....,BUSR,GBTMP,0 INR
0D8	FTA ARR1,AMB,CRCMP CW	118	ROM /0EC,ARR2,APB,.....,SYD,.....,BUSR,,0 RT3B UPDATE STACK	158	ROM /0A6,AWR1,DBIO,CIOR,MYC,.....,SEQBUS,GCSEL,CRIO INR
0D9	FTA AEP,AMB,CRCMP CWP	119	ROM /1F3,AWR2,AMB,.....,GFSTOV,0 STD	159	ROM /188,AWR1,AOB,.....,SLQ,.....,0 INR TST SST
0DA	ROM /124,APSW,ALUA,CALU,MYC,.....,GFSYS,0 PAGE FAULT	11A	ROM /1F2,AWR2,AMB,.....,GFSTOV,0 STDP	15A	ROM /0A4,AWA1,ALUA,.....,GCRVZO,0 DIV MEMO SIGNE DIVD
0DB	ROM /122,AWR2,ALUB,.....,SYD,.....,WEXM,.....,BUSR,GBOM,0 PAGE FAULT	11B	ROM /1E9,ARR1,ALUA,.....,BUSR,6CRVML,CRL06 SLA CR	15B	ROM /0A3,AWA1,DIVSH,SLQ,CTP1,REPEAT,.....,0 DIV 1ST PASS
0DC	ROM /08C,ARR1,ALUA,CLUR,MYC,.....,BUSR,GBCH,0 CC	11C	ROM FLAG,185,ARR1,ALUA,.....,0 SLN TEST NORM	15C	ROM /0A2,AWA1,DIVSH,SLQ,CTP1,REPEAT,.....,0 DIV PROCESS
0DD	ROM /11D,ALUB,MYQ,SM2,WMEM,SEQBUS,BUSR,0 PAGE FAULT	11D	ROM /13A,AWR2,BSHR,.....,0 SLN RESULT	15D	ROM FLAG,186,AWA1,DIVALU,SLQ,.....,GCRFNU,0 DIV 16TH PASS
0DE	ROM FLAG,118,ALUB,MYQ,SM2,WMEM,SEQBUS,BUSR,0 EX K1K2 TEST K3	11E	ROM /1EC,AW1215,DBIO,.....,RBUS,SEQBIO,.....,0 KEYS IN (RCP) LR	15E	ROM /0A0,ATWO,APB,CLUR,MYC,.....,PYD,.....,0 TEST DLA
0DE	ROM FLAG,0F8,BCR,CALU,MYC,.....,0 EX K1K2N TEST MD	11F	ROM /1FF,AR1215,ALUA,MYC,.....,GFKYZO,0 LOAD OR READ R	15F	ROM /091,AWA2,BCR,SRQ,.....,0 TEST DLA
0E0	ROM /07B,ALUB,.....,SYD,.....,0 WER	120	ROM /009,.....,RBUS,.....,GBOK,0 VERIF MAN. REG L	160	ROM /039,.....,RBUS,.....,GBOK,0 TEST Q,CHARG NO.
0E1	ROM FLAG,197,AEP,ALUA,CALU,MYC,.....,0 MVF	121	ROM FLAG,1A5,ATEN,TWOA,CALU,MYC,.....,PYD,.....,0 SRM	161	FTA AWA2,ALUB,0 DIV ST CORTD QUOT
0E2	ROM /117,ATWO,APB,CALU,MYC,SM2,.....,WEXM,SEQBUS,BUSR,GBOM,0 PAF	122	ROM /188,AWR1,ASHR,SRQ,CTP1,REPEAT,GCRRSR,0 SRLC	162	ROM /09F,.....,0 TMP TEST
0E3	ROM /0ED,.....,CBIO,MYC,.....,SEQBUS,.....,0 RT7C	123	FTA ARAL,ALUA,CRL06 ML DSH	163	ROM /033,AZ,ALUA,MYC,.....,0 TMP TEST
0E4	ROM /057,ALUB,MYQ,PM2,WMEM,.....,BUSR,,0 MVF PREP WRITE	124	ROM /0DA,.....,MYQ,.....,0 DLA END	164	ROM /09A,AWA1,DBIO,.....,RBUS,SEQBIO,GBOF,U FFX LD A1
0E5	ROM /057,ALUB,MYQ,PM2,WMEM,.....,BUSR,GBEX,0 MVSU PREP WRITE	125	ROM /0D7,AWA2,BSHR,CALU,MYC,.....,GCRFNU,0 DLA END	165	ROM /170,.....,RBUS,.....,BUSR,GBOF,0 FFX RESET BSY
0E6	ROM FLAG,108,.....,SLQ,.....,0 EX K1K2K3 TEST K4	126	ROM /019,.....,BINV,MYC,.....,0 ROUT.AFFICH,INCR	166	ROM /022,AWCT,ALUB,.....,CTP1,.....,0 TEST NO 3
0E7	ROM FLAG,0F8,BCR,CALU,MYC,.....,0 EX K1K2K3N TEST MD	127	ROM ETAT,07E,ALUB,CALU,MYC,.....,SEQBUS,.....,0 DISPLAY INCR	167	ROM /023,AWCT,ALUB,.....,SLQ,CTP1,.....,0 TEST NO 3
0E8	ROM /132,AW15,AMB,CIOR,MYC,.....,RBUS,SEQBIO,GFSTOV,0 PAGE FAULT	128	ROM /1E9,ARR1,ALUA,.....,BUSR,6CRVML,CRL06 DLA END	168	ROM ETAT,0DE,.....,SEQBUS,.....,0 VERIF MAN REG L
0E9	ROM /1E9,AEP,ALUA,.....,SYD,.....,SEQBUS,BUSR,CRRTN RTN 15R2N	129	ROM /164,AEP,ASHR,CALU,MYC,.....,0 DLN,DRN END	169	ROM FLAG,09C,APSW,AANDB,.....,GCRFNU,0 TEST TMP-TPM
0EA	ROM /110,ATEN,TWOA,CALU,MYC,PM2,.....,0 TRAP	12A	ROM /041,ARAL,ACR,CLUR,MYC,.....,0 TEST RB	16A	ROM /030,ARAZ,ALUA,.....,WBUS,SEQBUS,BUSR,GBTMF,0 FFL LD M1
0EB	ROM /073,AEP,ALUA,CBIO,MYQ,MYC,SYD,.....,SEQBUS,GFPLR,CRRTN RTNA15	12B	ROM FLAG,034,ARAZ,ALUA,.....,GCRFNU,0 TEST DLA	16B	ROM /093,AWA2,TWOA,.....,GCRFNU,0 DA
0EC	ROM /02E,ATWO,TWOA,CALU,MYC,.....,0 CF	12C	ROM /043,BCR,.....,SYD,.....,WBUS,.....,0 TEST TMP-TPM	16C	ROM /092,AWA1,DADD,.....,BUSR,CRADD DA
0ED	ROM FLAG,130,AWA2,ALUB,.....,WBUS,.....,0 EX	12D	ROM /0CE,ZERO,SRQ,.....,0 DRA	16D	FTA AWA2,ASHR,0 DAS END
0EE	ROM /06B,ATWO,TWOA,CALU,MYC,.....,0 CF 15R1	12E	ROM /115,ATWO,TWOA,CALU,MYC,PM2,.....,0 TRAP CORR LONG EX.	16E	ROM /090,AEP,TWOA,MYC,.....,0 TEST DLA
0EF	ROM /10D,AW15,AMB,.....,SLQ,SYD,.....,GFSTOV,0 TRAP	12F	ROM /115,ATWO,TWOA,CALU,MYC,.....,0 TRAP	16F	ROM /081,AEP,APB,MYQ,.....,PYD,.....,0 TEST DLA
0F0	ROM /04E,ATEN,ACR,CALU,MYC,.....,0 RER	130	ROM /0BF,.....,GFRZO,0 VERIF MAN. REG M	170	ROM /019,BSHR,CALU,MYC,.....,0 ROUT.AFFICH,INCR
0F1	ROM FLAG,0F7,AEP,ALUA,CALU,MYC,.....,0 MVB	131	ROM /0CD,.....,MYQ,.....,0 DSH	171	ROM /152,AEP,ALUA,CBIO,MYC,SYD,SEQBUS,.....,0 DAR* DA
0F2	ROM /105,APSW,ALUA,.....,SLQ,SP2,.....,WMEM,.....,BUSR,6FYSYS,0 TRAP	132	ROM /0DC,AWA2,ALUB,.....,BUSR,6CRFNU,0 DSH	172	ROM /1F7,AEP,ALUA,CBIO,MYC,SYD,SEQBUS,BUSR,.....,0 EL
0F3	ROM /12C,.....,DBIO,.....,SYD,.....,SEQBUS,BUSR,.....,0 RT6C	133	ROM /038,.....,MYQ,.....,0 TEST Q,CHARG NO.	173	ROM /088,AEP,ALUA,CIOR,MYC,SYD,SEQBUS,.....,0 CC
0F4	ROM /0EB,ARAL,ALUA,.....,SYD,.....,BUSR,,0 MVB PREP 1ST AD	134	ROM /0CA,AEN,ALUA,CALU,MYC,.....,PYD,CTP1,.....,0 MLRI	174	ROM /189,AWA2,TWOA,.....,GCRFNU,0 DS
0F5	ROM /0EB,ARAL,ALUA,.....,SYD,.....,BUSR,GBEX,0 MVSU PREP 1ST AD	135	ROM /1D3,AWR2,APB,.....,SYD,PM2,.....,BUSR,,0 TEST DLA	175	ROM /142,AEP,ALUA,CBIO,MYC,SYD,SEQBUS,.....,0 DSR
0F6	ROM FLAG,0F8,BCR,CALU,MYC,.....,0 EX K1K2K3K4 TEST MD	136	ROM /0E9,.....,0	176	ROM /0E9,.....,0 TEST DLA
0F7	ROM FLAG,0DD, AZ,ALUA,CALU,MYC,.....,GAEXL,0 EX 150PC TRAP	137	ROM /0C7,AEP,APB,CALU,MYC,.....,CTP1,.....,0 MLK	177	ROM /087,AWAD,ALUB,MYQ,.....,BUSR,,0 CW CA
0F8	ROM /0C8,AEN,ALUA,CALU,MYC,PM2,.....,0 MLK	138	ROM /0C6,AEN,ALUA,.....,PYD,.....,0 ML	178	FTA ARAO,AMB,CRCMP CW CA
0F9	FTA AWR1,BINV,CRL06 C1	139	ROM /1C3,ARR2,ALUA,CALU,MYC,PM2,.....,BUSR,,0 ML	179	ROM /083,AWR1,BCR,CIOR,MYC,.....,SEQBUS,.....,0 LC
0FA	ROM /104,AEP,ALUA,MYQ,SP2,.....,WMEM,SEQBUS,BUSR,.....,0 TRAP	13A	ROM /0C4,AEN,ALUA,.....,PYD,.....,0 MS	17A	ROM /027,ZERO,MYQ,SRQ,.....,0 TEST NO 2
0FB	ROM /103,ALUB,.....,SYD,.....,SEQBUS,.....,0 TRAP	13B	ROM /183,ARCT,ALUA,.....,PM2,CTP1,WMEM,.....,BUSR,,0 MS	17B	ROM /026,ALUB,MYQ,.....,PYD,.....,0 TEST NO 2
0FC	ROM /135,.....,SM2,.....,BUSR,,0 TRAP	13C	ROM /0C2,AEN,ALUA,CALU,MYC,.....,PYD,.....,0 MSRD	17C	ROM /1C8,AEP,ALUA,.....,SYD,.....,BUSR,,0 LC
0FD	ROM /101,.....,BINV,.....,WMEM,.....,BUSR,.....,CRL06 C1S	13D	ROM /1A3,ARCT,ALUA,.....,PM2,CTP1,WMEM,.....,BUSR,,0 MSRD	17D	FTA AWR1,AOB,0 LC END
0FE	ROM /1E9,AEP,ALUA,.....,SYD,.....,SEQBUS,BUSR,.....,0 STORE RESULT	13E	ROM FLAG,00A,AEP,ALUA,.....,WMEM,.....,0 TEST MEMOIRE NO2	17E	ROM /080,AEP,APB,.....,PYD,.....,WBUS,.....,0 TEST DLA
0FF	ROM FLAG,0DD, AZ,ALUA,CALU,MYC,.....,GAEXL,0 TRAP	13F	ROM /053,ARAL,TWOA,MYC,MYC,SM2,PM2,WMEM,.....,0 TEST MEMOIRE NO1	17F	ROM /18E,AWAD,ZERO,.....,SEQBIO,GFETCH,0 TEST DLA

Table 2-4A contd.

ADD.		ADD.	
180	ROM /028...MYQ...RBUS...GBOK,0	1C0	ROM /1E8...RBUS...GBOK,0
181	ROM /020...CB10.MYC...SEQBUS...0	1C1	ROM /037.AWA1.AMB...0
182	ROM /050...MYQ...0	1C2	ROM FLAG./02C..BSHR..MYC.QYC...GCRFNU,0
183	ROM /0E9...0	1C3	ROM /085.ATW0.ACR..MYC.QYC...GFRZO,0
184	ROM /07A.ARR1.ALUA...WBUS..BUSR..GBTME,0	1C4	ROM FLAG./0F1.AWR2.ALUB..MYQ...0
185	ROM /188.ARR1.ALUA...WBUS.SEQBUS...0	1C5	ROM /188.AWR1.DB10.CB10...SEQBUS...CRRTN
186	ROM FLAG./054.AR15.AMB...SRQ...GCRFNU,0	1C6	ROM /0CC...CB10..QYC...SEQBUS...0
187	ROM FLAG./064.ARCT.AMB...SLQ...CTP1...GCRFNU,0	1C7	ROM /029..ALUB..CLUR..MYC...WBUS...0
188	ROM /076.AWR2.APB...SYD...BUSR...0	1C8	ROM /036.AWA2.AMB...0
189	ROM SNPLA./100..DB10...SM2.PYD...SEQBUS.BUSR..0	1C9	ROM /03B.ATW0.APB.CALU.MYC...0
18A	ROM /09F...0	1CA	ROM /09F...0
18B	ROM /01B.ZERO..SRQ..PM2...0	1CB	ROM /071.ATW0.ASHR..MYC...0
18C	ROM /072.ATW0.APB.CALU.MYC...GFENB,0	1CC	ROM /07F..BINV...QYC...0
18D	ROM /06E.AWA0.BINV.MYQ...0	1CD	ROM /098.ATW0.ASHR..MYC.QYC...GFRZO,0
18E	ROM FLAG./024.ARA1.AXB...GCRFNU,0	1CE	ROM /0AC.AWA0.ALUB...SEQBUS..GFLOT,0
18F	ROM /061.ATW0.APLB1..MYC...0	1CF	ROM /188.AEP.ALUA...SYD...SEQBUS..GFLOT,0
190	ROM /019..BINV..MYC...0	1D0	ROM /02F...0
191	ROM /18E.ARA0.AOB.CLUR.MYC.QYC...0	1D1	ROM /06A.AWR1.AMB...SYD...0
192	ROM /09F...0	1D2	ROM FLAG./07C..BSHR..MYC.SRQ...GCRFNU,0
193	ROM FLAG./002...CB10.MYC...SEQBUS...0	1D3	ROM /032.ARA0...GCTLD,0
194	ROM /06A.AWR1.AMB...SYD...GFSTOV,0	1D4	ROM FLAG./0EA.AWA1.APB...SYD...SEQBUS..GMOVE,0
195	ROM /068.APSW.ALUA...SP2..WMEM..BUSR..0	1D5	ROM FLAG./0EA.AWA1.APB...SYD...SEQBUS..GBEX,0
196	ROM /0E9...0	1D6	ROM ETAT./09E...SEQBUS...0
197	ROM /067.AEP.ALUA..MYQ..SP2..WMEM.SEQBUS.BUSR..0	1D7	ROM FLAG./0D8..ALUB.CB10..QYC...WBUS.SEQBUS...0
198	ROM /1E9..ALUB...SYD.PYD...SEQBUS.BUSR..0	1D8	ROM FLAG./084.ATEN.AXB...GCRFNU,0
199	ROM /05E..BSHR.CALU.MYC...0	1D9	ROM /017.AEP.AMB...PYD...0
19A	ROM /09F...0	1DA	ROM /09F...0
19B	ROM FLAG./078...MYQ...0	1DB	ROM /070.ATEN.TWOA..MYC...0
19C	ROM FLAG./05C.AEP.AXB...SM2.PM2...GCRFNU,0	1DC	ROM FLAG./098...MYQ...0
19D	ROM /0FB...RBUS...GBOK,0	1DD	ROM /078.ATW0.ASHR..MYC.QYC...0
19E	ROM /060.AWA0.BCR..MYC...0	1DE	ROM FLAG./00C..BINV...0
19F	ROM /051.AEP.APB..MYC..PYD..WBUS...0	1DF	ROM /049..ALUB...WBUS...0
1A0	ROM FLAG./13A.AWA0.ALUB..MYQ..PM2...0	1E0	ROM /100..ALUB...PYD...0
1A1	ROM FLAG./174.ATEN.AXB...GCRFNU,0	1E1	ROM FLAG./195.ARA1.ALUA...0
1A2	ROM /09F...0	1E2	ROM /09F...0
1A3	ROM FLAG./062...CB10.MYC...SEQBUS...0	1E3	ROM /021..BINV..MYC...0
1A4	ROM /05A.ARR2.AMB.CALU.MYC...0	1E4	ROM FLAG./01A...PM2...0
1A5	ROM /12B.AWA1.APB..MYQ..SYD...BUSR..0	1E5	ROM /033...MYQ...0
1A6	ROM /096.ATW0.ACR.CALU.MYC...WBUS.SEQB10..GFKYZO,0	1E6	ROM FLAG./04C..BSHR..MYC...GCRFNU,0
1A7	ROM FLAG./058...SLQ...WBUS...GBTMP,0	1E7	ROM /04C...MYQ...0
1A8	ROM FLAG./12A.AWA1.AMB...SYD...SEQBUS..GMOVE,0	1E8	ROM /016.AEP.APB.CALU.MYC.SRQ...0
1A9	ROM /13A.AWR2.ZERO...0	1E9	ROM FLAG./074.ATEN.AXB...GCRFNU,0
1AA	ROM /09F...0	1EA	ROM /09F...0
1AB	ROM /013.AWA2.FORA...0	1EB	ROM /033.ATEN.ASHR.CALU.MYC...0
1AC	ROM FLAG./052..ALUB...SM2.PM2..WMEM.SEQBUS...0	1EC	ROM /012.ATEN.ALUA..MYC.QYC...0
1AD	ROM /080.ARA0.ALUA...SYD.PYD...0	1ED	ROM FLAG./044.ARA2.AMB...SRQ...GCRFNU,0
1AE	ROM SNPLA./1FF.AWA1.ALUB.CIOR.MYC.QYC..PM2...SEQB10..GFETCH,0	1EE	ROM /1E9.AEP.ALUA...SYD...SEQBUS..CRFLO
1AF	ROM /081.AWA0.ALUB...SYD.PYD...GFRZO,0	1EF	ROM FLAG./010...SLQ.SP2...WEXM.SEQBUS.BUSR..GBOF,0
1B0	ROM /05B.ATW0.ALUA.CALU.MYC.QYC...0	1F0	ROM /100..ALUB...PYD...0
1B1	ROM /046.AWA0.BSHR..MYQ...0	1F1	ROM FLAG./185.AWA1.ASHR...SRQ...GCRDSR,0
1B2	ROM FLAG./03C..BSHR..MYC...GCRFNU,0	1F2	ROM /09F...0
1B3	ROM FLAG./088.ATW0.AOB..MYC.QYC...GFRZO,0	1F3	ROM /07F...0
1B4	ROM FLAG./0F2.AWA0.ALUB...PM2...0	1F4	ROM /0C1...SM2.PM2...SEQBUS...0
1B5	ROM FLAG./108.AWA2.AMB...0	1F5	ROM /063..ALUB...SYD.PYD...0
1B6	ROM ETAT./0BE...SEQBUS...0	1F6	ROM /097..DB10...WBUS.SEQBUS...0
1B7	ROM FLAG./02B..ALUB..MYQ..PM2..WMEM..BUSR..0	1F7	ROM /08B.ARR1.ALUA.CALU.MYC...WBUS..BUSR..GBTMP,0
1B8	ROM /162..ALUB...WMEM..BUSR..0	1F8	ROM /036.AEP.ALUA.CALU.MYC...GCTLD,0
1B9	ROM /03A.ARA0.AOB...SYD...RBUS..BUSR..GBTME,0	1F9	ROM /007.AWA1.AMB...PP2...GCTLD,0
1BA	ROM /09F...0	1FA	ROM /03E.AEP.ALUA.CALU.MYC...GCTLD,0
1BB	ROM /01A.ZERO..SRQ...0	1FB	ROM /005.AWA1.APB...PP2...GCTLD,0
1BC	ROM /058.ATEN.ALUA.CALU.MYC...WBUS.SEQB10..GFETCH,0	1FC	ROM FLAG./06C.ARA1.AXB...SM2.PM2...GCRFNU,0
1BD	ROM /0B3...SP2...SEQBUS.BUSR..GBTMP,0	1FD	ROM FLAG./0C0.ARA0.ALUA.CALU.MYC..SYD.PYD...0
1BE	ROM /040..BCR..MYC...GFKYZO,0	1FE	ROM /120.ATW0.TWOA.CALU.MYC.QYC...GCTLD,0
1BF	ROM FLAG./014.AEP.AXB..MYQ...GCRFNU,0	1FF	ROM /125.ATW0.TWOA.CALU..QYC...GCTLD,0
	DISPLAY INCR		VERIF MAN. PUPIT
	VERIF MAN REGM		DER MVF REST A1
	ROUT.AFFICH.INCR		ROUT.AFFICH.INCR
	ROUT.AFFICH.INCR		TEST NO 2
	WER		DER MOVE REST R2
	WER		RER
	TEST NO 3		TEST Q.CHARG NO.
	TEST NO 3		TEST Q.CHARG NO.
	RTN		DER MVF REST A2
	RTN		DER MOVE M=L6TH
	TEST NO 2		TEST DLA
	TEST NO 2		TEST DLA
	COMPUTE MASK RTNA15		TEST NO 2
	COMPUTE MASK RTNA15		TEST NO 3
	TEST DLA		FPP OP/S LD EXP
	TEST RB		FPP OP LD EXP
	ROUT.AFFICH.INCR		NOT USED
	MASK STACK PSW RTNA15		CF.15R2N
	TEST MEMOIRE NO1		ROUT.AFFICH.INCR
	TEST MEMOIRE NO1		TEST NO 3
	CF.15R2		MVB WRITE
	CF		MVUS WRITE
	TEST DLA		TEST Q.CHARG NO.
	CF		ROUT.AFFICH.INCR
	CF END		TEST NO 2
	EX		TEST NO 2
	TEST NO 3		TEST DLA
	TEST NO 3		TEST RB
	TEST MEMOIRE NO2		TEST NO 3
	TMP TEST		TEST NO 3
	TEST RB		TEST NO 1
	TEST RB		VERIF MAN. REG M
	MVF TEST LGHT=0		TRAP DLN
	EX TEST IF T2		DLN
	TEST MEMOIRE NO2		TEST NO 1
	TEST MEMOIRE NO2		TEST NO 1
	MVF LGHT-2 IN M		TEST NO 2
	MVF AL=SOURCE		TEST NO 2
	TEST TMP-TPM		ROUT.AFFICH.INCR
	MVF WRITE DEST		TEST NO 1
	MVF CLEAR LGHT		TEST NO 2
	TEST NO 3		TEST NO 2
	TEST NO 3		TEST RB
	TEST MEMOIRE NO1		TEST RB
	TEST MEMOIRE NO1		TEST NO 3
	TEST MEMOIRE NO1		TEST NO 3
	TEST MEMOIRE NO1		FPP STORE ST EXP
	TEST MEMOIRE NO1		FPP ST MANTI.
	MVF		TRAP DRN
	RER		DRN
	ROUT.AFFICH.INCR		TEST NO 1
	TEST NO 1		TEST NO 1
	MVB TEST LENGTH		TEST MEMOIRE NO2
	MVB PREPARE A2		TEST MEMOIRE NO2
	VERIF MAN. REG M		VERIF MAN REG L
	MVB PREPARE WRITE		CIO OTR
	INT		DER MVB READ
	RER		DER MVB WRITE
	TEST NO 3		DER MVF READ
	TEST NO 3		DER MVF WRITE
	TEST MEMOIRE NO1		TEST MEMOIRE NO1
	FPP DOUBLE LD 2ND WORD		TEST MEMOIRE NO1
	TEST RB		PAGE FAULT ML
	TEST RB		PAGE FAULT

Table 2-4B Control-ROM Microinstruction Listing P856

ADD.		ADD.		ADD.	
000	ROM /1E8,AR15,ALUB,CALU,MYC,QYC,,,WBUS,,,D	040	FTA AWRI,AANDB,CRLQG	080	ROM /008,ALUB,,,SYD,,,,,D
001	FTA ,0	041	ROM FLAG,/1C8,APSW,AANDB,CALU,MYC,,,,,BUSR,,D	081	ROM /100,,,,,D
002	ROM /1E9,,,,,SP2,PP2,,,BUSR,,D	042	ROM /0FD,,,CB10,MYC,,,PP2,,,SEGBUS,,,D	082	ROM /182,AWA2,DB10,,,,,RBUS,SEGB10,,,D
003	ROM FLAG,/0DD,AZ,ALUA,CALU,,,QYC,,,,,GAEXL,D	043	ROM /0F5,,,CB10,MYC,,,PP2,,,SEGBUS,,,D	083	ROM SNPLA,/100,ARR2,ALUA,CALU,MYC,QYC,,,,,D
004	ROM FLAG,/1F8,ASYS,TWOA,CALU,,,QYC,,,,,D	044	ROM /101,ARR1,AANDB,,,,,WMEM,,BUSR,,CRLQG	084	ROM /188,AWR1,TWOA,,,,,CTP1,,REPEAT,,,D
005	ROM /1FB,,,,,SM2,PM2,,,GFKYZD,D	045	ROM /101,ZERO,,,,,WMEM,,BUSR,,D	085	ROM /188,AWR1,ASHL,,,SLQ,,,CTP1,,REPEAT,,GCSEL,D
006	ROM /1FF,AEP,ALUA,CALU,MYC,,,,,D	046	FTA ARR1,AANDB,CRLQG	086	ROM /169,,,,,CTP1,,,D
007	ROM /1FF,,DB10,CB10,MYC,,,,,SEGBUS,,,D	047	ROM /1E9,,,,,BUSR,,CRFLQ	087	ROM /179,ARR2,ALUA,CALU,,,QYC,,,,,GCTLD,D
008	FTA AWRI,ALUB,CRLQG	048	ROM FLAG,/1A6,ARA1,DIVALU,,,,,GCRFNU,D	088	ROM FLAG,/0B3,ZERO,CALU,MYC,,,,,BUSR,GBTMF,D
009	ROM /1E9,ALUB,,,SYD,PYD,,,BUSR,,CRLQG	049	ROM FLAG,/140,ARA1,AXB,CALU,MYQ,QYC,,,,,D	089	ROM /0BA,AWA1,ZERO,CALU,,,QYC,,,,,GCRVZD,D
00A	ROM /16D,ASYS,ALUA,,,WBUS,,,D	04A	ROM /0E2,AEP,AMB,CALU,MYC,,,,,D	08A	ROM /130,ARA0,ALUA,,,WBUS,,BUSR,,D
00B	ROM /1FF,ARA0,ALUA,MYC,,,,,D	04B	ROM FLAG,/1B5,AWR1,TWOA,,,PP2,,,D	08B	ROM FLAG,/0DD,AZ,ALUA,CALU,,,QYC,,,,,GAEXL,D
00C	ROM /101,ARR1,ALUA,,,WMEM,,BUSR,,D	04C	ROM FLAG,/1B2,ARCT,ALUA,,,SP2,PM2,CTP1,WMEM,SEGBUS,,,D	08C	ROM FLAG,/0AB,ATEN,ACR,CALU,MYC,QYC,,,WEXM,,,GBOF,D
00D	ROM /101,AEP,ALUA,,,WMEM,,BUSR,,D	04D	ROM /1E9,ALUB,,,SYD,PYD,,,BUSR,,D	08D	ROM /1A8,,,,,SEGBUS,,GFRZD,D
00E	ROM /0E6,ATWO,ALUA,CALU,MYC,,,,,D	04E	ROM FLAG,/1A8,,,SLQ,,,GFRZD,D	08E	ROM FLAG,/104,AEP,ALUA,,,SP2,,,SEGBUS,,GCRFNU,D
00F	ROM /0E5,ATWO,ALUA,CALU,MYC,,,,,D	04F	ROM /172,,DB10,CB10,MYC,,,WMEM,SEGBUS,,GBCP,D	08F	ROM /159,,BSHR,CALU,MYC,SLQ,,,,,D
010	ROM FLAG,/1DD,ASYS,ALUA,CALU,,,QYC,,,,,D	050	FTA AWRI,A0B,CRLQG	090	ROM /1AA,ATEN,ACR,CALU,MYC,,,,,D
011	ROM /1E9,ALUB,,,SYD,PYD,,,BUSR,,D	051	ROM FLAG,/1A8,APSW,A0B,CALU,MYC,,,,,D	091	ROM SNPLA,/103,AEP,ALUA,,,SYD,,,,,D
012	ROM FLAG,/0E0,AWA0,ALUB,,,RBUS,,GBOK,D	052	ROM /0EC,,DB10,,,SYD,PP2,,,SEGBUS,BUSR,,D	092	ROM /1ED,AEP,ALUA,CALU,MYC,,,RBUS,SEGB10,,GFETCH,D
013	ROM /0E0,ARA0,ALUA,,,SYD,PYD,,,D	053	ROM /0E8,,DB10,,,SYD,PP2,,,SEGBUS,BUSR,,D	093	ROM SNPLA,/100,AEP,ALUA,CALU,MYC,QYC,,,,,D
014	ROM /108,ATWO,TWOA,CALU,MYC,,,,,D	054	ROM /101,ARR1,A0B,,,WMEM,,BUSR,,CRLQG	094	ROM /188,AWR1,ASHR,,,CTP1,,REPEAT,,GCRDSR,D
015	ROM /108,ATWO,TWOA,CALU,MYC,,,PM2,,,GFKYZD,D	055	ROM /0AB,AWA0,BSHR,,MYQ,,,,,D	095	ROM FLAG,/1C7,AWA0,ALUB,,,,,D
016	FTA ,0	056	ROM FLAG,/1A0,,,SLQ,,,,,D	096	ROM SNPLA,/100,ARCT,ALUA,CALU,MYC,,,,,D
017	ROM ETAT,/1FE,,,,,SEGBUS,,GIDLE,D	057	ROM FLAG,/190,ASYS,TWOA,CALU,,,QYC,,,,,GBCP,D	097	ROM /169,AEP,ALUA,CALU,,,QYC,,,CTP1,,,D
018	ROM FLAG,/193,AEP,ALUA,CALU,MYC,,,,,GCRVZD,D	058	ROM /1A6,AWA1,DIVALU,,,,,D	098	ROM FLAG,/0B3,ZERO,CALU,MYC,,,,,BUSR,GBTMF,D
019	ROM FLAG,/15B,AEP,ALUA,CALU,MYC,SLQ,,,,,GCRVZD,D	059	ROM FLAG,/140,ARA1,AXB,CALU,MYQ,QYC,,,,,D	099	ROM /0A5,ARA2,TWOA,CALU,,,QYC,,,,,D
01A	ROM FLAG,/17B,ARR1,ALUA,CALU,,,QYC,,,,,D	05A	ROM /0E2,AEP,AMB,CALU,MYC,,,,,D	09A	ROM /095,ARA1,ALUA,,,WBUS,,BUSR,GBTMF,D
01B	ROM /0CE,AWA1,ASHL,,,SLQ,,,CTP1,,REPEAT,,GCSEL,D	05B	ROM FLAG,/1A5,AWR1,ASHR,,,SRQ,PP2,,,GCRDSR,D	09B	ROM /0FE,AWR2,ALUB,,,,,D
01C	ROM FLAG,/16B,AEP,ALUA,CALU,MYC,,,,,D	05C	ROM FLAG,/1A2,ARCT,ALUA,,,SM2,PM2,CTP1,WMEM,SEGBUS,,,D	09C	ROM FLAG,/0B3,ATWO,ASHR,CALU,MYC,,,,,BUSR,GBTMF,D
01D	ROM FLAG,/14B,AEP,ALUA,CALU,MYC,SLQ,,,,,D	05D	ROM /0BE,,ALUB,,MYQ,,SYD,PYD,,,D	09D	ROM /143,AEP,ALUA,CLUR,MYC,,,SP2,,,WMEM,SEGBUS,BUSR,GCSEL,D
01E	ROM /0DD,ARR1,ALUA,CALU,,,QYC,,,,,D	05E	ROM /198,,,SLQ,,,,,D	09E	ROM /160,AEP,ALUA,CALU,MYC,,,,,GFLOT,D
01F	ROM /0CE,AWA1,ASHR,,,SRQ,,,CTP1,,REPEAT,,GCRDSR,D	05F	ROM /1FF,APSW,ALUA,CALU,MYC,,,,,D	09F	ROM /09B,,,,,RBUS,,GBOF,CRFLQ
020	FTA AWRI,APB,CRADD	060	FTA AWRI,AXB,CRIQG	0A0	ROM /105,ZERO,,,SYD,,,,,GFSYS,D
021	ROM /1E9,AEP,APB,,,SYD,PYD,,,BUSR,,CRADD	061	ROM /047,APSW,ALUA,CALU,MYC,,,SP2,,,GFSYS,D	0A1	ROM /1E9,AEP,APB,,,SYD,PYD,,,BUSR,,D
022	ROM /0EC,ARR2,ALUA,,,SYD,,,BUSR,,D	062	ROM /0FC,,CB10,MYC,,,SEGBUS,,D	0A2	ROM /0E7,ATWO,ALUA,CALU,MYC,,,,,D
023	ROM /0E8,ARR2,ALUA,,,SYD,,,BUSR,,D	063	ROM /0F4,,,CB10,MYC,,,SEGBUS,,D	0A3	ROM SNPLA,/100,ARR2,ALUA,CALU,,,QYC,SYD,,,,,D
024	ROM /101,ARR1,APB,,,WMEM,,BUSR,,CRADD	064	ROM /101,ARR1,AXB,,,WMEM,,BUSR,,CRLQG	0A4	ROM /0DB,AWA1,ASHL,,,SLQ,,,CTP1,,REPEAT,,,D
025	ROM /101,AZ,APLB1,,,WMEM,,BUSR,,CRADD	065	FTA AWA2,BSHR,,D	0A5	ROM FLAG,/01F,ATWO,ALUA,,,PYD,,,,,D
026	ROM /077,ATWO,TWOA,CALU,MYC,,,,,D	066	FTA ARR1,AXB,CRLQG	0A6	ROM /155,,BSHR,CALU,MYC,,,PM2,,,D
027	ROM /19E,AW15,AMB,,,SYD,,,GFSTOV,D	067	ROM /180,AWA0,ALUB,,MYQ,,,,,D	0A7	ROM /0CB,AEP,ALUA,CALU,,,QYC,,,,,D
028	ROM FLAG,/0DD,AZ,ALUA,CALU,,,QYC,,,,,GAEXL,D	068	ROM FLAG,/0DD,AZ,ALUA,CALU,,,QYC,,,,,GAEXL,D	0A8	ROM /08D,,,,,BUSR,GBEX,D
029	ROM /0E3,ATEN,TWOA,CALU,MYC,QYC,,,PYD,,,,,D	069	ROM /05F,ARR2,ALUA,CALU,,,QYC,,,PYD,,,,,D	0A9	ROM /08E,,CB10,,,QYC,SP2,,,SEGBUS,BUSR,,D
02A	ROM /159,ATEN,ACR,CALU,MYC,QYC,,,,,D	06A	ROM /0D6,AWA0,ALUB,,,PM2,,,D	0AA	ROM /14F,,BSHR,CALU,MYC,,,,,D
02B	ROM /145,AEP,ALUA,,,SYD,,,RBUS,,BUSR,GBOK,D	06B	ROM FLAG,/195,AWA1,ASHL,,,SLQ,PP2,,,D	0AB	ROM /152,,CB10,MYC,SP2,PP2,,,SEGBUS,,,D
02C	ROM FLAG,/1D2,AWCT,DB10,,,MYQ,SM2,PM2,CTP1,SEGBUS,,,D	06C	ROM /0E4,AWR1,TWOA,,,CTP1,,REPEAT,,,D	0AC	ROM /101,ARR1,ALUA,,,WMEM,,BUSR,GBEX,D
02D	ROM /0DC,,ALUB,,,SYD,PYD,,,BUSR,,D	06D	ROM FLAG,/1D7,AWA0,ALUB,,,,,D	0AD	ROM /094,AWA2,APB,CALU,MYQ,QYC,,,,,D
02E	ROM /134,,CLUR,MYC,,,,,GCSEL,D	06E	ROM /1FF,,DB10,CB10,MYC,,,SP2,PP2,,,SEGBUS,,,D	0AE	ROM /1E9,AEP,ALUA,,,SYD,,,SEGBUS,,,D
02F	ROM FLAG,/1CO,APUP,ALUA,CALU,,,QYC,,,,,D	06F	ROM /1FF,,DB10,CB10,MYC,,,SEGBUS,,,D	0AF	ROM FLAG,/150,,,SLQ,SP2,,,SEGBUS,BUSR,GBTMM,D
030	FTA AWRI,AMB,CRSUB	070	ROM SNPLA,/100,ARR15,,,,,GCTLD,D	0B0	ROM /140,,BSHR,CALU,MYC,,,,,D
031	ROM /1E9,AEP,AMB,,,SYD,PYD,,,BUSR,,CRSUB	071	ROM /18F,ARA2,ALUA,CALU,,,QYC,,,GFKYZD,D	0B1	ROM /1E9,AEP,AMB,,,SYD,PYD,,,BUSR,,D
032	ROM SNPLA,/100,,CB10,MYC,QYC,SP2,PP2,,,SEGBUS,,,D	072	ROM /1AD,,DB10,,,SYD,,,SEGBUS,BUSR,,D	0B2	ROM FLAG,/170,AIPL,APB,CALU,MYC,,,WMEM,,,D
033	ROM SNPLA,/100,,CB10,MYC,QYC,PP2,,,SEGBUS,,,D	073	ROM /1AC,,DB10,,,SYD,,,SEGBUS,BUSR,,D	0B3	ROM SNPLA,/100,,,PP2,,,D
034	ROM /101,ARR1,AMB,,,WMEM,,BUSR,,CRSUB	074	FTA ARR1,ALUA,CRLQG	0B4	ROM /0D2,AWA1,ASHR,,,SRQ,,,CTP1,,REPEAT,,GCRDSR,D
035	ROM FLAG,/0BC,AEN,ALUA,,,GCRFNU,D	075	ROM /0DC,ARA2,ALUA,,,BUSR,GCRFNU,D	0B5	ROM FLAG,/00F,ATWO,ALUA,,,PYD,,,,,D
036	ROM /077,ATWO,TWOA,CALU,MYC,,,,,D	076	ROM /092,AWA1,DSUB,,,BUSR,CRSUB	0B6	ROM /1E9,AEP,ALUA,,,SYD,,,SEGBUS,,D
037	FTA AWRI,A0B,D	077	ROM /1C5,ATWO,ALUA,CALU,MYC,,,,,D	0B7	ROM FLAG,/148,,,SLQ,SP2,,,WEXM,SEGBUS,BUSR,GBOM,D
038	ROM FLAG,/0DD,AZ,ALUA,CALU,,,QYC,,,,,GAEXL,D	078	ROM /0C7,AEP,ALUA,CALU,MYC,,,CTP1,,,D	0B8	ROM FLAG,/0DD,AZ,ALUA,CALU,,,QYC,,,,,GAEXL,D
039	ROM /0DE,ARR1,ALUA,CALU,,,QYC,,,,,D	079	ROM /150,ATWO,ASHR,CALU,,,QYC,,,GBTMM,D	0B9	ROM /08A,,CB10,,,QYC,SP2,,,SEGBUS,BUSR,,D
03A	ROM FLAG,/1C4,,BSHR,CALU,MYC,,,,,GCRFNU,D	07A	ROM /195,AWA1,ASHL,,,SLQ,,,,,D	0BA	ROM /1E9,AW15,DB10,,,,,SEGBUS,,,D
03B	ROM ETAT,/1FF,AEP,ALUA,,,SYD,,,,,D	07B	ROM FLAG,/185,AWA1,ASHR,,,SRQ,PP2,,,GCRDSR,D	0BB	ROM /142,,CB10,MYC,SP2,PP2,,,SEGBUS,,,D
03C	ROM FLAG,/1C2,AWCT,DB10,,,SP2,PM2,CTP1,SEGBUS,,GCRVML,D	07C	ROM /0C5,AEP,ALUA,CALU,MYC,,,CTP1,,,D	0BC	ROM /13D,,BSHR,,,SYD,,,SEGBUS,,,D
03D	ROM /0DC,,ALUB,,,SYD,PYD,,,BUSR,,D	07D	ROM /148,ATWO,ASHR,CALU,,,QYC,,,WEXM,,GBOM,D	0BD	ROM /08B,AWA2,AMB,CALU,MYQ,QYC,,,,,D
03E	ROM /15F,ATWO,ACR,,,PYD,,,,,D	07E	ROM /0C3,AEP,ALUA,CALU,MYC,,,CTP1,,,D	0BE	ROM /09E,AZ,APLB1,CALU,MYC,,,,,BUSR,GCRVML,CRADD
03F	ROM FLAG,/1BD,,,SLQ,,,RBUS,,GBOK,D	07F	ROM FLAG,/1F4,,BCR,CALU,,,QYC,,,GCRFNU	0BF	ROM /1E9,AWA2,ALUB,,,,,BUSR,GCRVML,CRLQG

Table 2-4B contd.

ADD.		ADD.	
180	ROM /028,,,MYQ,,,RBSU,,,GBOK.O	1C0	ROM /1E8,,,,,RBSU,,,GBOK.O
181	ROM /020,,,CBIO.MYC,,,SEQBUS,,,D	1C1	ROM /037.AWA1.AMB,,,,,D
182	ROM /050.ATEN.TWOA.CALU.MYC,,,,,D	1C2	ROM FLAG/02C,,BSHR,,MYC.QYC,,,,,6CRFNU.O
183	ROM /0E9,,,,,D	1C3	ROM /085.ATWO.ACR,,MYC.QYC,,,,,6FRZO.O
184	ROM /07A.ARR1.ALUA,,,,,WBUS,,BUSR.GBTME.O	1C4	ROM FLAG/0F1.AWR2.ALUB,,MYQ,,,,,D
185	ROM /188.ARR1.ALUA,,,,,WBUS.SEQBUS,,,D	1C5	ROM /188.AWR1.DBIO.CBIO,,,,,SEQBUS,,,CRRTN
186	ROM FLAG/054.ARI5.AMB,,,SRQ,,,,,6CRFNU.O	1C6	ROM /0CC,,,CBIO,,QYC,,,SEQBUS,,,D
187	ROM FLAG/064.ARCT.AMB,,,SLQ,,,CTP1,,,6CRFNU.O	1C7	ROM /029,,ALUB.CLUR.MYC,,,,,WDUS,,,D
188	ROM /076.AWR2.APB,,,SYD,,,BUSR,,D	1C8	ROM /036.AWA2.AMB,,,,,D
189	ROM SNPLA/100,,DBIO,,,SM2.PYD,,,SEQBUS.BUSR,,D	1C9	ROM /03B.ATWO.APB.CALU.MYC,,,,,D
18A	ROM /09F,,,,,D	1CA	ROM /09F,,,,,D
18B	ROM /018,,ZERO,,,SRQ,,PM2,,,,,D	1CB	ROM /071.ATWO.ASHR,,MYC,,,,,D
18C	ROM /072.ATWO.APB.CALU.MYC,,,,,6FENB.O	1CC	ROM /07F,,BINU,,QYC,,,,,D
18D	ROM /06E.AWA0.BINU.MYQ,,,,,D	1CD	ROM /09B.ATWO.ASHR.MYC.QYC,,,,,6FRZO.O
18E	ROM FLAG/024.ARA1.AXB,,,,,6CRFNU.O	1CE	ROM /0AC.AWA0.ALUB,,,,,SEQBUS,,6FLOT.O
18F	ROM /061.ATWO.APLB1.MYC,,,,,D	1CF	ROM /18B.AEP.ALUA,,,SYD,,,SEQBUS,,6FLOT.O
190	ROM /019,,BINU,,MYC,,,,,D	1D0	ROM /02F,,,,,D
191	ROM /18E.ARA0.AOB.CLUR.MYC.QYC,,,,,D	1D1	ROM /06A.AWR1.AMB,,,SYD,,,,,D
192	ROM /09F,,,,,D	1D2	ROM FLAG/07C,,BSHR,,MYC.SRQ,,,,,6CRFNU.O
193	ROM FLAG/002,,,CBIO.MYC,,,SEQBUS,,,D	1D3	ROM /032.ARA0,,,,,GCTLD.O
194	ROM /06A.AWR1.AMB,,,SYD,,,GFSTOV.O	1D4	ROM FLAG/0EA.AWA1.APB,,,SYD,,,SEQBUS,,GM0VE.O
195	ROM /068.APSW.ALUA,,,SP2,,,WMEM,,BUSR,,D	1D5	ROM FLAG/0EA.AWA1.APB,,,SYD,,,SEQBUS,,GBEX.O
196	ROM /0E9,,,,,D	1D6	ROM ETAT/09E,,,,,SEQBUS,,,D
197	ROM /067.AEP.ALUA,,MYQ,,SP2,,,WMEM.SEQBUS.BUSR,,D	1D7	ROM FLAG/008,,ALUB.CBIO,,QYC,,,WBUS.SEQBUS,,,D
198	ROM /1E9,,ALUB,,,SYD.PYD,,,SEQBUS.BUSR,,D	1D8	ROM FLAG/084.ATEN.AXB,,,,,6CRFNU.O
199	ROM /05E,,BSHR.CALU.MYC,,,,,D	1D9	ROM /017.AEP.AMB,,,PYD,,,,,D
19A	ROM /09F,,,,,D	1DA	ROM /09F,,,,,D
19B	ROM FLAG/078,,,MYQ,,,,,D	1DB	ROM /070.ATEN.TWOA.MYC,,,,,D
19C	ROM FLAG/05C.AEP.AXB,,,SM2.PM2,,,6CRFNU.O	1DC	ROM FLAG/098,,,MYQ,,,,,D
19D	ROM /0FB,,,,,RBSU,,,GBOK.O	1DD	ROM /078.ATWO.ASHR.MYC.QYC,,,,,D
19E	ROM /060.AWA0.BCR.MYC,,,,,D	1DE	ROM FLAG/00C,,BINU,,,,,D
19F	ROM /051.AEP.APB.MYC,,,PYD,,WBUS,,,D	1DF	ROM /049,,ALUB,,,,,WBUS,,,D
1A0	ROM FLAG/13A.AWA0.ALUB,,MYQ,,PM2,,,,,D	1E0	ROM /100,,ALUB,,,PYD,,,,,D
1A1	ROM FLAG/174.ATEN.AXB,,,,,6CRFNU.O	1E1	ROM FLAG/195.ARA1.ALUA,,,,,D
1A2	ROM /09F,,,,,D	1E2	ROM /09F,,,,,D
1A3	ROM FLAG/062,,,CBIO.MYC,,,SEQBUS,,,D	1E3	ROM /021,,BINU,,MYC,,,,,D
1A4	ROM /05A.ARR2.AMB.CALU.MYC,,,,,D	1E4	ROM FLAG/01A,,,,,PM2,,,,,D
1A5	ROM /12B.AWA1.APB.MYQ.SYD,,,BUSR,,D	1E5	ROM /033,,,MYQ,,,,,D
1A6	ROM /096.ATWO.ACR.CALU.MYC,,,WBUS.SEQBIO,,GFKYZO.CRIO TMP T	1E6	ROM FLAG/04C,,BSHR,,MYC,,,,,6CRFNU.O
1A7	ROM FLAG/058,,,SLQ,,,WBUS,CBTMP.O	1E7	ROM /04C,,,MYQ,,,,,D
1A8	ROM FLAG/12A.AWA1.AMB,,,SYD,,,SEQBUS,,GM0VE.O	1E8	ROM /016.AEP.APB.CALU.MYC.SRQ,,,,,D
1A9	ROM /13A.AWR2.ZERO,,,,,D	1E9	ROM FLAG/074.ATEN.AXB,,,,,6CRFNU.O
1AA	ROM /09F,,,,,D	1EA	ROM /09F,,,,,D
1AB	ROM /013.AWA2.FORA,,,,,D	1EB	ROM /033.ATEN.ASHR.CALU.MYC,,,,,D
1AC	ROM FLAG/052,,ALUB,,,SM2.PM2,,WMEM.SEQBUS,,,D	1EC	ROM /012.ATEN.ALUA.MYC.QYC,,,,,D
1AD	ROM /080.ARA0.ALUA,,,SYD.PYD,,,,,D	1ED	ROM FLAG/044.ARA2.AMB,,,SRQ,,,,,6CRFNU.O
1AE	ROM SNPLA/1FF.AWA1.ALUB.CIOR.MYC.QYC,,PM2,,,SEQBIO,,6FETCH.O	1EE	ROM /1E9.AEP.ALUA,,,SYD,,,SEQBUS,,CRFLO
1AF	ROM /0B1.AWA0.BCR,,,SYD.PYD,,,6FRZO.O	1EF	ROM FLAG/010,,,SLQ,SP2,,,WEXM.SEQBUS.BUSR.GBOF.O
1B0	ROM /05B.ATWO.ALUA.CALU.MYC.QYC,,,,,D	1F0	ROM /100,,ALUB,,,PYD,,,,,D
1B1	ROM /046.AWA0.BSHR.MYQ,,,,,D	1F1	ROM FLAG/185.AWA1.ASHR,,,SRQ,,,,,6CRDSR.U
1B2	ROM FLAG/03C,,BSHR,,MYC,,,,,6CRFNU.O	1F2	ROM /09F,,,,,D
1B3	ROM FLAG/0B8.ATWO.AOB.MYC.QYC,,,,,6FRZO.O	1F3	ROM /07F,,,,,D
1B4	ROM FLAG/0F2.AWA0.ALUB,,,PM2,,,,,D	1F4	ROM /0C1,,,SM2.PM2,,,SEQBUS,,,D
1B5	ROM FLAG/10B.AWA2.AMB,,,,,D	1F5	ROM /063,,ALUB,,,SYD.PYD,,,,,D
1B6	ROM ETAT/0BE,,,,,SEQBUS,,,D	1F6	ROM /097,,DBIO,,,WBUS.SEQBUS,,,D
1B7	ROM FLAG/02B,,ALUB,,MYQ,,PM2,,WMEM,,BUSR,,D	1F7	ROM /0BB.ARR1.ALUA.CALU.MYC,,,WBUS,,BUSR.GBTMP.O
1B8	ROM /162,,ALUB,,,,,WMEM,,BUSR,,D	1F8	ROM /036.AEP.ALUA.CALU.MYC,,,,,GCTLD.O
1B9	ROM /03A.ARA0.AOB,,,SYD,,,RBSU,,BUSR.GBTME.O	1F9	ROM /007.AWA1.AMB,,,PP2,,,GCTLD.O
1BA	ROM /09F,,,,,D	1FA	ROM /03E.AEP.ALUA.CALU.MYC,,,,,GCTLD.O
1BB	ROM /01A,,ZERO,,,SRQ,,,,,D	1FB	ROM /005.AWA1.APB,,,PP2,,,GCTLD.O
1BC	ROM /058.ATEN.ALUA.CALU.QYC,,,WBUS.SEQBIO,,6FETCH.O	1FC	ROM FLAG/06C.ARA1.AXB,,,SM2.PM2,,,6CRFNU.O
1BD	ROM /0B3,,,SP2,,,SEQBUS.BUSR.GBTMP.O	1FD	ROM FLAG/0C0.ARA0.ALUA.CALU.MYC,,,SYD.PYD,,,,,D
1BE	ROM /040,,BCR.MYC,,,,,GFKYZO.O	1FE	ROM /120.ATWO.TWOA.CALU.MYC.QYC,,,,,GCTLD.O
1BF	ROM FLAG/014.AEP.AXB.MYQ,,,,,6CRFNU.O	1FF	ROM /125.ATWO.TWOA.CALU.QYC,,,,,GCTLD.O

Table 2-5A Control-ROM Binary Content P857

ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	6P	CR	ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	6P	CR	ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	6P	CR			
000	00	111101000	00011	00001	00	10	11	00	00	0	111	00	0	00000	00000	040	11	111111111	01100	01000	11	10	11	10	11	0	000	10	0	11100	10000	080	00	000001000	00000	00001	00	00	00	01	00	0	000	00	0	00000	00000			
001	11	111111111	00000	00000	11	10	11	10	11	0	000	10	0	11100	00000	041	01	111001000	10110	01000	00	10	00	00	00	0	000	10	0	10000	00000	081	00	100000000	00000	00000	00	00	00	00	00	0	000	00	0	00000	00000			
002	00	111101001	00000	00000	00	00	00	10	11	0	000	00	1	00000	00000	042	00	011111101	00000	00000	01	10	00	00	11	0	000	10	0	00000	00000	082	00	110110010	01010	01110	00	00	00	00	00	0	000	00	0	100	11	0	00000	00000
003	01	011010000	10000	00011	00	00	11	00	00	0	000	00	0	11011	00000	043	00	011110101	00000	00000	01	10	00	00	11	0	000	10	0	00000	00000	083	10	100000000	00101	00011	00	10	11	00	00	0	000	00	0	00000	00000			
004	01	111110000	11110	10010	00	00	11	00	00	0	000	00	0	00000	00000	044	00	100000001	00100	01000	00	00	00	00	00	0	011	00	1	00000	10000	084	00	110001011	01100	10010	00	00	00	00	00	0	1	000	01	0	00000	00000		
005	00	111110101	00000	00000	00	00	00	11	10	0	000	00	0	01110	00000	045	00	100000001	00000	01010	00	00	00	00	00	0	011	00	1	00000	00000	085	00	110001011	01100	10001	00	00	01	00	00	0	1	000	01	0	00110	00000		
006	00	111111111	10100	00011	00	10	00	00	00	0	000	00	0	00000	00000	046	11	111111111	00100	01000	11	10	11	10	11	0	000	10	0	11100	10000	086	00	101101001	00000	00000	00	00	00	00	00	0	1	000	00	0	00000	00000		
007	00	111111111	00000	01110	01	10	00	00	00	0	000	10	0	00000	00000	047	00	111101001	00000	00000	00	00	00	00	00	0	000	00	1	00000	01100	087	00	101111001	00101	00011	00	00	11	00	00	0	000	00	0	00010	00000			
008	11	111111111	01100	00001	11	10	11	10	11	0	000	10	0	11100	10000	048	01	110100110	00001	11100	00	00	00	00	00	0	000	00	0	00100	00000	088	01	010110011	00000	01010	00	10	00	00	00	0	000	00	1	11001	00000			
009	00	111101001	00000	00001	00	00	00	01	01	0	000	00	1	00000	10000	049	01	101000000	00001	00111	00	11	11	00	00	0	000	00	0	00000	00000	089	00	010111010	01001	01010	00	00	11	00	00	0	000	00	0	01111	00000			
00A	00	101101101	11110	00011	00	00	00	00	0	111	00	0	0	00000	00000	04A	00	011100010	10100	00000	00	10	00	00	00	0	000	00	0	00000	00000	08A	00	100110000	00000	00011	00	00	00	00	0	111	00	1	00000	00000				
00B	00	111111111	00000	00011	00	10	00	00	00	0	000	00	0	00000	00000	04B	01	110110101	01100	10010	00	00	00	11	0	000	00	0	00000	00000	08B	01	011010000	10000	00011	00	00	11	00	00	0	000	00	0	11011	00000				
00C	00	100000001	00100	00011	00	00	00	00	0	011	00	1	00000	00000	04C	01	110110010	00111	00011	00	00	10	10	1	011	10	0	00000	00000	08C	01	010101011	11010	01011	00	10	11	00	00	0	001	00	0	10100	00000					
00D	00	100000001	10100	00011	00	00	00	00	0	011	00	1	00000	00000	04D	00	111101001	00000	00001	00	00	00	01	01	0	000	00	1	00000	00000	08D	00	110101000	00000	00000	00	00	00	00	00	0	000	10	0	01101	00000				
00E	00	011100110	11011	00011	00	10	00	00	0	000	00	0	00000	00000	04E	01	110101000	00000	00000	00	00	01	00	00	0	000	00	0	01101	00000	08E	01	110101000	11000	00011	00	00	10	00	0	000	10	0	00100	00000					
00F	00	011100101	11011	00011	00	10	00	00	0	000	00	0	00000	00000	04F	00	1011110010	00000	01110	01	10	00	00	0	011	10	0	10001	00000	08F	00	101011001	00000	01101	00	10	01	00	00	0	000	00	0	00000	00000					
010	01	111010000	11110	00011	00	00	11	00	00	0	000	00	0	00000	00000	050	11	111111111	01100	00010	11	10	11	10	11	0	000	10	0	11100	10000	090	00	110101010	11010	01011	00	10	00	00	00	0	000	00	0	00000	00000			
011	00	111101001	00000	00001	00	00	00	01	01	0	000	00	1	00000	00000	051	01	110001000	10110	00010	00	10	00	00	00	0	000	00	0	00000	00000	091	10	100000011	10100	00011	00	00	01	00	0	000	00	0	00000	00000				
012	01	011100000	01000	00001	00	00	00	00	0	100	00	0	10011	00000	052	00	011101100	00000	01110	00	00	00	01	11	0	000	10	1	00000	00000	092	00	111101101	10100	00011	00	10	00	00	00	0	100	11	0	11100	00000				
013	00	011100000	00000	00011	00	00	00	01	01	0	000	00	0	00000	00000	053	00	011101000	00000	01110	00	00	00	00	00	0	000	10	1	00000	00000	093	10	100000000	10100	00011	00	10	11	00	00	0	000	00	0	00000	00000			
014	00	111011000	11011	10010	00	10	00	00	00	0	000	00	0	00000	00000	054	00	100000001	00100	00010	00	00	00	00	00	0	011	00	1	00000	10000	094	00	110001011	01100	01111	00	00	00	00	00	0	1	000	01	0	00011	00000		
015	00	111011000	11011	10010	00	10	00	00	00	0	000	00	0	01110	00000	055	00	010101000	01000	01101	00	11	00	00	00	0	000	00	0	00000	00000	095	01	111000111	01000	00001	00	00	00	00	00	0	000	00	0	00000	00000			
016	11	111111111	00000	00000	11	10	11	10	11	0	000	10	0	11100	00000	056	01	110100000	00000	00000	00	00	01	00	00	0	000	00	0	00000	00000	096	10	100000000	00111	00011	00	10	00	00	00	0	000	00	0	00000	00000			
017	11	111111110	00000	00000	00	00	00	00	00	0	000	10	0	00001	00000	057	01	110010000	11110	10010	00	00	11	00	00	0	000	00	0	10001	00000	097	00	101101001	10100	00011	00	00	11	00	00	0	1	000	00	0	00000	00000		
018	01	110010011	10100	00011	00	10	00	00	0	000	00	0	01111	00000	058	00	110100110	01001	11100	00	00	00	00	00	0	000	00	0	00000	00000	098	01	010110011	00000	01010	00	10	00	00	00	0	000	00	1	11001	00000				
019	01	101010101	10100	00011	00	10	01	00	00	0	000	00	0	01111	00000	059	01	101000000	00001	00111	00	11	11	00	00	0	000	00	0	00000	00000	099	00	010100101	00010	10010	00	00	11	00	00	0	000	00	0	00000	00000			
01A	01	101110101	00100	00011	00	00	11	00	00	0	000	00	0	00000	00000	05A	00	011100010	10100	00000	00	10	00	00	00	0	000	00	0	00000	00000	09A	00	010010101	00001	00011	00	00	00	00	00	0	111	00	1	11001	00000			
01B	00	011001110	01001	10001	00	00	01	00	00	1	000	01	0	00110	00000	05B	01	110100101	01100	01111	00	00	10	00	11	0	000	00	0	00011	00000	09B	00	011111110	01101	00001	00	00	00	00	00	0	000	00	0	00000	00000			
01C	01	101101011	10100	00011	00	10	00	00	0	000	00	0	00000	00000	05C	01	110100010	00111	00011	00	00	11	10	1	011	10	0	00000	00000	09C	01	010110011	11011	01111	00	10	00	00	00	0	000	00	1	11001	00000					
01D	01	101001011	10100	00011	00	10	01	00	00	0	000	00	0	00000	00000	05D	00	010111110	00000	00001	00	11	00	01	0	000	00	0	00000	00000	09D	00	101000011	10100	00011	10	10	00	10	00	0	011	10	1	00110	00000				
01E	00	011011101	00100	00011	00	00	11	0																																										

Table 2-5A contd.

ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR	ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR	ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR
0C0	11	11111111	01100	00001	11	10	11	10	11	0	000	10	0	11100	00000	100	00	00011111	00000	00000	00	00	00	00	0	000	00	0	1001	00000	140	00	00111111	00000	00000	00	00	00	00	0	100	00	0	10011	00000		
0C1	00	11110100	00000	00001	00	00	00	01	01	0	000	00	1	00000	00000	101	00	11001010	00000	00011	00	11	00	00	01	0	000	00	1	00000	00000	141	00	11110100	01101	00000	00	00	00	00	0	000	00	1	01011	00000	
0C2	00	10011010	00000	01001	00	00	00	00	00	0	000	00	1	01100	00000	102	00	01110110	00101	00110	00	00	01	00	0	000	00	1	00000	00000	142	00	11100111	01100	01010	00	00	00	00	0	000	00	1	00000	00000		
0C3	00	01110110	00000	00000	01	10	00	00	11	0	000	10	0	00000	00000	103	00	11010110	00101	00110	00	00	01	00	0	000	00	1	00000	00000	143	00	10000001	10000	00000	00	00	00	00	0	011	00	1	00000	11000		
0C4	00	00100111	01010	00110	00	00	00	00	00	0	000	00	0	00000	00000	104	00	01101001	00000	00000	01	10	00	00	0	100	11	0	00000	00000	144	00	11000100	00000	00001	00	00	00	00	0	111	11	0	00000	01000		
0C5	00	11110100	00000	00011	00	00	00	01	01	0	000	00	1	00000	00000	105	01	01101010	00000	00011	00	00	00	00	0	000	00	0	00100	00000	145	00	01011011	00010	00011	00	00	11	00	0	000	00	0	0000	00000		
0C6	00	01000100	00100	00011	00	00	11	00	00	0	000	00	0	00000	00000	106	01	01101000	10000	00011	00	00	11	00	0	000	00	0	11011	00000	146	01	00001100	11011	00110	00	10	00	00	0	000	00	0	00100	00000		
0C7	11	11111111	00100	00000	11	10	11	10	11	0	000	10	0	11100	11100	107	00	001100110	00000	01101	00	10	00	00	0	000	00	0	00000	00000	147	00	00001100	00000	00000	00	01	00	00	0	000	00	0	00000	00000		
0C8	00	010000110	00100	01011	10	10	00	00	00	0	000	00	1	10000	00000	108	01	011010000	10000	00011	00	00	11	00	0	000	00	0	11011	00000	148	00	01011010	01001	10100	00	10	00	00	1	000	01	0	00000	00000		
0C9	11	11111111	01100	01001	11	10	11	10	11	0	000	10	0	11100	00000	109	00	001001011	00101	00011	00	00	00	01	0	000	00	0	00000	00000	149	00	01101011	01001	01000	00	11	00	00	0	000	00	0	11101	00000		
0CA	00	11110100	00000	01110	00	00	00	01	01	0	000	10	1	00000	00000	10A	00	011101000	00101	00110	00	00	01	00	0	000	00	1	00000	00000	14A	00	01011010	01010	01101	00	00	00	00	0	000	00	0	00000	00000		
0CB	00	10011010	00000	00000	00	00	00	00	00	0	000	00	0	01001	00000	10B	00	110110100	00101	00110	00	00	01	00	0	000	00	1	00000	00000	14B	00	01101011	00010	00011	00	00	00	00	0	000	00	0	00100	00000		
0CC	00	100000001	00100	00011	00	00	00	00	00	0	011	00	1	10000	00000	10C	00	001001010	11011	00011	00	10	11	00	0	000	00	0	00000	00000	14C	00	010100110	00000	00001	00	10	10	00	0	000	10	1	11001	00000		
0CD	00	10011101	00000	01101	00	00	00	01	00	0	000	00	0	01110	00000	10D	00	111101001	00000	00011	00	00	01	01	0	000	00	1	00000	00000	14D	00	001000010	00000	00000	00	00	10	00	0	000	10	1	11001	00000		
0CE	01	10010100	00000	00110	10	10	01	00	00	0	000	00	0	00000	00000	10E	00	111101011	00000	00000	00	00	00	01	0	000	00	0	00000	00000	14E	00	001010011	11011	01111	00	10	11	11	10	0	011	00	0	00000	00000	
0CF	00	01110110	00000	00000	11	10	11	00	00	0	111	11	0	11100	00000	10F	00	100100101	11011	10010	00	00	11	00	0	000	00	0	00000	00000	14F	00	001101100	01001	00001	00	00	11	10	0	000	00	0	00000	00000		
0D0	01	011010000	10000	00011	00	00	11	00	00	0	000	00	0	11011	00000	110	00	011011111	00000	00000	00	00	00	0	000	00	0	01101	00000	150	00	010011111	00000	00000	00	00	00	00	0	000	00	0	01101	00000			
0D1	01	011010000	10000	00011	00	00	11	00	00	0	000	00	0	11011	00000	111	10	100000000	00101	00110	00	00	11	01	00	0	000	00	0	00000	00000	151	00	010100001	01001	01001	00	10	11	00	0	000	00	0	00000	00000	
0D2	00	100100101	01101	00001	00	00	00	00	00	0	000	00	0	00000	00000	112	00	100101100	00101	00110	00	00	01	00	0	000	00	1	00000	00000	152	01	000110000	00000	00000	00	11	00	10	0	000	10	1	11001	00000		
0D3	10	100000000	00000	01110	01	00	11	01	11	0	000	10	0	00000	00000	113	10	100000000	10100	00011	01	10	11	01	00	0	000	10	0	00000	00000	153	00	101110011	00000	10010	00	00	01	00	0	000	00	0	00000	00000	
0D4	01	100011011	01010	00000	01	10	00	01	00	0	000	10	0	00111	00000	114	00	001001000	01010	00110	01	10	00	01	00	0	000	10	0	00111	00000	154	00	000010000	00000	00101	00	00	11	10	00	0	001	10	1	10100	00000
0D5	00	001010110	01001	00110	00	00	00	00	00	0	000	00	0	00000	00000	115	00	001010110	01010	00110	00	00	00	00	0	000	00	0	00000	00000	155	00	000010000	00000	00000	00	00	10	00	0	001	10	1	10100	00000		
0D6	01	100100000	00000	00000	00	00	01	00	00	0	000	00	0	00000	00000	116	00	010101110	11010	00011	00	10	00	00	0	000	00	0	01101	00000	156	00	011101001	00000	00000	00	00	00	00	0	000	00	0	00000	00000		
0D7	01	011111000	00000	01001	00	00	11	00	00	0	000	00	0	00000	00000	117	10	100000000	00000	00000	01	10	11	00	0	000	10	0	00000	00000	157	00	010100111	00000	00010	00	00	01	00	0	100	00	1	10111	00000		
0D8	11	11111111	00100	00000	11	10	11	10	11	0	000	10	0	11100	11100	118	00	011101100	01101	00110	00	00	01	00	0	000	00	1	00000	00000	158	00	010100110	01100	01110	11	10	00	00	0	000	10	0	00110	01000		
0D9	11	11111111	01000	00000	11	10	11	10	11	0	000	10	0	11100	11100	119	00	111110011	01101	00000	00	00	00	00	0	000	00	0	01011	00000	159	00	110001000	01100	00010	00	00	01	00	0	000	00	0	00000	00000		
0DA	00	100100100	10110	00011	00	10	00	00	00	0	000	00	0	01001	00000	11A	00	111110010	01101	00000	00	00	00	00	0	000	00	0	01011	00000	15A	00	010100100	01001	00011	00	00	00	00	0	000	00	0	01111	00000		
0DB	00	100100010	00011	00011	00	00	01	00	00	0	001	00	1	10101	00000	11B	00	111101001	00100	00011	00	00	00	00	0	000	00	1	00101	10000	15B	00	010100011	01001	11110	00	00	01	00	1	000	00	0	00011	00000		
0DC	00	010001000	00100	00011	10	10	00	00	00	0	000	00	1	10000	00000	11C	01	111101010	00100	00011	00	00	00	00	0	000	00	0	00000	00000	15C	00	010100010	01001	11110	00	00	01	00	1	000	01	0	00000	00000		
0DD	00	100011101	00000	00001	00	11	00	11	00	0	011	10	1	00000	00000	11D	00	100111010	01101	01101	00	00	00	00	0	000	00	0	00000	00000	15D	01	110110110	01001	11100	00	00	01	00	0	000	00	0	00100	00000		
0DE	01	100011000	00000	00000	00	00	01	00	00	0	000	00	0	00000	00000	11E	00	111101100	01101	01110	00	00	00	00	0	100	11	0	00000	00000	15E	00	010100000	11011	00110	10	10	00	00	1	000	00	0	00000			

Table 2-5A contd.

ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR	ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR
180	00	000101000	00000	00000	00	11	00	00	00	0	100	00	0	10011	00000	1C0	00	111101000	00000	00000	00	00	00	00	0	100	00	0	10011	00000	
181	00	000100000	00000	00000	01	10	00	00	00	0	000	10	0	00000	00000	1C1	00	000110111	01001	00000	00	00	00	00	0	000	00	0	00000	00000	
182	00	001010000	00000	00000	00	11	00	00	00	0	000	00	0	00000	00000	1C2	01	000101100	00000	01101	00	10	11	00	0	000	00	0	00100	00000	
183	00	011101001	00000	00000	00	00	00	00	00	0	000	00	0	00000	00000	1C3	00	010000101	11011	01011	00	10	11	00	0	000	00	0	01101	00000	
184	00	001111010	00100	00011	00	00	00	00	00	0	111	00	1	10110	00000	1C4	01	011110001	01101	00001	00	11	00	00	0	000	00	0	00000	00000	
185	00	110001000	00100	00011	00	00	00	00	00	0	111	10	0	00000	00000	1C5	00	110001000	01100	01110	01	00	00	00	0	000	10	0	00000	00100	
186	01	001010100	00011	00000	00	00	10	00	00	0	000	00	0	00100	00000	1C6	00	011001100	00000	00000	01	00	11	00	0	000	10	0	00000	00000	
187	01	001100100	00111	00000	00	00	01	00	00	1	000	00	0	00100	00000	1C7	00	000101001	00000	00001	10	10	00	00	0	111	00	0	00000	00000	
188	00	001110110	01101	00110	00	00	01	00	0	000	00	1	00000	00000	1C8	00	000110110	01010	00000	00	00	00	00	0	000	00	0	00000	00000		
189	10	100000000	00000	01110	00	00	00	11	01	0	000	10	1	00000	00000	1C9	00	000111011	11011	00110	00	10	00	00	0	000	00	0	00000	00000	
18A	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000	1CA	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000		
18B	00	000011011	00000	01010	00	00	10	00	10	0	000	00	0	00000	00000	1CB	00	001110001	11011	01111	00	10	00	00	0	000	00	0	00000	00000	
18C	00	001110010	11011	00110	00	10	00	00	0	000	00	0	01010	00000	1CC	00	001111111	00000	00101	00	00	11	00	0	000	00	0	00000	00000		
18D	00	001101110	01000	00101	00	11	00	00	0	000	00	0	00000	00000	1CD	00	010011000	11011	01111	00	10	11	00	0	000	00	0	01101	00000		
18E	01	000100100	00001	00111	00	00	00	00	0	000	00	0	00100	00000	1CE	00	010101100	01000	00001	00	00	00	00	0	000	10	0	11010	00000		
18F	00	001100001	11011	00100	00	10	00	00	0	000	00	0	00000	00000	1CF	00	110110000	10100	00011	00	00	00	01	00	0	000	10	0	11010	00000	
190	00	000011001	00000	00101	00	10	00	00	0	000	00	0	00000	00000	1D0	00	000101111	00000	00000	00	00	00	00	0	000	00	0	00000	00000		
191	00	110111110	00000	00010	10	10	11	00	00	0	000	00	0	00000	00000	1D1	00	001101010	01100	00000	00	00	00	01	00	0	000	00	0	00000	00000
192	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000	1D2	01	001111100	00000	01101	00	10	10	00	00	0	000	00	0	00100	00000	
193	01	000000010	00000	00000	01	10	00	00	0	000	10	0	00000	00000	1D3	00	000110010	00000	00000	00	00	00	00	0	000	00	0	00010	00000		
194	00	001101010	01100	00000	00	00	01	00	0	000	00	0	01011	00000	1D4	01	011101010	01001	00110	00	00	00	01	00	0	000	10	0	00111	00000	
195	00	001101000	10110	00011	00	00	10	00	0	011	00	1	00000	00000	1D5	01	011101010	01001	00110	00	00	00	01	00	0	000	10	0	10010	00000	
196	00	011101001	00000	00000	00	00	00	00	0	000	00	0	00000	00000	1D6	11	010011110	00000	00000	00	00	00	00	0	000	10	0	00000	00000		
197	00	001100111	10100	00011	00	11	00	10	00	0	011	10	1	00000	00000	1D7	01	011011000	00000	00001	01	00	11	00	0	111	10	0	00000	00000	
198	00	111101001	00000	00001	00	00	01	01	0	000	10	1	00000	00000	1D8	01	010000100	11010	00111	00	00	00	00	0	000	00	0	00100	00000		
199	00	001011110	00000	01101	00	10	00	00	0	000	00	0	00000	00000	1D9	00	000010111	10100	00000	00	00	00	01	0	000	00	0	00000	00000		
19A	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000	1DA	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000		
19B	01	001111000	00000	00000	00	11	00	00	0	000	00	0	00000	00000	1DB	00	001110000	11010	10010	00	10	00	00	0	000	00	0	00000	00000		
19C	01	001011100	10100	00111	00	00	11	10	0	000	00	0	00100	00000	1DC	01	010011000	00000	00000	00	11	00	00	0	000	00	0	00000	00000		
19D	00	011111011	00000	00000	00	00	00	00	0	100	00	0	10011	00000	1DD	00	001111000	11011	01111	00	10	11	00	00	0	000	00	0	00000	00000	
19E	00	001100000	01000	01001	00	10	00	00	0	000	00	0	00000	00000	1DE	01	000001100	00000	00101	00	00	00	00	0	000	00	0	00000	00000		
19F	00	001010001	10100	00110	00	10	00	00	0	111	00	0	00000	00000	1DF	00	001001001	00000	00001	00	00	00	00	0	111	00	0	00000	00000		
1A0	01	100111010	01000	00001	00	11	00	00	10	0	000	00	0	00000	00000	1E0	00	100000000	00000	00001	00	00	00	00	01	0	000	00	0	00000	00000
1A1	01	101110100	11010	00111	00	00	00	00	0	000	00	0	00100	00000	1E1	01	110010101	00001	00011	00	00	00	00	00	0	000	00	0	00000	00000	
1A2	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000	1E2	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000		
1A3	01	001100010	00000	00000	01	10	00	00	0	000	10	0	00000	00000	1E3	00	000100001	00000	00101	00	10	00	00	0	000	00	0	00000	00000		
1A4	00	001010101	00101	00000	00	10	00	00	0	000	00	0	00000	00000	1E4	01	000011010	00000	00000	00	00	00	10	0	000	00	0	00000	00000		
1A5	00	100101011	01001	00110	00	11	00	01	00	0	000	00	1	00000	00000	1E5	00	000110011	00000	00000	00	11	00	00	0	000	00	0	00000	00000	
1A6	00	010010110	11011	01011	00	10	00	00	0	111	11	0	01110	01000	1E6	01	001001100	00000	01110	00	10	00	00	0	000	00	0	00100	00000		
1A7	01	001011000	00000	00000	00	00	01	00	0	111	00	0	10111	00000	1E7	00	001001100	00000	00000	00	11	00	00	0	000	00	0	00000	00000		
1A8	01	100101010	01001	00000	00	00	01	00	0	000	10	0	00111	00000	1E8	00	000010110	10100	00110	00	10	10	00	00	0	000	00	0	00000	00000	
1A9	00	100111010	01101	01010	00	00	00	00	0	000	00	0	00000	00000	1E9	01	001110100	11010	00111	00	00	00	00	0	000	00	0	00100	00000		
1AA	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000	1EA	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000		
1AB	00	000010011	01010	10000	00	00	00	00	0	000	00	0	00000	00000	1EB	00	000110011	11010	01111	00	10	00	00	0	000	00	0	00000	00000		
1AC	01	001010010	00000	00001	00	00	11	10	0	011	10	0	00000	00000	1EC	00	000010010	11010	00111	00	10	11	00	00	0	000	00	0	00000	00000	
1AD	00	010110000	00000	00011	00	00	00	01	01	0	000	00	0	00000	00000	1ED	01	001000100	00010	00000	00	00	10	00	0	000	00	0	00100	00000	
1AE	10	111111111	01001	00001	11	10	11	00	10	0	000	11	0	11100	00000	1EE	00	111101001	10100	00011	00	00	00	01	00	0	000	10	0	00000	01100
1AF	00	0																													

Table 2-5B Control-ROM Binary Content P856

ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR	ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR	ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR	
000	00	111101000	00011	00001	00	10	11	00	00	0	111	00	0	00000	00000	040	11	111111111	01100	01000	11	10	11	10	11	0	000	10	0	11100	10000	080	00	000001000	00000	00001	00	00	00	01	00	0	000	00	0	00000	00000	
001	11	111111111	00000	00000	11	10	11	10	11	0	000	10	0	11100	00000	041	01	111001000	10110	01000	00	10	00	00	00	0	000	10	1	00000	00000	081	00	100000000	00000	00000	00	00	00	00	0	000	00	0	00000	00000		
002	00	111101001	00000	00000	00	00	10	11	0	000	00	1	00000	00000	042	00	011111010	00000	00000	01	10	00	00	11	0	000	10	0	00000	00000	082	00	110110010	01010	01110	00	00	00	00	0	000	00	0	100	11	0	00000	00000
003	01	011010000	10000	00011	00	00	11	00	00	0	000	00	0	11011	00000	043	00	011110101	00000	00000	01	10	00	00	11	0	000	10	0	00000	00000	083	10	100000000	00101	00011	00	10	11	00	00	0	000	00	0	00000	00000	
004	01	111110000	11110	10010	00	00	11	00	00	0	000	00	0	00000	00000	044	00	100000001	00100	01000	00	00	00	00	0	011	00	1	00000	10000	084	00	110001011	01100	10010	00	00	00	00	0	1	000	01	0	00000	00000		
005	00	111110111	00000	00000	00	00	11	10	0	000	00	0	01110	00000	045	00	100000001	00000	01010	00	00	00	00	0	011	00	1	00000	00000	085	00	110001011	01100	10001	00	00	01	00	00	1	000	01	0	00000	00000			
006	00	111111111	10100	00011	00	10	00	00	00	0	000	00	0	00000	00000	046	11	111111111	00100	01000	11	10	11	10	11	0	000	10	0	11100	10000	086	00	101101001	00000	00000	00	00	00	00	0	1	000	00	0	00000	00000	
007	00	111111111	00000	01110	01	10	00	00	00	0	000	10	0	00000	00000	047	00	111101001	00000	00000	00	00	00	00	0	000	10	1	00000	01100	087	00	101111001	00101	00011	00	00	11	00	00	0	000	00	0	00000	00000		
008	11	111111111	01100	00001	11	10	11	10	11	0	000	10	0	11100	10000	048	01	110100110	00001	11100	00	00	00	00	0	000	00	0	00100	00000	088	01	101110011	00000	01010	00	10	00	00	0	0	000	00	1	11111	00000		
009	00	111101001	00000	00001	00	00	01	01	0	000	00	1	00000	10000	049	01	101000000	00001	00111	00	11	11	00	00	0	000	00	0	00000	00000	089	00	010110110	01001	01010	00	00	11	00	00	0	000	00	0	01111	00000		
00A	00	101101011	11110	00011	00	00	00	00	0	111	00	0	00000	00000	04A	00	011100010	10100	00000	00	10	00	00	0	000	00	0	00000	00000	08A	00	100110000	00000	00011	00	00	00	00	0	111	00	1	00000	00000				
00B	00	111111111	00000	00011	00	10	00	00	00	0	000	00	0	00000	00000	04B	01	110110101	01100	10010	00	00	00	11	0	000	00	0	00000	00000	08B	01	011010000	10000	00011	00	00	11	00	00	0	000	00	0	11011	00000		
00C	00	100000001	00100	00011	00	00	00	00	0	011	00	1	00000	00000	04C	01	110110010	00111	00011	00	00	10	10	1	011	10	0	00000	00000	08C	01	010101011	11010	01011	00	10	11	00	00	0	001	00	0	10101	00000			
00D	00	100000001	10100	00011	00	00	00	00	0	011	00	1	00000	00000	04D	00	111101001	00000	00001	00	00	01	01	0	000	00	1	00000	00000	08D	00	110101000	00000	00000	00	00	00	00	0	0	000	10	0	01101	00000			
00E	00	011100110	11011	00011	00	10	00	00	0	000	00	0	00000	00000	04E	01	110101000	00000	00000	00	01	00	00	0	000	00	0	01101	00000	08E	01	111010100	10100	00011	00	00	10	00	0	000	10	0	00100	00000				
00F	00	011100101	11011	00011	00	10	00	00	0	000	00	0	00000	00000	04F	00	101110010	00000	01110	01	10	00	00	0	011	10	0	10001	00000	08F	00	101011001	00000	01101	00	10	01	00	00	0	000	00	0	00000	00000			
010	01	111101000	11110	00011	00	00	11	00	00	0	000	00	0	00000	00000	050	11	111111111	01100	00010	11	10	11	10	11	0	000	10	0	11100	10000	090	00	110101010	11010	01011	00	10	00	00	0	0	000	00	0	00000	00000	
011	00	111101001	00000	00001	00	00	01	01	0	000	00	1	00000	00000	051	01	110001000	10110	00010	00	10	00	00	00	0	000	00	0	00000	00000	091	10	100000011	10100	00011	00	00	01	00	0	000	00	0	00000	00000			
012	01	011100000	01000	00001	00	00	00	00	0	100	00	0	10011	00000	052	00	011101100	00000	01110	00	00	01	11	0	000	10	1	00000	00000	092	00	111101101	10100	00011	00	10	00	00	0	100	11	0	11100	00000				
013	00	011100000	00000	00011	00	00	01	01	0	000	00	0	00000	00000	053	00	011101000	00000	01110	00	00	01	11	0	000	10	1	00000	00000	093	10	100000000	10100	00011	00	10	11	00	00	0	000	00	0	00000	00000			
014	00	111011000	11011	10010	00	10	00	00	0	000	00	0	00000	00000	054	00	100000001	00100	00010	00	00	00	00	0	011	00	1	00000	10000	094	00	110001011	01100	01111	00	00	00	00	0	1	000	01	0	00011	00000			
015	00	111011000	11011	10010	00	10	00	00	0	000	00	0	01110	00000	055	00	010101000	01000	01101	00	11	00	00	0	000	00	0	00000	00000	095	01	111000111	01000	00001	00	00	00	00	0	000	00	0	00000	00000				
016	11	111111111	00000	00000	11	10	11	10	11	0	000	10	0	11100	00000	056	01	110100000	00000	00000	00	01	00	00	0	000	00	0	00000	00000	096	10	100000000	00111	00011	00	10	00	00	0	0	000	00	0	00000	00000		
017	11	111111111	00000	00000	00	00	00	00	0	000	10	0	00001	00000	057	01	110010000	11110	10010	00	11	00	00	0	000	00	0	10001	00000	097	00	101101001	10100	00011	00	00	11	00	00	1	000	00	0	00000	00000			
018	01	110010011	10100	00011	00	10	00	00	0	000	00	0	01111	00000	058	00	110100110	01001	11100	00	00	00	00	0	000	00	0	00000	00000	098	01	010110011	00000	01010	00	10	00	00	0	0	000	00	1	11111	00000			
019	01	101011011	10100	00011	00	10	01	00	00	0	000	00	0	01111	00000	059	01	101000000	00001	00111	00	11	11	00	00	0	000	00	0	00000	00000	099	00	010100101	00010	10010	00	00	11	00	00	0	000	00	0	00000	00000	
01A	01	101110111	00100	00011	00	00	11	00	00	0	000	00	0	00000	00000	05A	00	011100010	10100	00000	00	10	00	00	0	000	00	0	00000	00000	09A	00	010010101	00001	00011	00	00	00	00	0	111	00	1	11111	00000			
01B	00	011001110	01001	10001	00	00	01	00	00	1	000	01	0	00110	00000	05B	01	110100101	01100	01111	00	00	10	00	11	0	000	00	0	00011	00000	09B	00	011111110	01101	00001	00	00	00	00	0	0	000	00	0	00000	00000	
01C	01	101101011	10100	00011	00	10	00	00	0	000	00	0	00000	00000	05C	01	110100010	00111	00011	00	00	00	11	10	1	011	10	0	00000	00000	09C	01	010110011	11011	01111	00	10	00	00	0	000	00	1	11111	00000			
01D	01	101001011	10100	00011	00	10	01	00	00	0	000	00	0	00000	00000	05D	00	010111110	00000	00001	00	11	00	01	0	000	00	0	00000	00000	09D	00	110100011	10100	00011	10	10	00	10	00	0	011	10	1	00110	00000		
01E	00	011011101	00100	00011	00	00	11	00	00	0	000	00	0	00000	00000	05E	00	110011000	00000	00000	00	00	01	00	0	000	00	0	00000	00000	09E	00	101100000	10100	00011	00	10	00	00	0	0	000	00	0	1			

Table 2-5B contd.

ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR	ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR	ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR
OC0	11	11111111	01100	00001	11	10	11	10	11	0	000	10	0	11100	00000	100	00	00011111	00000	00000	00	00	00	00	0	000	00	0	01001	00000	140	00	00111111	00000	00000	00	00	00	00	0	100	00	0	10011	00000		
OC1	00	11101001	00000	00001	00	00	01	01	0	000	00	1	00000	00000	101	00	110011010	00000	00011	00	11	00	01	0	000	00	1	00000	00000	141	00	11101001	01101	00000	00	00	00	00	0	000	00	1	01011	00000			
OC2	00	10010101	00000	01001	00	00	00	00	00	0	000	00	1	01100	00000	102	00	011101100	00101	00110	00	00	01	00	0	000	00	1	00000	00000	142	00	11100111	01100	01010	00	00	00	00	0	000	00	1	00000	00000		
OC3	00	01101010	00000	00000	01	10	00	00	11	0	000	10	0	00000	00000	103	00	110101101	00101	00110	00	00	01	00	0	000	00	1	00000	00000	143	00	100000001	10000	00000	00	00	00	00	0	111	00	1	00000	11000		
OC4	00	00100111	01010	00110	00	00	00	00	00	0	000	00	0	00000	00000	104	00	011010011	00000	00000	01	10	00	00	0	100	11	0	00000	00000	144	00	110001000	00000	00001	00	00	00	00	0	111	11	0	00000	01000		
OC5	00	11101001	00000	00011	00	00	00	01	01	0	000	00	1	00000	00000	105	01	011010100	00000	00011	00	00	00	00	0	000	00	0	00100	00000	145	00	010110111	00010	00011	00	00	11	00	0	000	00	0	00000	00000		
OC6	00	010001000	00100	00011	00	00	11	00	00	0	000	00	0	00000	00000	106	01	011010000	10000	00011	00	00	11	00	0	000	00	0	11011	00000	146	01	000011100	11011	00110	00	10	00	00	0	000	00	0	00000	00000		
OC7	11	11111111	00100	00000	11	10	11	10	11	0	000	10	0	11100	11100	107	00	001100110	00000	01101	00	10	00	00	0	000	00	0	00000	00000	147	00	000011000	00000	00000	00	00	01	00	0	000	00	0	00000	00000		
OC8	00	010000110	00100	01011	10	10	00	00	00	0	000	00	1	10000	00000	108	01	011010000	10000	00011	00	00	11	00	0	000	00	0	11011	00000	148	00	010110110	01001	10100	00	00	10	00	0	1	000	01	0	00000	00000	
OC9	11	11111111	01100	01001	11	10	11	10	11	0	000	10	0	11100	00000	109	00	001001011	00101	00011	00	00	00	01	0	000	00	0	00000	00000	149	00	010110101	01001	10100	00	11	00	00	0	000	00	0	11011	00000		
OCA	00	11101001	00000	01110	00	00	01	01	0	000	00	1	00000	00000	10A	00	011010000	00101	00110	00	00	01	00	0	000	00	1	00000	00000	14A	00	010110100	01001	10100	00	11	00	00	0	000	00	0	11011	00000			
OCB	00	100110010	00000	00000	00	00	00	00	00	0	000	00	0	01001	00000	10B	00	110101100	00101	00110	00	00	01	00	0	000	00	1	00000	00000	14B	00	011010111	00010	00011	00	00	00	00	0	000	00	0	00000	00000		
OCC	00	100000001	00100	00011	00	00	00	00	00	0	011	00	1	10000	00000	10C	00	001001010	11011	00011	00	10	11	00	0	000	00	0	00000	00000	14C	00	010101101	00000	00001	00	00	10	00	0	000	10	1	11111	00000		
OCD	00	10011101	00000	01101	00	00	01	00	0	000	00	0	01100	00000	10D	00	111010001	00000	00011	00	00	01	01	0	000	00	1	00000	00000	14D	00	001000010	00000	00000	00	00	10	00	0	000	10	0	000	10	1	11111	00000
OCE	01	100101000	00000	00110	10	10	01	00	00	0	000	00	0	00000	00000	10E	00	111101011	00000	00000	00	00	00	01	0	000	00	0	00000	00000	14E	00	001010011	11011	00110	00	10	11	10	0	011	00	0	00000	00000		
OCF	00	101101000	00000	00000	11	10	11	00	00	0	111	11	0	11100	00000	10F	00	100100101	11011	10010	00	00	11	00	0	000	00	0	00000	00000	14F	00	001101100	01001	00001	00	00	11	10	0	000	00	0	00000	00000		
OD0	01	011010000	10000	00011	00	00	11	00	00	0	000	00	0	11011	00000	110	00	011011111	00000	00000	00	00	00	00	0	000	00	0	01101	00000	150	00	010011111	00000	00000	00	00	00	00	0	000	00	0	01101	00000		
OD1	01	011010000	10000	00011	00	00	11	00	00	0	000	00	0	11011	00000	111	10	100000000	00101	00110	00	00	11	01	00	0	000	00	0	00000	00000	151	00	010100001	01001	01001	00	10	11	00	0	000	00	0	00000	00000	
OD2	00	100100101	01101	00001	00	00	00	00	00	0	000	00	0	00000	00000	112	00	100101100	00101	00110	00	00	01	00	0	000	00	1	00000	00000	152	01	000110000	00000	00000	00	11	00	10	0	000	10	1	11111	00000		
OD3	10	100000000	00000	01110	01	00	11	01	11	0	000	10	0	00000	00000	113	10	100000000	10100	00011	01	10	11	01	00	0	000	10	0	00000	00000	153	00	101110011	00000	10010	00	00	01	00	0	000	00	0	00000	00000	
OD4	01	100011011	01010	00000	01	10	00	01	00	0	000	10	0	00111	00000	114	00	001001000	01010	00110	01	10	00	01	00	0	000	10	0	00111	00000	154	00	000010000	00000	00101	00	00	11	10	00	0	001	10	1	10100	00000
OD5	00	001010110	01001	00110	00	00	00	00	00	0	000	00	0	00000	00000	115	00	001010110	01010	00110	00	00	00	00	0	000	00	0	00000	00000	155	00	000010000	00000	00000	00	00	10	00	0	001	10	1	10100	00000		
OD6	01	100100000	00000	00000	00	00	01	00	00	0	000	00	0	00000	00000	116	00	010101110	11010	00011	00	10	00	00	0	000	00	0	01101	00000	156	00	011101001	00000	00000	00	00	00	00	0	000	00	0	00000	00000		
OD7	01	011111000	00000	01001	00	00	11	00	00	0	000	00	0	00000	00000	117	10	100000000	00000	00000	01	10	11	00	0	000	10	0	00000	00000	157	00	010100111	00000	00010	00	00	01	00	0	100	00	1	10111	00000		
OD8	11	11111111	00100	00000	11	10	11	10	11	0	000	10	0	11100	11100	118	00	011011100	01101	00110	00	00	01	00	0	000	00	1	00000	00000	158	00	010100110	01100	01110	11	10	00	00	0	000	10	0	00110	01000		
OD9	11	11111111	10100	00000	11	10	11	10	11	0	000	10	0	11100	11100	119	00	111110011	01101	00000	00	00	00	00	0	000	00	0	01101	00000	159	00	110001000	01100	00010	00	00	01	00	0	000	00	0	00000	00000		
ODA	00	100100100	10110	00011	00	10	00	00	00	0	000	00	0	01001	00000	11A	00	111110010	01101	00000	00	00	00	00	0	000	00	0	01011	00000	15A	00	010100100	01001	00011	00	00	00	00	0	000	00	0	01101	00000		
ODB	00	100100010	00011	00011	00	00	01	00	0	001	00	1	10101	00000	11B	00	111010001	00100	00011	00	00	00	00	0	000	00	1	00101	10000	15B	00	010100010	01001	11110	00	00	01	00	0	1	000	00	0	00111	00000		
ODC	00	010001100	00100	00011	10	10	00	00	00	0	000	00	1	10000	00000	11C	01	110110101	00100	00011	00	00	00	00	0	000	00	0	00000	00000	15C	00	010100010	01001	11110	00	00	01	00	0	1	000	01	0	00000	00000	
ODD	00	100011101	00000	00001	00	11	00	11	00	0	011	10	1	00000	00000	11D	00	100111010	01101	01101	00	00	00	00	0	000	00	0	00000	00000	15D	01	110110110	01001	11100	00	00	01	00	0	000	00	0	00100	00000		
ODE	01	100011000	00000	00000	00	00	01	00	0	000	00	0	00000	00000	11E	00	111101100	01110	01110	00	00	00	00	0	100	11	0	00000	00000	15E	00	010100000	11011	00110	10	10	00	01	0	000	00	0	00000	00000			
ODF	01	011111000	00000	01001	00	00	11	00	00	0	000	00	0	00000	00000	11F</																															

Table 2-5B contd.

ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR	ADD.	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR
180	00	000101000	00000	00000	00	11	00	00	00	0	100	00	0	10011	00000	100	00	01001	00000	00	00	00	00	0	100	00	0	10011	00000		
181	00	000100000	00000	00000	01	10	00	00	00	0	000	10	0	00000	00000	101	00	01001	00000	00	00	00	00	0	000	00	0	00000	00000		
182	00	001010000	11010	10010	00	10	00	00	00	0	000	00	0	00000	00000	102	01	000101000	00000	01101	00	10	11	00	0	000	00	0	00100	00000	
183	00	011101001	00000	00000	00	00	00	00	00	0	000	00	0	00000	00000	103	00	010000101	11011	01011	00	10	11	00	0	000	00	0	01101	00000	
184	00	001111010	00100	00011	00	00	00	00	00	0	111	00	1	10110	00000	104	01	011110001	01101	00001	00	11	00	00	0	000	00	0	00000	00000	
185	00	110001000	00100	00011	00	00	00	00	00	0	111	10	0	00000	00000	105	00	110001000	01100	01110	01	00	00	00	0	000	10	0	00000	00100	
186	01	001010100	00011	00000	00	00	10	00	00	0	000	00	0	00100	00000	106	00	011001100	00000	00000	01	00	11	00	0	000	10	0	00000	00000	
187	01	001100100	00111	00000	00	00	01	00	00	1	000	00	0	00100	00000	107	00	000101001	00000	00001	10	10	00	00	0	111	00	0	00000	00000	
188	00	001110110	01101	00110	00	00	01	00	00	0	000	00	1	00000	00000	108	00	000110110	01010	00000	00	00	00	00	0	000	00	0	00000	00000	
189	10	100000000	00000	01110	00	00	00	11	01	0	000	10	1	00000	00000	109	00	000111011	11011	01110	00	10	00	00	0	000	00	0	00000	00000	
18A	00	010011111	00000	00000	00	00	00	00	00	0	000	00	0	00000	00000	10A	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000	
18B	00	000011011	00000	01010	00	00	10	00	10	0	000	00	0	00000	00000	10B	00	001110001	11011	01111	00	10	00	00	0	000	00	0	00000	00000	
18C	00	001110010	11011	00110	00	10	00	00	00	0	000	00	0	01010	00000	10C	00	001111111	00000	00101	00	00	11	00	0	000	00	0	00000	00000	
18D	00	001101110	01000	00101	00	11	00	00	00	0	000	00	0	00000	00000	10D	00	010011000	01000	00001	00	11	00	00	0	000	10	0	01101	00000	
18E	01	000100100	00001	00111	00	00	00	00	0	000	00	0	00100	00000	10E	00	010101100	01000	00001	00	00	00	00	0	000	10	0	11111	00000		
18F	00	001100001	11011	00100	00	10	00	00	00	0	000	00	0	00000	00000	10F	00	110111000	10100	00011	00	00	01	00	0	000	10	0	11111	00000	
190	00	000011001	00000	00101	00	10	00	00	00	0	000	00	0	00000	00000	100	00	000101111	00000	00000	00	00	00	00	0	000	00	0	00000	00000	
191	00	110111110	00000	00010	10	10	11	00	00	0	000	00	0	00000	00000	101	00	001101010	01100	00000	00	00	00	01	00	0	000	00	0	00000	00000
192	00	010011111	00000	00000	00	00	00	00	00	0	000	00	0	00000	00000	102	01	001111100	00000	01101	00	10	10	00	0	000	00	0	00100	00000	
193	01	000000010	00000	00000	01	10	00	00	00	0	000	10	0	00000	00000	103	00	000110010	00000	00000	00	00	00	00	0	000	00	0	00010	00000	
194	00	001101010	01100	00000	00	00	00	01	00	0	000	00	0	01011	00000	104	01	011101010	01001	00110	00	00	00	01	00	0	000	10	0	00111	00000
195	00	001101000	10110	00011	00	00	10	00	00	0	011	00	1	00000	00000	105	01	011101010	01001	00110	00	00	00	01	00	0	000	10	0	10010	00000
196	00	011101001	00000	00000	00	00	00	00	00	0	000	00	0	00000	00000	106	11	010011110	00000	00000	00	00	00	00	0	000	10	0	00000	00000	
197	00	001100111	10100	00011	00	11	00	10	00	0	011	10	1	00000	00000	107	01	011011000	00000	00001	01	00	11	00	0	111	10	0	00000	00000	
198	00	111101001	00000	00001	00	00	01	01	0	000	10	1	00000	00000	108	01	010000100	11010	00111	00	00	00	00	0	000	00	0	00100	00000		
199	00	001011110	00000	01101	00	10	00	00	00	0	000	00	0	00000	00000	109	00	000010111	10100	00000	00	00	00	01	0	000	00	0	00000	00000	
19A	00	010011111	00000	00000	00	00	00	00	00	0	000	00	0	00000	00000	10A	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000	
19B	01	001111000	00000	00000	00	11	00	00	00	0	000	00	0	00000	00000	10B	00	001110000	11010	10010	00	10	00	00	0	000	00	0	00000	00000	
19C	01	001011100	10100	00111	00	00	11	10	0	000	00	0	00100	00000	10C	01	010011000	00000	00000	00	11	00	00	0	000	00	0	00000	00000		
19D	00	011111011	00000	00000	00	00	00	00	00	0	100	00	0	10011	00000	10D	00	001111000	11011	01111	00	10	11	00	0	000	00	0	00000	00000	
19E	00	001100000	01000	01001	00	10	00	00	00	0	000	00	0	00000	00000	10E	01	000001100	00000	00101	00	00	00	00	0	000	00	0	00000	00000	
19F	00	001010001	10100	00110	00	10	00	00	01	0	111	00	0	00000	00000	10F	00	001001001	00000	00001	00	00	00	00	0	111	00	0	00000	00000	
1A0	01	100111010	01000	00001	00	11	00	00	10	0	000	00	0	00000	00000	1E0	00	100000000	00000	00001	00	00	00	01	0	000	00	0	00000	00000	
1A1	01	101110100	11010	00111	00	00	00	00	00	0	000	00	0	00100	00000	1E1	01	110010101	00001	00011	00	00	00	00	0	000	00	0	00000	00000	
1A2	00	010011111	00000	00000	00	00	00	00	00	0	000	00	0	00000	00000	1E2	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000	
1A3	01	001100010	00000	00000	01	10	00	00	00	0	000	10	0	00000	00000	1E3	00	000100001	00000	00101	00	10	00	00	0	000	00	0	00000	00000	
1A4	00	001011010	00101	00000	00	10	00	00	00	0	000	00	0	00000	00000	1E4	01	000011010	00000	00000	00	00	00	10	0	000	00	0	00000	00000	
1A5	00	100101011	01001	00110	00	11	00	01	00	0	000	00	1	00000	00000	1E5	00	000110011	00000	00000	00	11	00	00	0	000	00	0	00000	00000	
1A6	00	010010110	11011	01011	00	10	00	00	00	0	111	11	0	01110	01000	1E6	01	001001100	00000	01101	00	10	00	00	0	000	00	0	00100	00000	
1A7	01	001011000	00000	00000	00	00	01	00	00	0	111	00	0	10111	00000	1E7	00	001001100	00000	00000	00	11	00	00	0	000	00	0	00000	00000	
1A8	01	100101010	01001	00000	00	00	01	00	00	0	000	10	0	00111	00000	1E8	00	000010110	10100	00110	00	10	00	00	0	000	00	0	00000	00000	
1A9	00	100111010	01101	01010	00	00	00	00	00	0	000	00	0	00000	00000	1E9	01	001110100	11010	00111	00	00	00	00	0	000	00	0	00100	00000	
1AA	00	010011111	00000	00000	00	00	00	00	00	0	000	00	0	00000	00000	1EA	00	010011111	00000	00000	00	00	00	00	0	000	00	0	00000	00000	
1AB	00	000010011	01010	10000	00	00	00	00	00	0	000	00	0	00000	00000	1EB	00	000110011	11010	01111	00	10	00	00	0	000	00	0	00000	00000	
1AC	01	001010010	00000	00001	00	00	00	11	10	0	011	10	0	00000	00000	1EC	00	000010010	11010	00011	00	10	11	00	0	000	00	0	00000	00000	
1AD	00	010110000	00000	00011	00	00	00	01	01	0	000	00	0	00000	00000	1ED	01	001000100	00010	00000	00	00	10	00	0	000	00	0	00100	00000	
1AE	10	111111111	01001	00001	11	10	11	00	10	0	000	11	0	11100	00000	1EE	00	1111													

2.48 INSTRUCTION WORD LOGIC

The instruction word logic (Figure 2-8AA) comprises the K register and the instruction decoder. This logic stores and decodes the program instruction word for use by the Microprogram Control.

2.49 K -- Instruction Register

This 16-bit register holds the instruction word which is obtained from memory via the GP-Bus BIO lines. The instruction-word format is given in Section I (paragraph 1.62). Four type 74175 IC chips are used as a straight 16-bit buffer register, with both high and low outputs available for decoding. Bus data BIO00-15R are loaded by CLK at the trailing edge of BP, when GFETCH is active. GFETCH is a general-field command bit (Table 2-3).

2.50 Instruction Decoder

The instruction decoder uses a programmable logic array (PLA) as the main decoding logic. The PLA provides an address code to the microinstruction control-store for 96 different input combinations. A set of gate-logic decoders in parallel with the PLA are used for ineffective branches and FPP instructions without attached FPP. The PLA0, PLA1 control logic controls some of the PLA and the gate-logic decoding. These instruction-decoder outputs are used with the Microprogram Control and are described in paragraphs 2.37 and 2.40 (Microinstr. Instr. Word, M.S. Pointer).

2.51 Logic gates are used to decode the K register into basic fields for controlling the PLA decoder and gate decoders. These gates may indicate when the instruction R1 or R2 fields contain 0 or 15, or when a branch condition is indicated and verified. Some bits of the K register are used as direct control bits for the PLA decoder. Complete PLA decoding is shown in Table 2-6.

2.52 PLA ROM maps are provided (Table 2-7) to show the PLA output code (as a two-bit hexadecimal number) produced for the various instructions or addressing types. The PLA output code relates directly to the control-store ROM address used by the microprogram control.

Table 2-6 Instruction Decoder PLA

	Data Input	Output	Comments	Group
	I14 ← → I0	F8 ← → F1		
	PLA0 PLA1 K0 K1 K2 K3 K4 R1E0 R1E15 K9 K10 R2E0 R2E15 K15	RAD1,2 3,4 5,6 RAD7,8		
1	LL - - - - -	LLLLLLLL	Inhibit mode	
2	HHL LHL L H - - - - -	LLLLLLLL	INH, RIT, HLT	Privileged Instruction Detection
3	HHL HL L -L - - - - -	LLLLLLLL	I/O instructions	
4	HHL HHH -L - - - - -	LLLLLLLL	RER, WER	
5	HHH - - - -H - - - - -	LLLLLLLL	Format 1 with R1=15	
6	HHHL LLL - -LHLH -	LLLLLLLL	LD, ST with R2=15	
7	HHHL HHH - -LHLH -	LLLLLLLL	ML, MS with R2=15	
8	HH -HHH -H - - - -HL	LLLLLLLL	MVF, MVB, RTN with R2=15	
9	HHHL HHHH - - - - -	LLLLLLLL	TS, TS	
10	HHHHL HL L - - - - -	LLLLLLLL	EL, ES	
11	H -LH - - - - -	L - - - - -	K1	
12	H -L -H - - - - -	-L - - - - -	K2	
13	H -L - -H - - - - -	- -L - - - -	K3	
14	H -L - - -H - - - - -	- - -L - - -	K4	
15	H -L - - - -H - - - - -	- - - - -L	OR1	
16	H -L L L L L - - - - -	LL - - - - -	LDK	
17	H -L L L L H - - - - -	- - - - -L	ABK	
18	H -LHLH - - - - -	- - - - -L	RF, RB	
19	H -LHL L L - -HLH -L	- - - - -L	WMP	
20	H -L L H H H - - - -H -L	- - - - -L - -	Shift with n=0	
21	H -H - - - - -	- - - - -L -	K0	Addressing Routines Format 1
22	H -H - - - - -H - - - -	-L - - - - -	K9	
23	H -H - - - - -H - - - -	- -L - - - -	K10	
24	H -H - - - - -H - - - -	- - -L - - -	OR2	
25	H -H - - - - - -H - - - -	- - - - -L	K15	
26	H -H L L L H - -LHL - -	- - - - -L	OPC-1 T3 } Store	
27	H -H H H L L - - - - -	L - - - - -L	OPC-12 Char. Instr.	
28	H -HL L L L - -LHLHL	L - - - - -	LDR T3 15R2	
29	H -HL H H H L -LHLHL	- - - - -L - -	MLR T3 15R2	
30	H -H H L L -L - - - - -	L - - - - -L	Floating Point	
31	H -H - - - - -L L - - -	L - - - - -L	T1 routine	
32	H -H -L L L - - - - -H	L - - - - -L	ST, TS, MS	
33	H -HLHL L H - - - - -H	L - - - - -L	CM	
34	H -HL H H H - - - - -	L - - - - -L	OPC-7, K0	
35	H -HHLHL L - - - - -	L - - - - -L	EL, ES	
36	H -H H H L H - - - - -H	L - - - - -L	CC	
37	H -HHLH -H -LHL - -	L - - - - -L	Dar *, DSR *	
38	H -HHLH -H -H - - - -	L - - - - -L	DA, DS	
39	H -H H H H L - - - - -L	- - - - -L - -	Return	
40	H H H H H H L - - - - -L	- - -L - - - -	Return, User	

	Data input	Output	Comments	Group
	114 ← → 10	F8 ← → F1		
	PLA0 PLA1 K0 K1 K2 K3 K4 R1E0 R1E15 K9 K10 R2E0 R2E15 K15	RAD1,2 3,4 5,6 RAD7,8		
41	L H H H - - - - -	L - - - L - - -	K0.K1	Execution
42	L H H - H - - - -	- L - - - - -	K0.K2	
43	L H H - - H - - -	- - L - - - -	K0.K3	
44	L H - - - - H - - -	- - - L - - -	K4	
45	L H H - - - - H - - H	- - - - L - -	K0.K9.K15	
46	L H - - - - - H - - - -	- - - - - L -	OR1	
47	L H H L L H H - - L L - - H	- - - - - L - L	NGR	
48	L H H L H - L - - L L - - H	- - - - - L L -	TM, TNM	
49	L H - L L L H - - - - -	- - - - - L -	AB	
50	L H - L L L L - - - - -	- - - - - L - - -	LD, ST	
51	L H H L L L L - - L H - H H	- - - - - L -	STR with R2=15	
52	L H H H L H - H - L H H - -	- - - - - L -	DAK, DSK	
53	- - H H L H - H - L L - - -	- - - - - L - -	DAR, DSR (ADD. also)	
54	L H H H H L L - - L L - - -	- - - - - L -	ECR	
55	L H - H H H L - - L L - - -	- - - - - L - -	CR, EX	
56	L H H H H H H - - - - -	- - - - - L -	C1	
57	L H - L H H H - - - - -	- - - - - L - - -	ML, MS, TL, TS, Shift	
58	L H H L H H H L - L H L H H	- - - - - L -	MSR with R2=15	
59	L H H L H H H - - L H H - -	L - - - - - -	MLK	
60	L H L L H H H - - H - - - -	- - - - - L -	Shift K9	
61	L H L L H H H - - - H - - -	- - - - - L - -	Shift K10	
62	L H - H H H L - - - - H L	- - - - - L -	RTN with R2=15	
63	L H H H L L H L - L L - - -	- - - - - L -	FFX, FFL	
64	H - - L H H L H - - - - -	L L L L L L L L	XRK OR1	Trap Detection
65	H - L H L L - H - - - - -	L L L L L L L L	I/O instr. OR1	
66	H - - H L H H - - - - - H	L L L L L L L L	RB K15	
67	H - L H H L L - - - - -	L L L L L L L L	OPC-12 K0	
68	H - H L - - - - L H H - H	L L L L L L L L	K0.K1.K15 (T2)	
69	H - H L L L H - - H - - H	L L L L L L L L] AB K15	
70	H - H L L L H - - - H - H	L L L L L L L L		
71	H - H L L H H H - L L - - H	L L L L L L L L	SUR K15	
72	H - H L H L - H - - - - L	L L L L L L L L	AN, OR OR1, K15	
73	H - H L H L L H - L L - - -	L L L L L L L L	ANR OR1	
74	H - H L H L H H - - - - H	L L L L L L L L	ORS OR1	
75	H - H L H L H - - L L - - H	L L L L L L L L	ORR K15	
76	H - H L H H H - - L L - - -	L L L L L L L L	OPC-7 (T1)	
77	H - H L H H H H - L H H - -	L L L L L L L L	MLK OR1	
78	H - H H L - - H - - - - H	L L L L L L L L	OPC-8 to 11 OR1 .K15	
79	H - H H L - - - - L H H - H	L L L L L L L L	OPC-8 to 11 T2.K15	
80	H - H H L - L L - L L - - -	L L L L L L L L	OPC-8, 10 T1.OR1	
81	H - H H L L - L - L H H - -	L L L L L L L L	OPC-8, 9 T2.OR1	

	Data Input	Output	Comments	Group
	114 ← → 10	F8 ← → F1		
	PLA0 PLA1 K0 K1 K2 K3 K4 R1E0 R1E15 K9 K10 R2E0 R2E15 K15	RAD1,2 3,4 5,6 RAD7,8		
82	H - H H L H L L - L H H - -	L L L L L L L L	DAK OR1	Trap Detection
83	H - H H H L L H - - - - -	L L L L L L L L	OPC-12 OR1	
84	H - H H H L L - - L H H - H	L L L L L L L L	OPC-12 T2.K15	
85	H - H H H L H H - H - - - H	L L L L L L L L] CC OR1	
86	H - H H H L H H - - H - - H	L L L L L L L L		
87	H - H H H H L L - - - - - L	L L L L L L L L	OPC-14 OR1, K15	
88	H - H H H H - H - L L - - L	L L L L L L L L	OPC-14,15 OR1, K15, T2	
89	H - H H H H - - - L H H - L	L L L L L L L L	OPC-14,15 K15, T2	
90	H - H H H H - H - H - - - L	L L L L L L L L	OPC-14,15 OR1, K15, K9	
91	H - H H H H H H - L L - - -	L L L L L L L L	C1R OR1	
92	H - H H H H H - - L H H - -	L L L L L L L L	C2 T2	
93	H - H H H H H H - L H L - L	L L L L L L L L	C1R * OR1	
94	L H H - - - - - H - - H	- - - - - L - -	K0 K10 K15	
95	L H H H L L H L - L L - - H	- - - - - L - -	FFX	
96	L H H H H H L - H - - - - H	- - - - - L -	CF with R1=15	

Table 2-7 PLA ROM Map

PLA Output
 F:

8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---

 Example: Addressing type RT5S = /43 = 0100 0011₂
 MSB LSB

MSB \ LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		NO JUMP	NO JUMP P+2	NO FPP												
1		ABK														
2	ADK	ADKP	RT3	RT3S			RTN									
3	SJK	SJKP	RT2	RT2S			RTN USER									
4	ANK	HLY RIT INH	RT5	RT5S												
5	ORK	ENB SMD LKM	RT4	RT4S												
6	XRK		RT7	RT7S												
7	SM n ≠ 0	DSH n ≠ 0	RT6	RT6S	SH n ≠ 0	DSH n ≠ 0										
8	C10 CTR	WMP		RT1				RT1D								
9	SST TST INF			RT1P				RT1DP								
A		RF	RT3B	RT3C				RT3BM								
B		RB		RT2C												
C	LCK	LCKP		RT5C												
D	CWC	CWCP		RT4C												
E	WER	WVP		RT7C												
F	RER	WVB		RT6C												

ADDRESSING MODE

MSB \ LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0									LC	LCP			ST	STP	STD	STDP
1		AB							SLA SLN	OLA OLN	SLL SLC	OLL OLC	SRA SRN	DRA DRN	SPL SRC	ORL ORC
2	AD	ADP			ADS	IM										
3	SU	SUP			SUS	C2 NGR										
4	AN				ANS	CP	TM									
5	OR				ORS											
6	XR				XRS		TNM									
7									ML	TL			MS	TS	MSRC	
8									FL	ML			FS			
9									FC	CV	FFL		FCS		FFX	
A									EL	DAR DA		DAK	ES		DAR	
B										DSR DS		DSK			DSR	
C									LC	ECP			CC			
D									CW	CWP			CC			
E										RTN		RTN A15	CF	EX	CF 15R1	
F									MLK	C1				C15		

EXECUTION MODE

2.53 DATA HANDLING LOGIC

The Data Handling Logic (Figure 2-4) comprises the arithmetic logic unit (ALU), data and address storage and handling registers, and the logic for the data path. The basic components of the Data Handling Logic are :

Logic	Para.	Diagram
ALU -- Arithmetic Unit	2.59	2-8 GG
M -- CPU Working Register (multi-in) : load C ; load Q .	2.64	2-8 GG
D Selector : ALU direct ; ALU exchange character ; ALU shift right ; BIO .	2.61	2-8 HH
L -- Data Register (multi-in) : Load D direct ; load D shift left. (operation result; output buffer)	2.63	2-8 HH
C Selector : D0-15 or 8-15; BIO0-15 or 8-15; INTAD 0-5; ASRO-7.	2.65	2-8 JJ
Q -- Shift Register : load; shift right; shift left.	2.67	2-8 JJ
S -- Address Register/Counter	2.71	2-8 KK
A - Bus Selection	2.72	2-8 LL
IPL -- Initial Program Loader	2.76	2-8 LL
P -- Program Register/Counter	2.77	2-8 LL
A0-A15 -- Scratchpad	2.79	2-8 MM
PSW - Program Status Word	2.84	
PLR -- Priority Level Register	2.85	2-8 NN
CR -- Condition Register	2.86	2-8 NN
GF -- General Flip-Flops	2.89	2-8 PP
A, D, L Command	2.56	2-8 FF

The Data Handling Logic performs parallel processing of 16-bit words. Double-length operations are provided for processing of 31-bit words. Two 16-bit data paths loop through the arithmetic unit (ALU) : one loop for operand A and one loop for operand B.

2.54 Operand A is supplied by the A-Bus. A-Bus control logic selects which of the sources on the A-Bus is to be used as the operand. The various sources are connected by open-collector gates so only the selected inputs activate the bus. Operand A includes the scratchpad A0-A15, program counter P, program-status word PSW, initial program loader IPL, constant selector, and control-panel selector which supplies control-panel commands and system commands. Operand B is supplied by CPU working register M, shift register Q, and data selector C. The Q register is used as a shift register in double-length operations and as an auxiliary operand accumulator.

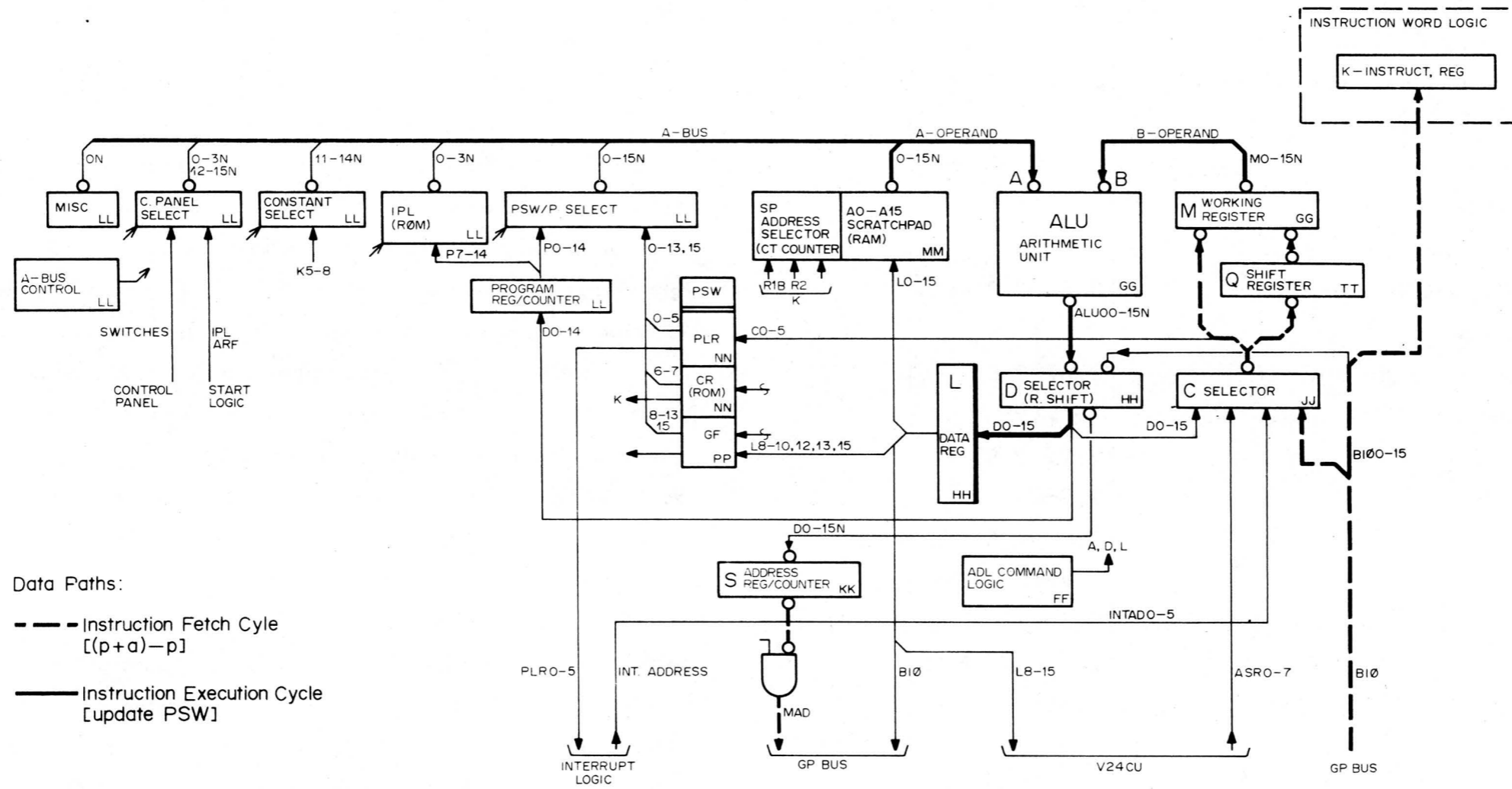
2.55 Data Path

Figure 2-4 shows the data path for instruction fetch operations (heavy dashed line) and instruction execution operations (heavy solid line). The fetch operation loads the instruction word into the instruction register K, and the eight least significant bits, via C, into the Q and M registers. The execution operation processes both operands through the ALU and outputs the result via D to the L register. Actual processing of data may be done by registers other than the ALU : the D selector and L and Q registers can perform shift operations in the data-path loop. With D selector at BIO direct, there is a direct loop from L, via BIO and D, to L. Scratchpad read, rewritten as same clock : path with D (exch. char.; shift right) and L (shift left).

2.56 A, D, L Command

The A, D, L command logic (Figure 2-8FF) controls the arithmetic operations by providing simultaneous command signals to the ALU, the D-selector, and the multiple-input L register. Thirty-two 8-bit command codes are stored in the ADL read-only memory (ROM). The command codes are selected by microinstruction bits μ ADL0-4 and some internal flags : Q15, Q16, FSIG, and M00N.

2.57 The first three address bits (μ ADL0-2) are used for direct ROM addressing (inputs E, D, C); they also control a dual 4-input multiplexer (type 9309) which selects the last two address bits (inputs B, A) of the ROM. ROM inputs B and A may be the last two microinstruction bits (μ ADL3-4, inverted)



Note: AA, BB etc. refers to logic diagrams 2-8AA, BB etc.

Figure 2-4 Data Handling Logic

or the flag bits Q15, Q16, FSIG, M00N, as follows :

Multiplexer Select Input		Multiplexer Output to ADL Command ROM	
μ ADL: 0, 1, 2	RALUA: 1N, 2N	RADL3	RADL4
0 0 0	1 1	μ ADL3N	μ ADL4N
0 0 1	1 1	μ ADL3N	μ ADL4N
0 1 0	1 1	μ ADL3N	μ ADL4N
0 1 1	1 1	μ ADL3N	μ ADL4N
1 0 0	1 1	μ ADL3N	μ ADL4N
1 0 1	1 0	Q15	Q16
1 1 0	0 1	μ ADL3N	FSIGN
1 1 1	0 0	μ ADL3N	DIVFLAGN

$$\text{DIVFLAG} = \text{FSIGN} \oplus \text{M00N}$$

Table 2-8 shows the complete microinstruction/flag control of the ADL ROM, the ROM contents, and the output functions.

2.58 The ADL-ROM outputs ALUS0N-3N are used as the ALU selection inputs S0-4 (Figure 2-8 GG). The ALUCO16 signal is sent to the ALU and the ALU look-ahead-carry unit as a carry control signal. The microinstruction bit μ ADL4 is also sent directly to the ALU CE input to select logic functions (CE-high) or arithmetic functions(CE-low). The ADL-ROM outputs DS0-1 (combined with microinstruction field bit GMULTI) are the control signals for the four-input D selector (Figure 2-8 HH). The D00DN signal that is generated as an auxiliary part of the ADL command logic is used as one input to the D selector, bit 0; this logic is described with the D selector. The ADL-ROM output LSEL is sent to the L register (Figure 2-8HH) to select direct load (LSEL-low) or shift left (LSEL-high).

Table 2-8 ADL Command ROM

MUADL 0 1 2 3 4	FLAGS	ADDRESS	ALU Selection				MNEMO	FUNCTION
			ALUS0N	ALUS1N	ALUS2N	ALUS3N		
0 0 0 1 1		00	0 0 0 0 0 1 1 0			ALUA	Operand A	
0 0 0 1 0		01	0 0 1 0 0 1 1 0			AOB	Logical OR of A and B	
0 0 0 0 1		02	1 0 1 0 0 1 1 0			ALUB	Operand B	
0 0 0 0 0		03	1 0 0 1 1 1 1 0			AMB	Arithmetic Subtract A minus B	
0 0 1 1 1		04	0 1 1 0 0 1 1 0			AXB	Logical XOR of A and B	
0 0 1 1 0		05	0 1 1 0 0 1 1 0			APB	Arithmetic Addition A plus B	
0 0 1 0 1		06	0 1 0 1 0 1 1 0			BINV	Operand B Inverted	
0 0 1 0 0		07	0 1 1 0 1 1 1 0			APLBI	Arithmetic Addition with Carry A plus B plus 1	
0 1 0 1 1		08	0 0 0 0 0 0 1 0			ACR	Operand A characters crossed on Selector D	
0 1 0 1 0		09	0 0 1 1 1 1 1 0			ZERO	ALU output is zero	
0 1 0 0 1		0A	1 0 1 0 0 0 1 0			BCR	Operand B characters crossed on Selector D	
0 1 0 0 0		0B	0 1 1 1 1 1 1 0			AANDB	Logical AND of A and B	
0 1 1 1 1		0C	0 0 0 0 0 1 0 0			ASHR	Operand A shifted right one position	
0 1 1 1 0		0D	0 0 0 0 0 0 0 0			DBIO	BIO Receivers selected on D	
0 1 1 0 1		0E	1 0 1 0 0 1 0 0			BSHR	Operand B shifted right one position	
- - - - -		0F	1 1 1 1 1 1 1 1			-	Not used	
- - - - -		10	1 1 1 1 1 1 1 1			-	Not used	
1 0 0 1 0		11	1 1 0 0 0 1 1 0			TWOA	Operand A added to itself	
1 0 0 0 1	Q15 Q16	12	0 0 0 0 0 1 1 1			ASHL	Operand A shifted left one position	
1 0 0 0 0		13	1 1 0 0 0 1 1 1			FORA	Operand A four lines (two in ALU and SAR)	
1 0 1 0 0	00	14	0 0 0 0 0 1 0 0				Operand ASHR	
1 0 1 0 0	01	15	0 1 1 0 0 1 0 0		MULTI		Operands A plus B and SHR	
1 0 1 0 0	10	16	1 0 0 1 1 1 0 0				Operand A minus B and SHR	
1 0 1 0 0	11	17	0 0 0 0 0 1 0 0				Operand ASHR	

MUADL 0 1 2 3 4	FLAGS	ADDRESS	ALUS0N ALUS1N ALUS2N ALUS3N ALUCO16 DS0N DS1N LSEL	MNEMO	FUNCTION
1 1 0 1 0	1	1 8	0 1 1 0 1 1 1 0	DADD	Arithmetic addition A plus B plus 1
1 1 0 1 0	0	1 9	0 1 1 0 0 1 1 0		Arithmetic addition A plus B
1 1 0 0 0	1	1 A	1 0 0 1 0 1 1 0	DSVB	Arithmetic subtract A minus B minus 1
1 1 0 0 0	0	1 B	1 0 0 1 1 1 1 0		Arithmetic Subtract A minus B
1 1 1 1 0	1	1 C	0 1 1 0 0 1 1 1	DIVSH	Addition and SHR 2 (A+B) + Q0
1 1 1 1 0	0	1 D	1 0 0 1 1 1 1 1		Subtract and SHR 2 (A-B) + Q0
1 1 1 0 0	1	1 E	0 1 1 0 0 1 1 0	DIVALU	Addition A + B
1 1 1 0 0	0	1 F	1 0 0 1 1 1 1 0		Subtract A - B

2.59 ALU -- Arithmetic Unit

The arithmetic unit (Figure 2-8 GG) uses four 4-bit ALU chips (type 74181) to perform arithmetic and logic functions on two 16-bit operands. The ALU chips are controlled by microinstruction bit μ ADL4 and the ADL-command logic (using bits μ ADL0-4). Bit μ ADL4 is applied to the CE input to select arithmetic (CE-low) or logic (CE-high) functions. All control bits are active high, while the data in and out (in this application) are active low. The ALU functions are shown in the following table. The control-signal logic, and the complete arithmetic unit functions combined with the D and L register operations, are discussed in the previous section (A, D, L Command) and shown in Table 2-8.

μ ADL4	ADL-Control ROM					Operation	
	ALUCO16	ALUSO-----3					
CE	CIN	S3	S2	S1	S0		
0	0	1	0	0	1	A plus B	ARITHMETIC
0	0	1	0	1	1	A or B	
0	0	1	1	0	0	A plus A	
0	1	0	0	0	1	A and B	
0	1	0	0	1	1	Zero	
0	1	0	1	1	0	A minus B	
1	-	0	1	0	1	B inverted	LOGIC
1	-	1	0	0	1	$A \oplus B$	
1	-	1	0	1	0	B	
1	-	1	1	1	1	A	
1	-	1	0	0	1	$A \oplus B + 1$	

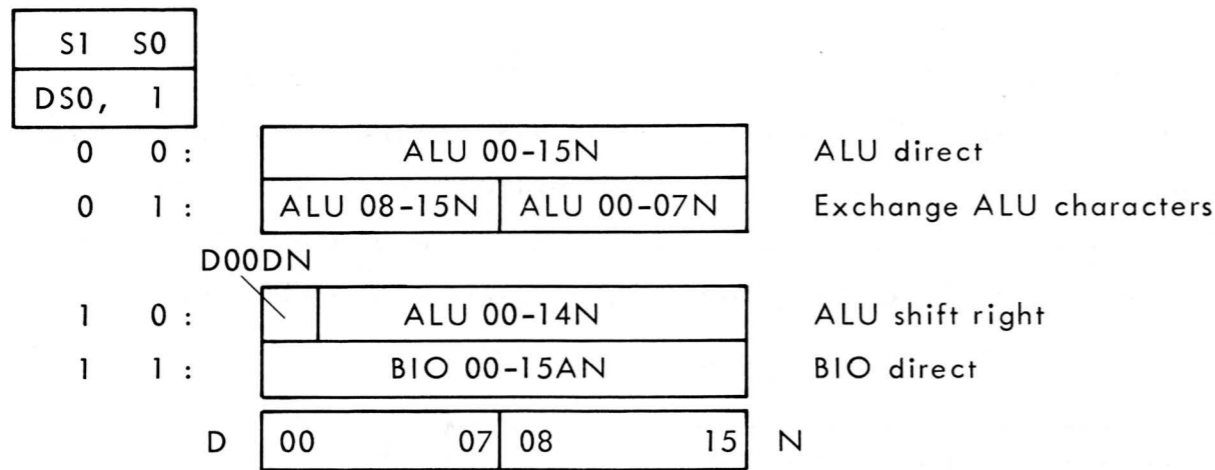
Control signals are all active high. Functions are for active-low data in and out.

A carry-look-ahead unit (type 74182) is used in conjunction with the ALU chips to provide anticipated carry across all four ALU circuits.

2.60 The zero-ALU (0ALUN) output is generated in subtract mode when both operands are equal (A=B). A zero-ALU, left-half (0ALUL) output sets the stack overflow indicator (STOV). Since stack operations forbid a memory address equal-to or less-than 128 (100 hexadecimal), the left-most character of the address (bits 0-7) must have at least one bit set. When S is loaded from the ALU during a stack operation, the all-zero ALU output (A=B) indicator STOV is therefore gated by GFSTOV to set GF flip-flop PIF (Figure 2-8 PP).

2.61 D Selector

The D selector (Figure 2-8 HH) is used as a control element in the arithmetic loop. The four D-selector functions (ALU direct, exchange ALU characters left and right, ALU shifted right, and BIO direct) are selected by the ADL-command logic (Table 2-8). The ADL-command signals DS0, DS1 operate the D selector as follows :



The D selector comprises eight type 9309 (dual 4-input multiplexer) circuits. These circuits provide both the true and complementary outputs of the active-low inputs. The inverted D-selector output (active-high) is sent to the L register, C selector, and P register/counter. The true output (active low) is sent to the S register/counter.

2.62 The ALU shift-right mode is used for shift-right and multiplication operations. The input bit D00DN used during this mode is provided by the ADL command logic (Figure 2-8 FF). Shift Right Arithmetic (SRA) and Shift Right Normalize (SRN) instructions use the ALU sign bit, ALU00. The Double Shift Right Circular (DRC) instruction uses Q15 for D00DN. Multiplication instructions set D00DN directly when negative overflow is detected (NOVF), or use the ALU sign bit ALU00 when positive overflow is not detected (POVF). The D00DN selection is shown in the following table.

GCRDSR.K9.ALU00	Shift Right, Arithmetic or Normalize (SRA, SRN)
GCRDSR.K08Q15	Double Shift Right Circular (DCR)
μADL0.NOVF	Multiply and negative overflow
μADL0.P0VN.ALU00	Multiply and positive overflow

2.63 L -- Data Register

The L register (Figure 2-8 HH) is a 16-bit, multiple-input register comprised of four type 74298 chips. The L register is used for storing the data from the ALU (or BIO) to be output to the scratchpad (A0-15) or the GP bus. The input

data to the L register is provided by the D selector. The L register is in the operand-A loop between the ALU and the scratchpad. Since the L register is loaded by the leading-edge of BP, an operand can be read from and re-written into the scratchpad on the same clock cycle. The two inputs to the L register are D direct and D shifted left. These two operating modes for L are selected by the ADL command logic (Table 2-4) bit LSEL :

LSEL	Data Source	Function
0	D 00-15	D direct
1	D 01-15 Q00	D shifted left
L	00 ————— 14 15	

Bit Q00 is shifted into the least-significant position during a Division instruction when the new remainder is loaded.

The L register stores active-high data with no inversion between input and output.

2.64 M -- CPU Working Register

The M register (Figure 2-8 GG) is a 16-bit, multiple-input register comprised of four type 74298 chips. The M register is used as the CPU working register in the operand-B data path. Microinstruction control of the register uses bits μMLOAD and μMSEL to load the register and to select the C selector or Q shift register for inputs, as follows :

Enable Clock	Select C or Q	Mnemonic	Data Source
μMLOAD	μMSEL		
0	-	No Op	Off; previously stored data available at output.
1	0	MYC	C 00 15
1	1	MYQ	Q 00 15
			M 00 15

The M register stores active-low data with no inversion between input and output. The μLOAD signal enables the CLMN input to the clock to store the selected data on the leading edge of BP.

2.65 C Selector

The C selector (Figure 2-8 JJ) uses two, quad 2-input multiplexers (type 8234) for the bit 0-7 selection, and four, dual 4-input multiplexers (type 9309) for the bit 8-15 selection. The result is a four-input multiplexer with two 16-bit sources and two 8-bit sources. Three microinstruction selection bits are used in such a manner that the two 16-bit sources (D and BIO) can be selected whole or as short constants (bits 8-15 only), or either of the 8-bit sources (INTAD and ASR) can be selected. The sources are selected by microinstruction bits $\mu C0$, $\mu C1$, and GCSLEN as follows:

$(\mu C0 + \overline{\mu C1})$ CLEFS0	$\mu C0$		$\mu C1$		Mnemonic	Source
	00-07		08-15			
	S0	S1	S0	S1		
1	0	0	1	CALU	D00 _____ 15	
0	0	1	1	CBIO	BIO00 _____ 15	
1	1	0	1	CLUR	D08 _____ 15	
1	1	1	1	CIOR	BIO08 _____ 15	
1	1	0	0	CLUR	INTAD0 _____ 5 0 0	
1	1	1	0	CIOR	ASR0 _____ 7	
1	0	0	0	(CALU)	D00 _____ 07 INTAD0 _____ 5 0 0	
0	0	1	0	(CBIO)	BIO00 _____ 07 ASR0 _____ 7	
C						
00N _____ 07N		08N _____ 15N				

The last two possibilities are not used.

2.66 The C selector inverts all selected data from active high to active low. The 0's supplied for the not-used bit positions are therefore output as inactive highs. The C-selector outputs are sent to the Q and M registers in the operand-B data path. The six least-significant bits (C00N-05N) are also sent to the priority level register during an interrupt routine.

2.67 Q Shift Register

The Q register (Figure 2-8 JJ) uses four type 74194 circuits as a 16-bit, left/right shift register. Q may be parallel-loaded from the C selector, shifted right (with ALU15N as a serial input), or shifted left (with the selected bit, Q15DN, as a serial input). Loading and shifting is done on the leading-edge of clock pulse BP (BPQ) according to the function selected by microinstruction bits $\mu Q0$, $\mu Q1$ as follows :

$\mu Q0, \mu Q1$	Mnemonic	Function
0 0	QI	Hold contents (No Op)
0 1	SLQ	Shift Left Q
1 0	SRQ	Shift Right Q
1 1	QYC	Parallel load Q from C

In the operand-B arithmetic loop, Q can be loaded from the M register, via ALU--D--C, and reloaded into the M register on the same BP clock pulse. No Q-register shifting may be performed on a single-clock-pulse loop since Q must be in the parallel-load mode.

2.68 The shift-right serial input, ALU15N, is used for double-length, shift-right operations. The shift-left serial input, Q15DN, is used for double-length left circular (DLC) shift and Divide instructions. Q15DN is derived from a 74151A circuit by the selection-inputs K08N, M00N, and ALU00N. For DLC shift instructions (K08N active low), the serial input is provided by microinstruction bit GCSELN if M00N is active; the serial input is a logic zero (high) if M00N is inactive (high). For Divide instructions (K08N high and RALUA1N ($\mu ADL 0.1$) = 0), Q15DN is produced from the quotient bit ALU00 \oplus M00. See following diagram.

K08N	M00N	ALU00N		
S2	S1	S0	Q15DN =	
L	L	L	GCSELN	DLC
L	L	H		
L	H	L		
L	H	H		
H	L	L	(μ ADL0.1)	Divide
H	L	H	H	
H	H	L	H	
H	H	H	(μ ADL0.1)	

L = low = active

2.69 The two least-significant bits of Q are used to control the next-cycle operation of a Multiply execution. One bit position (Q16) of a fifth 74194 is used to save the previous contents of Q15 at the end of the Multiply cycle, while Q14 is shifted into Q15. Q00, Q15N, and Q16N are used by the ADL command logic (Figure 2-8 FF) for arithmetic operation control. The three other bit positions of the fifth 74194 are used in the Bus Controller logic (Figure 2-8 TT). This fifth 74194 is used in the Hold (GCRFNUN = 0) or Parallel-Load (GCRFNUN = 1) modes only.

2.70 Q-register bit Q00/Q00N is used for microinstruction addressing in the Flag mode (paragraph 2.28 and Figure 2-8 DD). Q00 is also used as a serial input to the L register during a left-shift L operation.

2.71 S -- Address Register/Counter

The S register/counter (Figure 2-8 KK) is a 16-bit synchronous up/down counter using four type 74S169 circuits. Fourteen bits of the S register/counter are used to store the memory word address, and increment or decrement the address in one-word steps. Loading or counting takes place on the leading edge of the BP pulse, according to the following control from microinstruction bits μ S0 and μ S1:

μ S0, S1	Mnemonic	Function
0 0	SI	Inhibit counting or loading
0 1	SYD	Load S0-16N with D0-15N
1 0	SP2	S plus 2 (count down)
1 1	SM2	S minus 2 (count up)

Since S is loaded with negative (low-level active) data, the count-down mode increments the value of the contents and count-up decrements. The S register/counter is loaded from the D selector, as shown in the following diagram:



D00-14, representing the memory word address, is loaded into the least-significant position of S, while D15, representing the character address, is loaded into the most-significant position. S-register counting then affects a two-character address change (S plus 2, or S minus 2) for a single increment or decrement of the counter. The character indicator from the most-significant position in S is then switched back to the least-significant position on the memory address lines (MAD00-15). The S register/counter works with low-level data, and does not invert.

2.72 A-Bus Selection

The A-Bus selection logic determines which source is to be used as operand-A of the arithmetic operation. Control of the A-bus selection is by microinstruction bits μ A0-4 (Table 2-9). The possible sources for operand-A on the A-bus are:

- Scratchpad A0-A15
- Initial program loader IPL
- Program counter P

- Program status word PSW
- Instruction field (R1 or R3)
- Constant 2_{10} (/002) or 16_{10} (/010)
- Control-panel/System
- A-bus equals zero

All operand-A sources onto the A-bus are via active-low, open-collector output circuits (8234 multiplexers, 6200 ROM, 7489 RAM, and one 7403 gate). All these circuits provide a high-level output to the bus when they are not selected. This high is then over-ridden by any active low signal from a selected source.

2.73 The scratchpad (Figure 2-8 MM) is selected directly by $\mu A0$ to the chip-select inputs of the scratchpad logic. The specific scratchpad address (A0 to A15) is then selected by $\mu A2-4$ controlling the scratchpad-address logic. When the scratchpad is selected, $\mu A1$ differentiates between reading and writing the selected address.

2.74 Any A-bus source except the scratchpad is selected by the A-bus control logic (Figure 2-8 LL). The inverted $\mu A0$ input to the 1-of-10 decoder chip (type 9301) forces the decoder higher than the seven possible outputs that are connected; this effectively keeps this part of the A-bus selection switched off when the scratchpad is selected. If $\mu A0$ is active (high), and $\mu A1-3$ are all low, the non-connected output-0 is decoded and no A-bus source is selected; operand-A is then equal to zero. The active-low outputs from the 9301 decoder select either the IPL (output-1) or one of the type 8234 multiplexers. These multiplexers have their A inputs selected by an active low to S1 only, or their B inputs selected by an active low to S0.

2.75 Double-Word Trap. If a Trap occurs during a double-word operation, (P) must be decremented an extra position. The DWIF flip-flop (Figure 2-8 LL) is set for all double-word instructions. If a Trap routine (Figure 1-12) is initiated, the first microinstruction (address /OFF) produces GAEXL which is gated by DWIF to set A-bus bit A00N. This increments the normal microinstruction address by one to select /12E rather than /12F.

Table 2-9 A-Bus Selection

Microinstruction $\mu A0, A1, A2, A3, A4$					Control decoder (9301) output active	Mnemonic	Function
0	0	0	0	0		ARA0	Read scratchpad accumulator : A0
0	0	0	0	1		ARA1	A1
0	0	0	1	0		ARA2	A2
0	0	0	1	1		ARA15	A15
0	0	1	0	0		ARR1	R1 (Addressed by K5-7,8)
0	0	1	0	1		ARR2	R2 (Addressed by K11-14)
0	0	1	1	0		AR1215	(Addressed by K12-15)
0	0	1	1	1		ARCT	(Addressed by CT counter)
0	1	0	0	0		AWA0	Write scratchpad accumulator : A0
0	1	0	0	1		AWA1	A1
0	1	0	1	0		AWA2	A2
0	1	0	1	1		AWA15	A15
0	1	1	0	0		AWR1	R1 (Addressed by K5-7,8)
0	1	1	0	1		AWR2	R2 (Addressed by K11-14)
0	1	1	1	0		AW1215	(Addressed by K12-15)
0	1	1	1	1		AWCT	(Addressed by CT counter)
1	0	0	0	-	--	AZ	A-bus equals zero
1	0	0	1	-	1	AIPL	Read IPL
1	0	1	0	-	2	AEP	Read P
1	0	1	1	-	3	APSW	Read PSW
1	1	0	0	-	4	AEN	Read content of K5-8
1	1	0	1	1	5	ATW0	Constant 2_{10} (/002) onto A-bus
1	1	0	1	0	5	ATEN	Constant 16_{10} (/010) onto A-bus
1	1	1	0	-	6	APUP	Control panel inputs onto A-bus
1	1	1	1	-	7	ASYS	System inputs onto A-bus

2.76 IPL -- Bootstrap

The IPL circuit is a 64-word hardware bootstrap stored in a type 6200 Read Only Memory (ROM) of 256 four-bit words. The ROM contents are regrouped into 16-bit words when they are loaded into main memory by the IPL-microprogram routine (Figures 1-9, 1-10). The IPL/Bootstrap logic (Figure 2-8 LL) is addressed by the P-register/counter (bits P07-14) and is accessed through the A-bus. The A-bus-selection command code is $\mu A0, 1, 2, 3 = 1\ 0\ 0\ 1$ for the read-IPL routine. Table 2-10 is a listing of the bootstrap after being loaded into memory 00-63. Use of the bootstrap for initial program loading is discussed in Section III.

2.77 P -- Program Register/Counter

The P register/counter (Figure 2-8 LL) is a 16-bit synchronous up/down counter using four type 74S169 circuits. The fourteen least-significant positions of P are loaded from the D selector with the 14-bit word address of the program. The P register is also used as an internal counter: in this case, the most-significant bit (which isn't loaded from D) is sent to the flag-selection logic as PM1 (P minus 1). PM1 is used by microinstruction Addressing, Flag mode (paragraph 2.26) to indicate the end of a multiple word instruction.

2.78 Loading or counting of P takes place on the leading edge of the BP pulse (BPP) according to the following control from microinstruction bits $\mu P0$ and $\mu P1$:

$\mu P0, P1$	Mnemonic	Function
0 0	PI	Inhibit counting or loading
0 1	PYD	Load P0-14 with D0-14
1 0	PM2	P minus 2 (count down)
1 1	PP2	P plus 2 (count up)

The program word-address in P is accessed through the A-bus. The A-bus selection code $\mu A0, A1, A2, A3 = 1010$ for read-P. The eight least-significant address bits (P07-14) are also applied directly to the addressing inputs of the IPL ROM, for use when the IPL is selected.

Table 2-10 List of Bootstrap Loaded in Memory

Step	P Reg.	Mem. Cont.	Instruct.
000			IDENT BEBOOT
001			*
002			*
003			DISPLAY THE KEYS AS FOLLOWS:
004			*
005			BITS MEANING
006			0=1 IPL LOADED FROM ASR, FORMAT 4*4
007			1=1 DISK, THEN
008			2=1 MOVING HEADS
009			2=0 FIXED HEADS
010			3=1 I/O BUS
011			3=0 MULTIPLEX
012			4 to 7 BOU LINE (4 RIGHTMOST BITS)
013			8=1 MULTI DEVICE CONTROLLER
014			8=0 SINGLE DEVICE CONTROLLER
015			9=1 CDD DISK
016			10 to 15 DEVICE ADDRESS
017			*
018			*
019			*
020			DESCRIPTION
021			*
022			BEBOOT LOADS ONE RECORD ONTO LOCATION /80 THEN START AT /84
023			THE RECORD IS THE SECTOR # 1 IF DISK, OR 254 CHARACTERS OF THE
024			INPUT DEVICE, LEADING NULL CHARACTERS IGNORED
025			*
026			*
027			*
028			USED REGISTERS:
029			*
030			A1 BOU LINES, NOT TO BE DESTROYED IF BOOT IS CALLED AGAIN
031			A2 ADDR OF INR INSTRUCTION
032			A3 ADDR OF CIO INSTRUCTION (WHICH IS DESTROYED AND NEEDS TO BE
033			RESTORED IF BOOT IS CALLED AGAIN)
034			A4 ADDR OF SST INSTRUCTION
035			A5 MULTIPLEX: CONTENTS OF 1ST WORD TO BE SENT TO EXT REGISTER
036			I/O BUS: CHARACTER COUNT, INITIALIZED AT 254 AND DECREMENTED
037			A6 MULTIPLEX: CONTENTS OF 2ND WORD TO BE SENT TO EXT REGISTER
038			(LOADING ADDR)
039			I/O BUS: ADDR OF NEXT CHAR TO BE LOADED, INIT AT /80 AND
040			INCREMENTED
041			A7, A8 WORK REGISTERS
042			A9 WORK REGISTER
043			A10 TO A14 NOT USED
044			A15 CONTAINS THE KEYS'VALUE
045			*
046			*
047			*
048			*
049			*
050			*
051			28/8/73 ARE FIXED:
052			MULTIPLEX DBLE WORD INITIALIZATION
053			SST INST FOR MULTIPLE DEVICE CONTROLLER
054			*
055			EJECT
056			ADRG 0
057			*
058			*
059			BOOT EQU *
060			INITIALIZE REGISTERS
061	0000	0200	F LDK A2, INR ADDR OF INR INSTRUCTION
062	0002	0300	F LDK A3, CIO ADDR OF CIO INSTRUCTION
063	0004	0400	F LDK A4, SST ADDR OF SST INSTRUCTION
064			* EXTRACT DEVICE ADDR AND INIT I/O COMMANDS
065	0006	861E	LDR A6, A15
066	0008	263F	ANK A6, /3F DEVICE ADDR
067	000A	9629	ADRS A6, A2 INITIALIZE I/O INSTRUCTIONS
068	000C	962D	ADRS A6, A3
069	000E	9641	ADS A6, HIO
070	0010	0000	F -
071	0012	871E	LDR A7, A15 EXTRACT CONTROLLER ADDR AND INIT WER INST
072	0014	3FC8	SLC A7, 8 MULTI OR SINGLE DEVICE CONTROLLER
073	0016	5600	F RF(6) INIT20 SINGLE ONE
074	0018	260F	ANK A6, /F MULTIPLE ONE
075			INIT20 EQU * INITIALIZE MULTIPLEX DBLE WORD:
076	001A	9631	ADRS A6, A4 SST INSTRUCTION
077	001C	3E41	SLL A6, 1
078	001E	9641	ADS A6, WER1 SET UP WER INSTRUCTIONS
079	0020	0000	F
	0022	9641	ADS A6, WER2
	0024	0000	F

Step	P Reg.	Mem. Cont.	Instruct.		
080			*		
081	0026	811C	LDR	A1,A7	LOAD A1 WITH BOU CONTENTS
082	0028	0550	LDK	A5,80	
083			*		MULTIPLEX DOUBLEWORD: LOAD 80 CHAR INTO LOCATION /80
084	002A	0680	LDK	A6,/80	
085			*		CHECK IF DISK
086	002C	3FE7	SRC	A7,7	
087	002E	5600	F RF(6)		NODISK
088			*		NO FIXED HEADS ?
089	0030	3FC1	SLC	A7,1	
090	0032	5600	F RF(6)		NOSEEK
091	0034	0103	LDK	A1,3	YES
092	0036	41C0	CIO	A1,1,0	SEEK ZERO
093			NOSEEK EQU	*	CIO SEEK ZERO
094	0038	811E	LDR	A1,A15	
095	003A	3966	SRL	A1,6	SECTOR NUMBER
096	003C	213C	ANK	A1,/3C	
097	003E	8520	LDKL	A5,/80CD	1ST WORD OF MULTIPLEX FOR DISK DEVICE
	0040	80CD			
098			NODISK EQU	*	
099			*		EXECUTE WER, WHATEVER THE CHANNEL IS
100			WER1 EQU	*	
101	0042	7500	WER	A5,0	
102			WER2 EQU	*	
103	0044	7601	WER	A6,1	
104	0046	F031	EXR*	A4	SST
105	0048	F02D	EXR*	A3	
106	004A	5C06	RB(4)	* -4	
107	004C	871E	LDR	A7,A15	
108	004E	3F43	SLL	A7,3	
109	0050	5600	F RF(6)		MULTIPLEX
110			*		
111			*		IO BUS
112			*		
113	0052	8194	LDR	A9,A5	IF A9=A5 IGNORE LEADING CHAR.
114			INR EQU	*	
115	0054	4F00	INR	A7,0,0	READ ONE CHAR
116	0056	5C04	RB(4)	* -2	
117	0058	E994	CWR	A9,A5	LEADING CHAR?
118	005A	5400	F RF(4)		NO
119	005C	27FF	ANK	A7,/FF	CHECK IF NULL
120	005E	580C	RB(0)	INR	YES, IGNORE
121			*		NO, CHECK IF 4*4
122			*		
123			INR10 EQU	*	
124	0060	879E	LDR	A15,A15	4*4
125	0062	5600	F RF(6)		NO 8+8
126	0064	3F44	SLL	A7,4	YES, 4*4
127	0066	809C	LDR	A8,A7	SAVE LEFT BITS
128	0068	F029	EXR*	A2	READ NEXT CHARACTER
129	006A	5C04	RB(4)	* -2	
130	006C	270F	ANK	A7,/F	GET 4 RIGHT MOST BITS
131	006E	9702	ADR	A7,A8	
132			STORE EQU	*	
133	0070	E739	SCR	A7,A6	STORE CHAR
134	0072	1601	ADK	A6,1	NEXT CHAR ADDR
135	0074	1D01	SUK	A5,1	COUNT DONE ?
136	0076	5924	RB(1)	INR	NO
137			*		YES
138			*		
139	0078	4180	HIO	CIO	A1,0,0
140			*		
141			STATUS EQU	*	
142	007A	4FC0	SST	A7,0	
143	007C	5C04	RB(4)	* -2	
144	007E	0F84	AB	/84	
145			*		
146			*		
147			*		
148			END	BOOT	

SYMBOL TABLE

BOOT	0000	A	INR	0054	A	CIO	0036	A	SST	007A	A
HIO	0078	A	INIT20	001A	A	WER1	0042	A	WER2	0044	A
NODISK	0042	A	NOSEEK	0038	A	INR10	0060	A	STORE	0070	A
STATUS	007A	A									

ASS.ERR. 00000

:EOF

2.79 A0-A15 -- Scratchpad

These sixteen 16-bit accumulators comprise 15 working registers (A0-A14) and the stack pointer (A15). The working registers are used as an operand for some instructions. Fifteen registers (A1-A15) are program addressable. The scratchpad (Figure 2-8 MM) is controlled by the A-bus selection commands, μ A0-4. The complete command code for scratchpad addressing and function is given in Table 2-9. For the write-scratchpad modes, the contents of the L register (L00-15) are loaded in the selected address (A0-A15). For read-scratchpad modes, the inverted contents of the selected address are gated onto the A-bus.

2.80 The 16-word scratchpad comprises four 64-bit read/write memory chips, type 7489. These circuits provide data inversion (active high input, low output) and open-collector outputs with a high-level off-state when μ A0 to the chip-select input is 1 (scratchpad not selected). The scratchpad write-enable input, WSPN, from the start logic (Figure 2-8 RR), is active from the center of clock time T6 to T7, if μ A0,A1 = 01.

2.81 The four-bit scratchpad address (SPA0-3) is selected through four type 74151A eight-input multiplexers. The first four sets of inputs are wired to +5 or 0 volts to obtain the direct address code of A0, A1, A2, or A15. The inputs 4, 5, or 6 select different fields of the instruction word to address the scratchpad: the R1/R3 field is provided by buffered K05-08/K05-07 (K08 is given via K0008BUFN if K00 = 1, indicating format-1 instruction); the R2 field is provided by K11-14N; the K12-15 field gives the number of shifts (in complement form). The seventh input selects the CT counter for addressing successive scratchpad registers.

2.82 The five-bit CT counter (Figure 2-8 MM) is used to count repeat cycles for the sequensor control, or (for P857) to address sequential scratchpad registers during Multiple Load or Store instructions. The counter comprises a type 74161 BCD counter for the four least-significant bits (CT0). A buffer at the output of the counter maintains the scratchpad address while the counter is being incremented. The buffer copies CT at the trailing edge of each AP pulse.

Counter operation is controlled by microinstruction bit μ CT and general-field bit GCTLDN, as follows :

μ CT (H=1)	GCTLDN (L=1)	Function
0	0	No change
0	1	Load CT1-4 from SPA0-3; CT0 from K11
1	0	Count CT at each T3N

The counter is preset to 16 (CT0-4 = 10000) by KRYN during each instruction Fetch cycle. Other initial counts less than 16 may be loaded into CT1-4 via the scratchpad address selector.

2.83 The number of shifts to be performed are loaded in complement form from the K12-15 field and effectively decremented by the counter operation. The counter is loaded with (R2) for the Double Arithmetic instructions in T1D addressing mode, enabling the least-significant operand word to be processed first. For DAR and DSR instructions in T1D addressing mode, (R2) + 1 addressing is obtained by the CT counter providing the least-significant address bits to the scratchpad. Multiply and Divide instructions count from the preset count of 16. The CTEND signal is generated when the count reaches 31 (CT0-4 = 11111). CTEND is used by the Sequensor to end the repeat cycles.

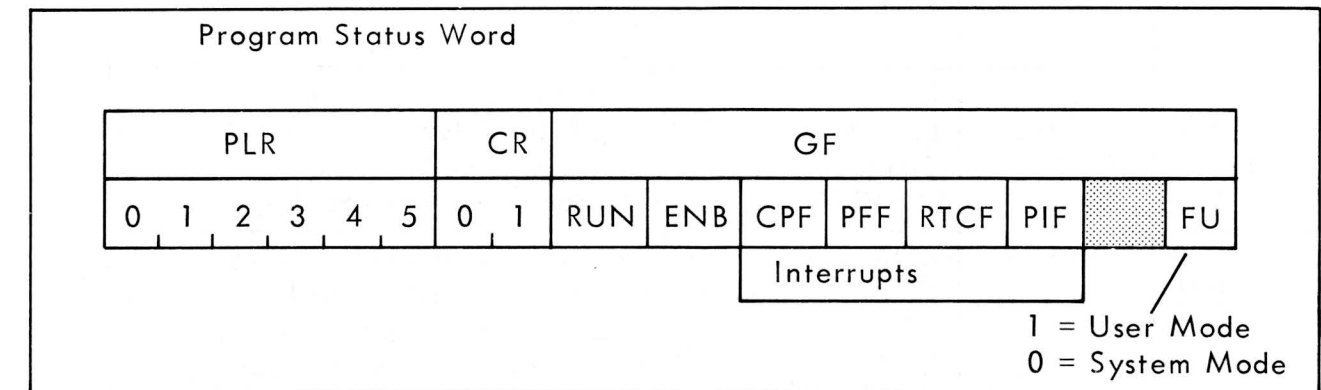
2.84 PSW -- Program Status Word

The PSW is a 16-bit status word comprising the priority level register (PLR), condition register (CR), and general flip-flops (GF). The complete PSW is saved in the stack, via the A-bus, during the following :

- Interrupt routine
- Trap routine
- Page Fault
- Call Function (CFR)
- Halt (HLT)
- Inhibit Interrupt (INH)
- Reset Internal Interrupt (RIT)
- Enable Interrupt (ENB)
- Link to Monitor (LKM)
- Set Mode (SMD)

During a Return (OPC14) instruction, System Mode (A15), the PLR, the CR, and

the ENB and FU-bits of GF are restored from the stack to the PSW; the other bits of GF are not changed by the Return. During a Return instruction, User Mode ($\overline{A15}$), only CR is restored from the stack.



RESTORED FROM STACK AFTER:	
RTN (A15)	RTN ($\overline{A15}$)
PLR	PLR
CR	
ENB	(ENB unchanged)
FU	

2.85 PLR -- Priority Level Register

The PLR (Figure 2-8 NN) is a 6-bit register that stores the priority level code of the running program for the Interrupt logic. This register, comprising two type 74175 circuits, is preset to priority level 63 by the system master-clear signal MCLN. The register output, PLR0-5, is used by the Interrupt logic (paragraph 2.97) for comparison with Interrupt Requests. If an Interrupt Request is accepted, the Interrupt Routine is initiated : the contents of the PLR are saved in the stack with the rest of the program status word, and the PLR is loaded with the new program's interrupt level by INTAD0-5 via the C selector.

2.86 CR -- Condition Register

The 2-bit condition register (Figure 2-8 NN) indicates the result of various CPU operations. The condition register flip-flops are updated at each BP clock time (BPQ). The CR input conditions (Table 2-11) are selected by a pair of type 74175A 8-input multiplexers, with the selection controlled by microinstruction bits μ CR0-2.

Table 2-11. CR Input Conditions

CR INPUT SELECTION				
Selection μCR0,1,2	Mnemonic	Input to CR		Function
		CR0	CR1	
0 0 0	CRNC	CR0	CR1	No change
0 0 1	CRRTN	BIO06R	BIO07R	CR load during Return
0 1 0	CRIO	Timeout	ACBN	Programmed channel loading
0 1 1	CRFLO	FLOCRO	FLOCR1	Floating Point loading
1 0 0	CRLOG	ROMCR1	ROMCR5	Logic operation loading
1 0 1	CRADD	ROMCR2	ROMCR6	Addition operation loading
1 1 0	CRSUB	ROMCR3	ROMCR7	Subtraction operation loading
1 1 1	CRCMP	ROMCR4	ROMCR8	Compare operation loading

CR INPUT ROM CODE

	ROM Addressing					ROM Output to CR Input Selector					
	0ALUL	0ALUR	ALU00N	A00N	M00N	LOGIC CR1,5	ADD CR2,6	SUB CR3,7	COMP CR4,8		
0	0	0	0	0	0	1	0	1	0	1	0
0	0	0	0	0	1	1	0	1	0	1	0
0	0	0	0	1	0	1	0	1	1	0	1
0	0	0	0	1	1	1	0	1	1	0	1
0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	0	1	0	1	1	0
0	0	0	1	1	0	0	1	0	1	0	1
0	0	0	1	1	1	0	1	0	1	0	1
0	1	0	0	0	0	1	0	1	0	1	0
0	1	0	0	0	1	1	0	1	0	1	0
0	1	0	0	1	0	1	0	1	1	0	1
0	1	0	0	1	1	1	0	1	1	0	1
0	1	1	0	0	0	0	1	1	0	1	0
0	1	1	0	0	1	0	1	0	1	0	1
0	1	1	1	0	0	0	1	0	1	0	1
0	1	1	1	0	1	0	1	0	1	0	1
0	1	1	1	1	0	0	1	0	1	0	1
0	1	1	1	1	1	0	1	0	1	0	1
1	0	0	0	0	0	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	0	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	1	1	0	1	0
1	0	1	0	0	1	0	1	1	1	1	0
1	0	1	1	0	0	0	1	0	1	0	1
1	0	1	1	0	1	0	1	0	1	0	1
1	0	1	1	1	0	0	1	0	1	0	1
1	0	1	1	1	1	0	1	0	1	0	1

not used

	ROM Addressing					ROM Output to CR Input Selector					
	0ALUL	0ALUR	ALU00N	A00N	M00N	LOGIC CR1,5	ADD CR2,6	SUB CR3,7	COMP CR4,8		
1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	0	0	1	1	1	1	1	1	1
1	1	1	0	1	0	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	1	1	0	0
1	1	1	1	0	1	0	0	0	1	1	0
1	1	1	1	1	0	0	0	0	1	1	0
1	1	1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1

not used

Input 0 copies the previous CR contents (no change). Inputs 1-3 select conditions for Return, programmed channel loading, and Floating Point loading.

2.87 Inputs 4-7 are selected for arithmetic operations, and the input conditions are provided by a type-7488A read only memory (ROM). One specific pair of the eight ROM output bits is selected for the type of arithmetic operation being performed (logic, addition, subtraction, or compare). The 2-bit condition codes are selected from one of 32 ROM addresses by : the sign bit of the two ALU operands, A00N and M00N; the sign of the result, ALU00N; and the zero contents of the result for both left and right characters, 0ALUL and 0ALUR.

2.88 An overflow condition, set into the OVF flip-flop, sets CR to 3 (11₂) by switching off the enable input to the ROM to produce an all-Ones output. The overflow condition is tested by two series-connected multiplexers (the 8-input 75141 and the quad 2-input 74157), as follows :

74157		74151					Function	
Input	Select A00	In	K01	ALU00	A01	Function		
(+5V) 1 μMLOAD	0 (I1) 1 (I0)	0 2	0 0	1 0	1 1	overflow OK	Shift	
μMLOAD (+5V) 1	0 (I1) 1 (I0)	1 3	0 0	1 0	0 0	OK overflow		
GCRDSRN OVFMUN	0 (I1) 1 (I0)	4 5	1 1	1 1	1 0	overflow OK	Divide	
(+5V) 1 GCRDSRN	0 (I1) 1 (I0)	6 7	1 1	0 0	1 0	OK overflow		

OVFMUN = GMULTI. $\overline{\text{ALUS0}}$.M00

The OVF signal fed back to the input-selection enable, prevents changing the overflow condition once an overflow has been set. The overflow flip-flop must be reset by command bit GCRVZ0N. The K01 input is 0 for Op-codes less than 8 to select shift instructions, or 1 for Op-codes of 8 or higher to select Multiply, Divide, or Double instructions. The shift instructions set overflow (via 74151A inputs 0, 3) on the test $A01N \oplus A00N$ to indicate a changing sign bit. The Divide instructions, specified by command bit GCRDSRN, set overflow (via 74151A inputs 4,7) on the test $\text{ALU00N} \oplus A00$, indicating that the quotient is greater than one word.

2.89 GF -- General Flip-Flops

These seven general control flip-flops (Figure 2-8 PP), which can be program controlled, are part of the program status word (PSW). All GF bits are stored in the stack along with the rest of the PSW, but only the FU (User Mode) is restored from the stack by the Return instruction.

GF	PSW Bit	Function	Loaded at CLG (set or reset)	Set by	Reset by
RUNF	8	Main CPU status pointer	L08	STARTF.T1	GFRZ0 +(PREQ. RUN.T1)+(FTDEL. RUN)+RSLF
ENBF	9	Enable interrupts	L09	GFENBN. $\overline{\text{MCLN}}$	GFSYSN
CPF	10	Operator's Interrupt	L10	(CPINT+V24INT).T1	MCLN
PFF	11	Power Failure Interrupt	$\overline{\text{M11}}$ resets	PWF $\overline{\text{J}}$	PWF.RSLF. UNLOCK
RTCF	12	Real Time Clock Interrupt	L12	RTCFZIN from C. Panel	MCLN
PIF	13	Op-Code (Program) Interrupt	L13	GSTOV.STOV from ALU	MCLN
FU	15	User Mode (0=System Mode)	CLPLR.C15	L15.CLG	GFSYSN+MCLN

Program control of the GF bits is by the A-bus command signal AEPSWN (at clock time T8). AEPSWN selects the program status word (with GF) to the A-bus and gates the L-register data back into the GF. The AEPSWN.T8 control signal CLG gates L08-10, 12, 13, 15 to the respective GF bits. GF bit 11 (PFF) is reset at that time if M11 = 0. Bit 11 uses the M-register rather than the L-register because of the required polarity. In addition to the program control of the GF bits, asynchronous set and reset inputs are provided to each flip-flop.

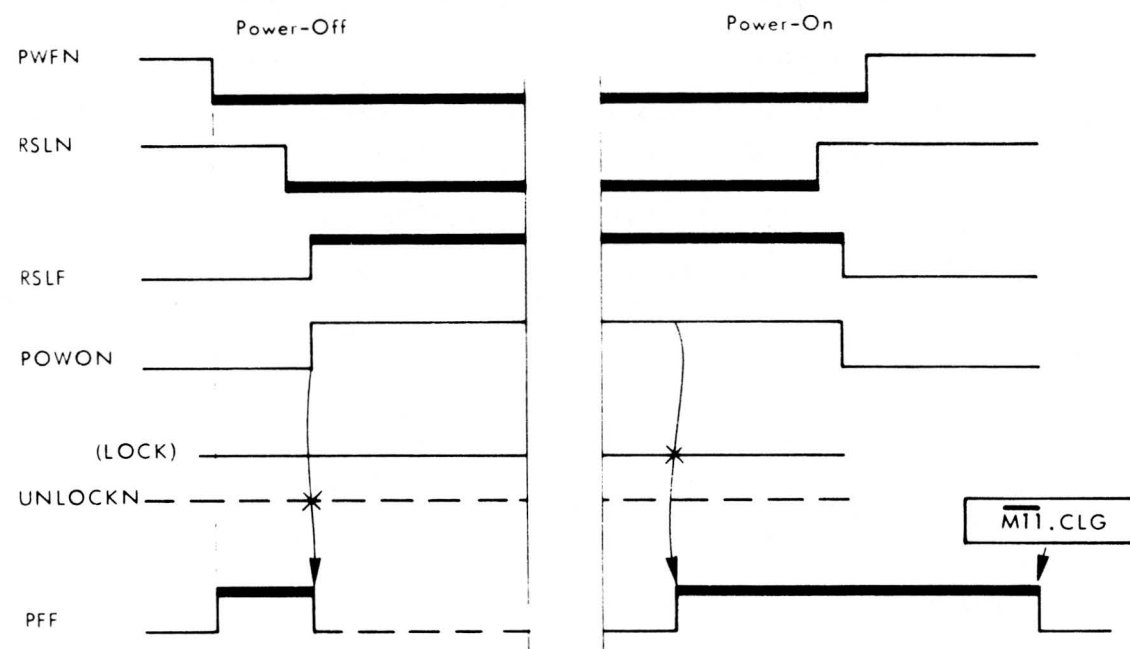
2.90 RUNF. This is the primary machine-state pointer (refer to Figure 1-8). The RUNF flip-flop is set at time T1 by the Start Command from the control panel. The RUNF flip-flop can be reset by any one of four ways :

- command bit GFRZ0;
- a Preset Address Compare from the control panel, at time T1;
- control-panel command Not Run, enabled by the buffered FETCH signal, FTDEL;
- RSLF which is set during the power-off sequence.

2.91 ENBF. This Enable-Interrupt flip-flop is set (with GFENBN) by the Enable Interrupt (ENB) instruction and reset (with GFSYSN) by the Inhibit Interrupt (INH) instruction. ENBF is also set or reset (with GFENBN) by the Return instruction when R2 = 15, and in function with ENB of the restored PSW.

2.92 CPF. This Operator's Interrupt can be set either by the INT switch on the control panel or by the interrupt signal BRGFN generated by the V24/Serial control unit. CPF is reset either by an RIT instruction with bit 0 = 0 (I10 = 0 during the CLG clock) or by the MCLN signal.

2.93 PFF. The Power Failure Interrupt flip-flop is set at the start of a power failure by the leading edge of the PWFN signal from the power supply; PFF is then reset during the power-off sequence if the key switch is set to any unlock position (OFF, ON, ONRTC). The Power Failure Interrupt flip-flop is set during the power-on sequence if the key switch is set to LOCK. PFF is reset by an RIT instruction with bit 11 = 0 (M11 = 0 during the CLG clock).



2.94 RTCF. The Real Time Clock Interrupt flip-flop is set by the RTCFZIN signal from the control panel. RTCFZIN is a one-microsecond signal generated once every 20ms by the power supply, and set via the control-panel key switch ONRTC or LOCK positions. The RTCF flip-flop is reset by an RIT instruction with bit 12 = 0 (L12 = 0 during the CLG clock) or by the MCLN signal.

2.95 PIF. The Operation-Code (Program) Interrupt flip-flop is set by a stack overflow condition or by the Link to Monitor (LKM) instruction to switch from User Mode to System Mode. Stack overflow (STOV from the arithmetic unit) sets PIF during memory stack operations (command bit GFSTOV) when the stack pointer decrements to less than 128_{10} . The LKM instruction sets PIF via the load input, with L13 = 1. The PIF flip-flop is reset by an RIT instruction with bit 13 = 0 (L13 = 0 during the CLG clock) or by the MCLN signal.

2.96 FU. The User Mode flip-flop is set during the Return A15 instruction if the previously-interrupted program was in User Mode (CLPLR.C15); the CLPLR signal loads the priority level register from the C-selector during the Return instruction, and C15 indicates the state of FU during the previous interrupt. The User Mode flip-flop is also set by the Set Mode (SMD) instruction (L15 = 1 during the CLG clock). FU is reset, to System Mode, by the Inhibit Interrupt (INH) instruction (with GFSYSN) or by the MCLN signal.

2.97 INTERRUPT LOGIC

The Interrupt logic (Figure 2-4 and 2-8 SS) generates the Scan External Interrupt signal (SCEIN), tests for internal and external interrupt requests, and initiates the Interrupt routine when a request is received of a higher priority than the running program. Refer to paragraph 1.16 for a description of the complete interrupt system. The Enable Interrupt control flip-flop, ENBF, is located with the general flip-flops (GF), along with the four dedicated internal interrupts : CPINTF, PFF, RTCF, and PIF.

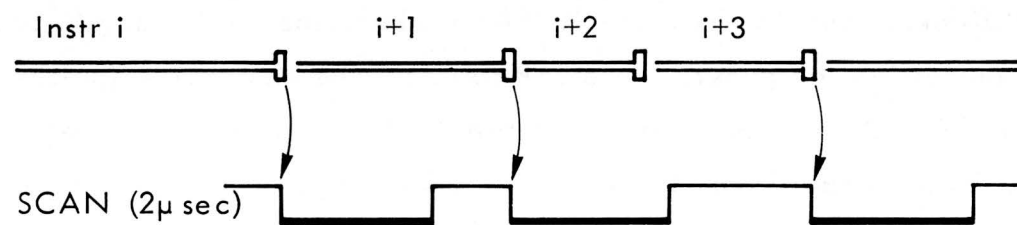
2.98 Internal Interrupts

Eight internal interrupt request lines (IS00-07N) are provided which have the highest system priority. At clock-time AP, all internal interrupt requests are

clocked into the two 74175 registers and are then available at the priority encoder 74148. If there are any internal interrupt requests waiting at the priority encoder, the low GS output switches the 74158 multiplexer to select the internal interrupts, and the high EO output (IECE) forces INTAD0,1 low (inactive). The highest-priority interrupt request is indicated by a 3-bit code at the 74148 A-outputs. The interrupt request code is switched through the 74158 multiplexer to the INTAD0-5 lines. The IECGSN signal is sent to the Start logic (Figure 2-8 RR) to enable setting the automatic restart flip-flop during the power-on/power-off sequences.

2.99 External Interrupts

A Scan External Interrupt (SCEIN) signal is generated during the status test at the end of each instruction, or during the Move instruction, if more than two microseconds have passed since the start of the previous scan. In order to validate the interrupt at the end of an LKM, ENB, INH, or SMD instruction, the microprogram does a 2.070 μ sec wait before ending those instructions.



The microinstruction control bits (Figure 2-8 SS) are NAETAN (μ SNA0.1) for status test, and GMOVEN for the Move instruction. Either of these inputs puts a high on the D input of the FETAT flip-flop, if there is no pending Interrupt Request (Not IR). At time T3, FETAT is set and the FETATN output triggers the SCEIT single-shot. SCEIT is inverted to provide the 2 μ sec SCEIN signal to the GP bus.

2.100 The BIECINH flip-flop is triggered by T5 90ns after SCEIT is started. The low BIECINH output is fed back to the OR'ed input of the single-shot to prevent SCEIT from being re-triggered. On the first T5 pulse after

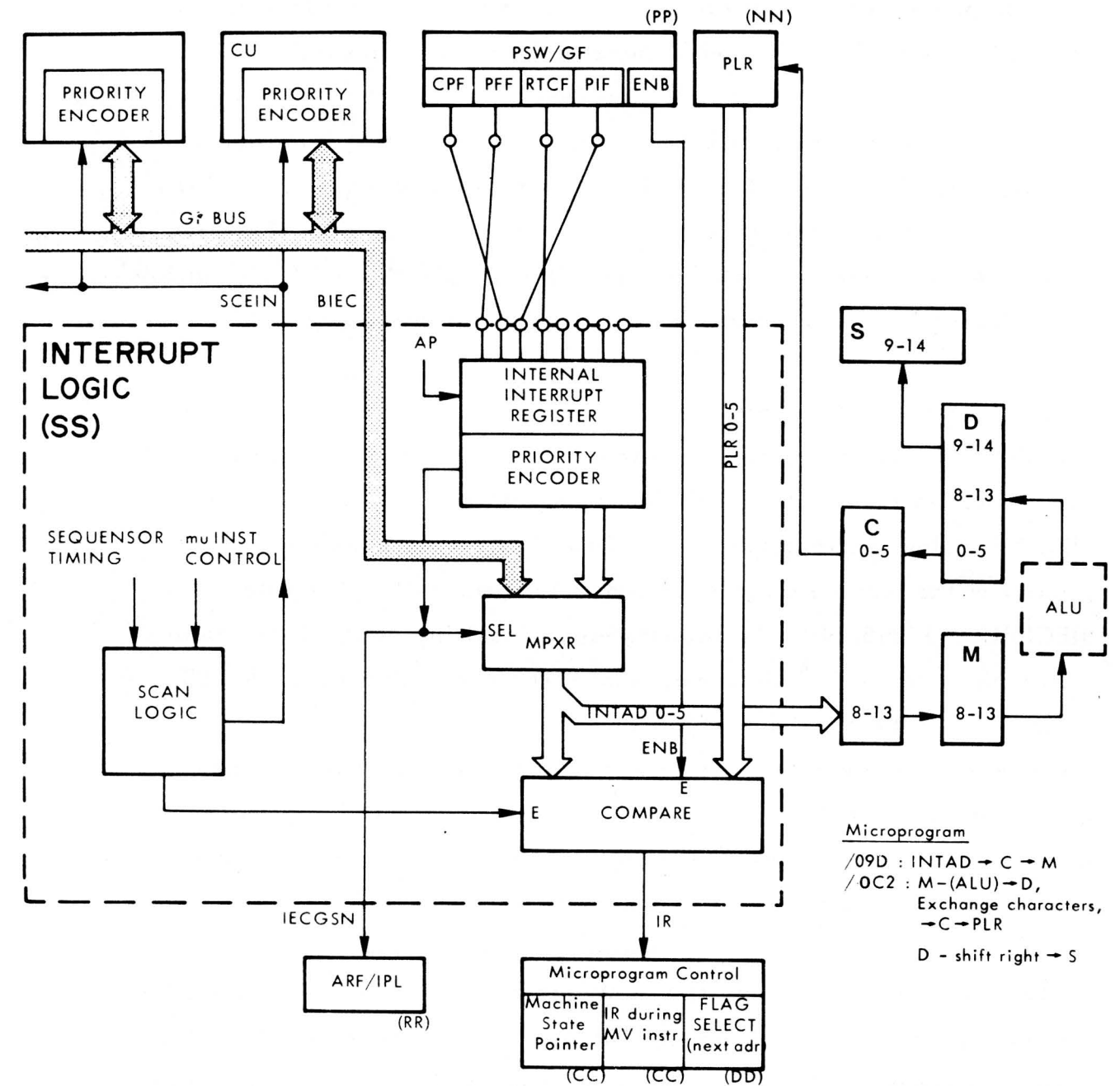


Figure 2-5 Interrupt System

the completion of the 2 μ sec SCEIT signal, BIECINH is reset, and the SCEIT single-shot is prepared for the next microinstruction-controlled start of a scan signal. The high BIECINH output applied to the most-significant input of the comparator inhibits comparison for 2 μ sec to wait for stabilization of the BIEC lines (propagation time plus selection of the highest priority level). IR could be sent between T3 and T5, but is never used during status test.

2.101 At the receipt of SCEIN, any CU with a pending interrupt request examines the BIEC lines. If there is no higher priority request, the CU's priority encoder puts the interrupt code on the BIEC0-5 lines. The BIEC code received at the Interrupt logic is switched through the 74158 multiplexer to the INTAD0-5 lines if there is no internal interrupt request.

2.102 Compare Interrupts

The highest-priority interrupt request (internal or external) is available at the INTAD0-5 lines. If the Enable Interrupt flip-flop ENBF is set and the BIECINH delay is completed, the INTAD lines are compared with the priority of the running program (stored in the priority level register PLR). BIECINH and ENBFN to the two most-significant inputs of the comparator must both be low to enable comparison. (The comparator inputs A3/B3 must compare equal before the A2/B2 inputs are tested, etc.; all four A and B inputs must compare before the cascade inputs from the lower-priority comparator are tested.) If the interrupt request on the INTAD lines is of higher priority (lower number) than the current PLR, the A<B output is activated to generate Interrupt Request signal IR.

2.103 The IR signal is sent to the Microprogram Control logic to initiate the Interrupt routine (Figure 2-5). The interrupt address on the INTAD0-5 lines is switched through the C and D selectors during the Interrupt routine to form part of the address in the S register and to reload the PLR with the new priority level.

2.104 SEQUENSOR (CPU CLOCK)

The Sequensor (Figure 2-8 EE) uses a crystal-controlled oscillator to drive twelve shift-register cells which provide the CPU timing signals :

- AP, T1, T2, T3, T4, T5, T6,
- BP, T7, T8, T9, T10.

The basic CPU clock signals are AP and BP. AP represents the beginning of a functional cycle (or microprogram) and is used to load the microinstruction address and read the microinstruction from the ROM control store; CPU logic controls and data paths are established by the command bits of the microinstruction. At each BP pulse time, the CPU logic executes the microprogram, according to the conditions established at AP time, and loads the results into the appropriate CPU registers.

2.105 Operating Modes and Cycles

The Sequensor operated in four different modes (Figure 2-6) controlled by microinstruction bits μ SEQ0,1. Within the basic modes, the Sequensor operates in one of six cycles. Five of the operating cycles begin with the AP pulse; the sixth (Repeat cycle) begins and ends with the BP pulse, skipping AP. For internal CPU operations (LOGIC and REPEAT modes), the cycle lengths are controlled by controlling the generation of AP to begin the next cycle (pulses T7-T10 are inserted or deleted after BP as required). For all GP-bus cycles (SEQBUS and SEQBIO modes) and Floating Point execution, the cycle lengths are controlled by controlling the generation of T6 just before BP (pulse T5 may be repeated as a waiting loop until a synchronizing reply signal triggers T6).

2.106 Internal CPU Operations (AP Generation)

Four fixed-length Sequensor cycles are provided for the internal operations LOGIC mode :

- Logic cycle (360ns)
- Scratchpad cycle (405ns)
- Flag cycle (450ns)
- Fetch cycle (540ns)

mu SEQ 0,1	Mnemonic	Function
0 0	LOGIC	Internal CPU Operations
0 1	REPEAT	Short Execution Cycle
1 0	SEQBUS	GP-Bus Cycles
1 1	SEQBIO	I/O Cycles with Bus not released

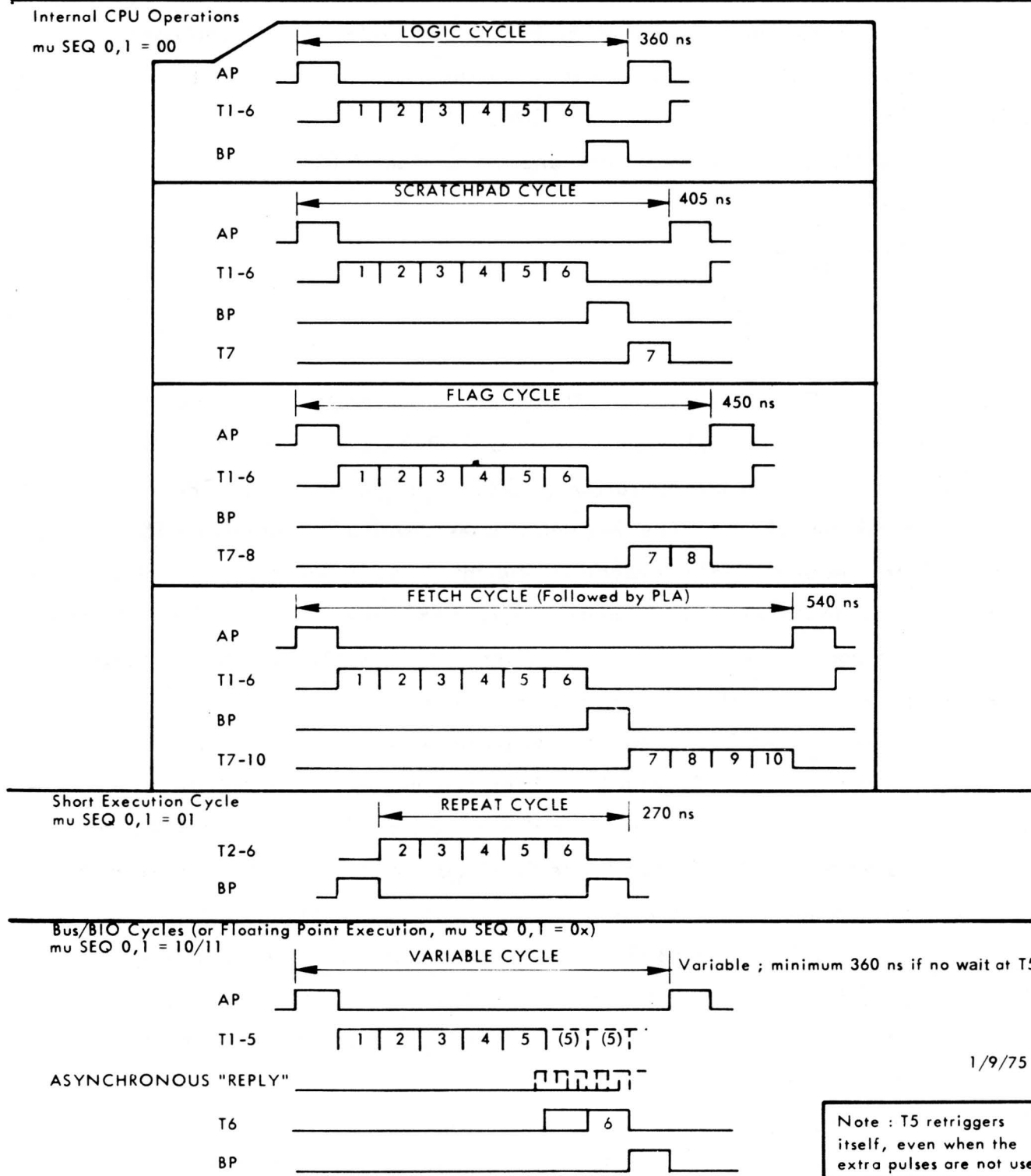


Figure 2-6 Sequencer Operating Cycles

All these cycles begin with AP and count directly through T1, T2, T3, T4, T5, T6, and BP. The time between BP and the end of the next cycle (next AP) is then controlled to provide the required execution time for the operation and to define all Next-Address data. The control is provided by a type 74151A multiplexer (APDN) which selects which clock pulse is to generate AP, as follows :

GFETCH	NAFLAGN	WSPRSLI	APDN Generation	
S2	S1	S0	Input	Function
0	0	0	T8	Flag Cycle (NAFLAG)
0	0	1	T8	
0	1	0	BP	Logic Cycle (NAFLAG.WSP)
0	1	1	T7	
1	0	0	T10	Fetch Cycle (GFETCH)
1	0	1	T10	
1	1	0	T10	
1	1	1	T10	

2.107 Logic Cycle. This cycle is used for register-to-register operations, with the result stored into a register other than the scratchpad. Input 13 of the APDN multiplexer is selected so that the AP pulse to begin the next cycle follows the BP pulse.

2.108 Scratchpad Cycle. This cycle is used for register-to-register operations with the result stored in the scratchpad. Data transfer from the L register to the scratchpad requires the extra clock pulse T7 between BP and the end of the cycle.

2.109 Flag Cycle. This cycle is used with a microinstruction that specifies Next-Address Flag mode. Two extra clock pulses (T7 and T8) are required to gate the Flag information onto the control-store address bus.

2.110 Fetch Cycle. The Sequensor fetch cycle is used during an instruction fetch cycle, where the Next Address must be decoded by the instruction-register PLA decoder. The timing required to decode the instruction word and to access the PLA needs four extra clock cycle (T7, T8, T9, T10) following BP.

2.111 Short Execution Cycle (No AP)

The Repeat cycle counts directly through T2, T3, T4, T5, T6, and BP without an AP or T1 clock pulse. This cycle is used for looping while the scratchpad CT counter counts. The microinstruction control inputs, with \overline{CTEND} , hold $\overline{REPENDI}$ high to gate the BP pulse directly to the T2 input. $\overline{REPENDI}$ also disables the APDN multiplexer to inhibit the AP pulse. The current microinstruction does not change during the Repeat cycles since AP is inhibited. When the CT counter reaches its full count of 31_{10} , \overline{CTEND} forces $\overline{REPENDI}$ low and the APDN multiplexer is enabled to generate AP to begin the next cycle.

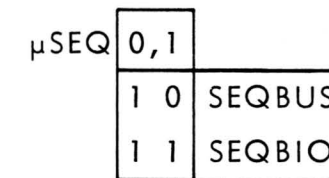
2.112 GP-Bus Cycles (T6 Generation)

Variable-length cycles are used for all GP-Bus operations and for Floating Point instruction executions. All these cycles begin with AP and count directly through T1, T2, T3, T4, and T5. The cycle-length parameters then control the generation of the T6 clock. The BP pulse and the succeeding AP pulse directly follow T6. The control of T6 is provided by a type 74151A multiplexer which selects the T6-enabling signal T6DN, as follows :

μ SEQ0 S2	FLOACT S1	BSYDL S0	T6DN Generation	
			Input	Function
0	0	0	1	Internal CPU operations (T6 follows T5)
0	0	1	1	
0	1	0	DONEF	Floating Point instruction execution
0	1	1	0	not used
1	0	0	0	
1	0	1	SYNCT6	Bus and Bus I/O cycles (SEQBUS, SEQBIO)
1	1	0	0	not used
1	1	1	0	

2.113 For the internal CPU operations, T6 directly follows T5. Sequensor control is provided through control of the AP pulse (preceeding paragraphs). The enabling One-bit is selected through the multiplexer by μ SEQ0=0 and the floating point command bit FLOACT=0.

2.114 The Floating Point instruction execution is an operation that requires the DONEF response signal from the FPP before the cycle can be terminated. DONEF is selected directly at the I2 input of the T6DN multiplexer by μ SEQ0=0, floating-point command bit FLOACT=1, and BSYDL=0. The FLOACT signal is the buffered microinstruction general-field bit GFLOT.



2.115 For all GP Bus operations (SEQBUS and SEQBIO), the SYNCT6 input to the multiplexer is selected by μ SEQ0=1, BSYDL=1, and FLOACT=0. The μ SEQ0 bit = 1 for all Bus operations. BSYDL = 1 95ns after the CPU takes control of the GP Bus. The Sequensor does a waiting loop on the T5 clock until the required reply is received via the 8-input Nand gate SYNCT6. For Bus transfers that don't require memory exchange, the reply signal is provided by the Microprogram Control and T6 may directly follow T5. The reply signal (listed in Table 2-12) must be one of the following:

- a. TRMN (SYNMEMN) must be received during a CPU/Memory exchange or during a CPU/External-Register exchange. The TRMN response signal (from Memory or External Register) on the Bus is received and gated by the Bus Controller logic (Figure 2-8 TT) to the Sequensor as input signal SYNMEMN.
- b. TPMN delayed 190ns (TPMDLN) is received from the CU during a CU/Memory exchange under CPU control (programmed-channel transfer). The 190ns delay of TPMN is produced by Bus Controller logic for assuring data timing.
- c. TIMEOUT generates T6 to end the cycle if no response is received to one of the Master-to-Slave signals TMRN, TMPN, TMEN within

6.48 μ sec. This may indicate that the addressed Memory, Programmed Channel, or External Register is not present.

- d. BIOEKEY gates T6 without extra T5 waiting loops during a control-panel to CPU transfer. BIOEKEY is generated on the Bus Controller logic (Figure 2-8 TT) by the microinstruction general-field bits GCRFNUN and GBOKFN, and gated into the Sequensor by GIDLEN.
- e. GFETCHN general-field bit gates T6 without extra T5 waiting loops during instruction (K) register loading from the L register, via the Bus BIO lines. GFETCHN is gated by microinstruction control bit μ TMRN to generate SYNIMN to the T6 input gate.
- f. DONEMN is received from the MMU during an MMU/Memory exchange which is always under CPU control.
- g. DONEFN is received from the FPP during an FPP/Memory exchange which is always under CPU control.
- h. MFAULTN, delayed 120ns, from the MMU indicates a page fault (PAFN) during a Move Table instruction. The MFAULTN signal sets the PAF flip-flop in the control-store addressing logic (Figure 2-8 CC). The PAFN signal is delayed 120ns at the Sequensor input (SYNCT6) for control-store address timing.
- i. BOFFN from the general-field bits GCRFNUN and GBOFN (Bus Controller logic) gates T6 without extra T5 waiting loops when FPP data is loaded, via the Bus BIO lines, into the CPU during a Fix instruction.

2.116 POWER FAILURE, RESTART, RESETS

The power-on and power-off sequences are shown on Figure 2-7. The logic is shown on Figure 2-8 RR, Start/Resets. While power is off, the Reset Logic (RSLN) signal is held at ground (active low) through a relay on the power supply.

2.117 Power-On Sequence

When the power supply is switched on, or when the mains power is restored after a failure, there is approximately an 850ms delay while the power supply stabilizes (Section 5, Power Sequence Logic) before RSLN goes inactive (high).

Table 2-12 Sequensor T6 Generation

μ SEQ0	FLOACT	BSYDL	Cycle (text)	Synchronize T6 on :
S2	S1	S0		
0	0	x	Internal CPU operations	T5
0	1	0	Internal, Floating Point execution	DONEFN from FPP
1	0	1	GP Bus transfers with Memory exchange	
			CPU/Memory (a)	TRMN (SYNMEMN) from memory
			CPU/External Register (a)	TRMN (SYNMEMN) from memory
			Programmed Channel CU/Memory (b)	TPMN from CU, delayed 190ns
			Addressed Device Missing (c)	TIMEOUT, generated 6.48 μ sec after TMRN/TMPN/TMEN sent.
			MMU/Memory (f)	DONEMN from MMU *
			FPP/Memory (g)	DONEFN from FPP *
			Page Fault (h)	MFAULTN from MMU (PAFN), delayed 120ns. *
			GP Bus transfers without Memory exchange	
			Control Panel (d)	T5, if GBOKFN/GCRFNUN (BIOEKEY), and GIDLEN
			Fetch execution (e)	T5, if GFETCHN and μ TMRN.
Load from FPP (Fix execution) (i)	T5, if GBOFN (BOFFN) and GCRFNUN *			
any (a-i)	TIMEOUT, if required reply not received within 6.48 μ sec			

* Note - Signals (f, g, h, i) used with 857 only.

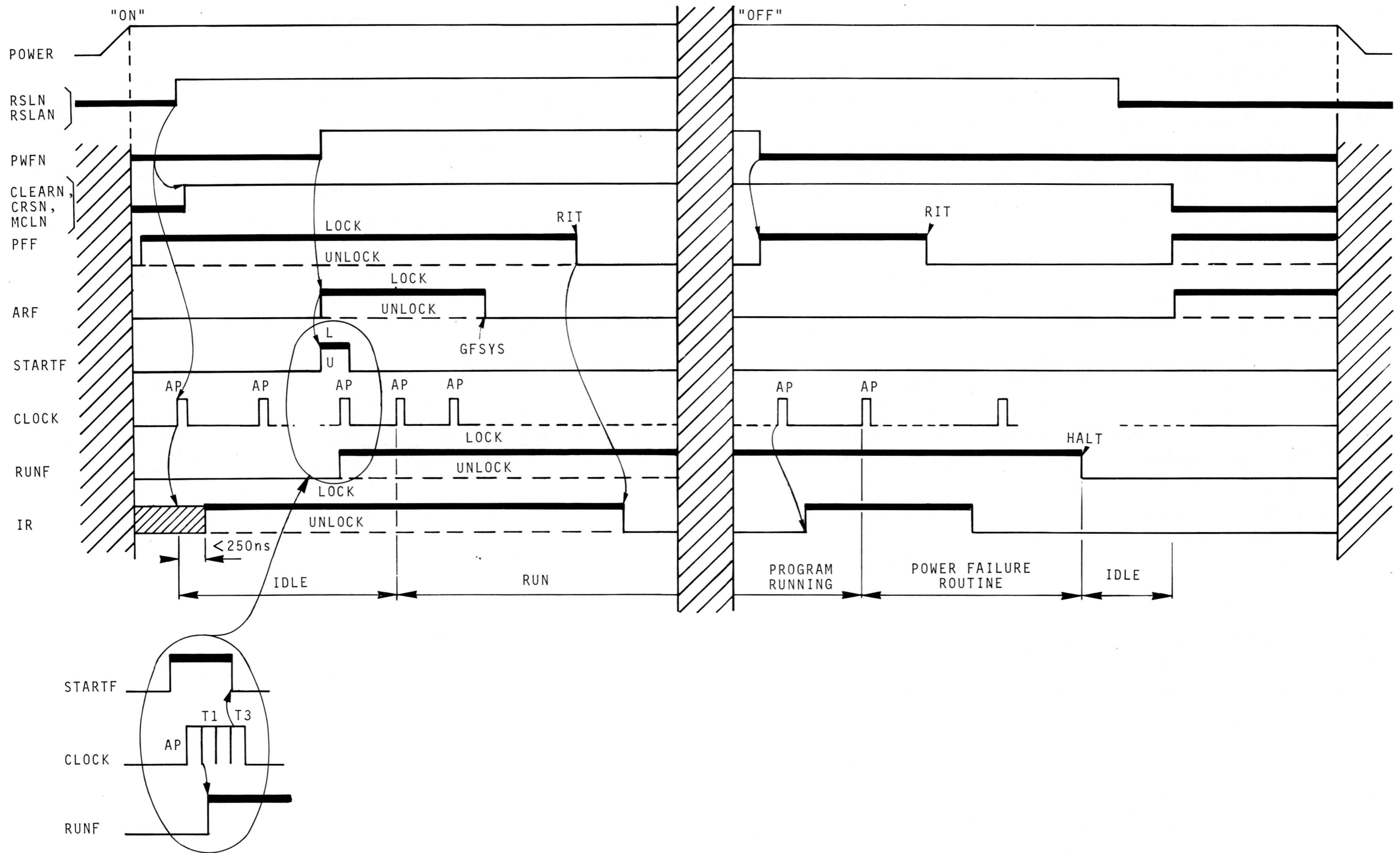


Figure 2-7 Power On/Off Sequence

When RSLN goes high, the OSCN pulse immediately resets RSLF to remove the logic reset condition. The inactive RSLAN/BN/CN signals to the Sequensor enable clock-pulses T1, T3, T5, T6, AP, BP, and the clock begins running.

2.118 Automatic Restart

Automatic restart is performed by the CPU if the control-panel key is at LOCK when the power comes on, therefore the key must be switched to LOCK when switching on or following a power failure while the key is not at LOCK. If the key is at LOCK when the +5 volts comes on to operate the flip-flops (and before RSLN, RSLF drop), PFF is set at the beginning of the power-on sequence (paragraph 2.93, PSW/GF, PFF). Beginning with the first AP, the internal interrupt request PFF is sampled by the Interrupt Logic. A level-0 (highest priority) Interrupt Request (IR) is then generated and the any-internal-interrupt signal IECGSN is active to the ARF flip-flop. The Power Failure (PWFN) signal from the power supply goes inactive (high) 0.5ms after RSLN goes inactive.

2.119 With the key at LOCK, the IECGSN signal sets Automatic Restart flip-flop ARF at the edge of the high-going PWFN signal. The ARFN output sets the STARTF flip-flop directly, via the asynchronous S-input. The RUNF flip-flop (2-8 PP) is set on the first T1 clock after STARTF is set. At the first AP after RUNF sets, the first microinstruction of the Automatic Restart routine (Figure 1-13) is read, and the CPU switches from Idle state to Run state. The second microinstruction of the Automatic Restart routine resets ARF with the GFENBN signal. The PFF flip-flop should be reset during the automatic restart subroutine by an RIT instruction.

2.120 Power-Off Sequence

The Power Failure (PWFN) signal is set by the power supply 10ms after detection of a mains loss, caused by power failure or by the key switched OFF. PWFN is sent via the GP Bus to the CPU and to any other elements on the Bus that must take action during power failure. The leading edge of PWF sets the PFF (GF/interrupt) flip-flop. This level-0 internal interrupt is detected

by the Interrupt Logic at the first AP clock following the start of PWFN. The resulting Interrupt Request (IR) can initiate an interrupt subroutine for a power failure.

2.121 PWFN to the Bus Controller logic inhibits setting the OKVAL flip-flop. The Bus Controller thus denies requests by any other Master for Bus access, and the CPU has exclusive control of the Bus. The power failure subroutine requested by IR and PFF should commence with the second AP clock following PWFN. The subroutine should then reset PFF with an RIT instruction and reset RUNF with an HLT instruction. This saving subroutine must not last more than 2ms, and must end with a HLT instruction. The HLT instruction selects Microprogram Control address /177 to place the CPU in the Idle state. The Reset signal (RSLN) from the power supply is activated 3ms after PWFN.

2.122 Resets

The CPU logic is reset at the end of the power-off sequence by the RSLN signal from the power supply. RSLN is held active low while power is off. The first OSCN pulse after RSLN is active sets the RSLF flip-flop. RSLF activates RSLAN, RSLBN, RSLCN, and WSPRSLI, and resets the flip-flops STARTF, WSP, and RUNF. The RSLF-derived signals set or reset the CPU registers and flip-flops to their off states (Table 2-13) and hold them in that condition until the power-on sequence. The RSLN signal also activates the Master Clear signals CLEARN (to the Bus), MCL, and MCLN. Sequensor clock pulses are disabled by RSLF so that the CPU clock is stopped, although the oscillator-generated signal OSC runs continuously.

2.123 Master Clear

The Master Clear signal CPMCN can be generated by the control-panel MC switch if the key is not at LOCK and if the CPU is not in Run mode. The CPMCN signal (Figure 2-8RR) is OR'ed with the RSLF output to activate the signals CLEARN, MCL, and MCLN. CLEARN is sent on the GP Bus as a reset/initialize signal to all System elements. MCL is sent to the V24 CU on the CPU card. MCLN is used by the CPU (Table 2-13) to set the priority level register (PLR) to

Table 2-13 Resets and Clears

Logic-Resets, by RSLN/RSLF only

Logic	Sig.	Action	Logic	Sig.	Action
(RR)	RSLF	RSLAN, BN, CN → 1 STARTF → 0 WSPRSLI → 0	(JJ)	RSLCN	Q-reg → 0
	WSPRSLI	WSP → 0	(LL)	RSLBN	DWIF → 1
	RSLAN	ARFN → 0 IPLF → 0	(NN)	RSLBN	OVF → 0 CR0,1 → 0
(AA)	RSLBN	K-reg → 0	(PP)	RSLF	RUNF → 0
	RSLAN	KxxBUFN → 0	(SS)	RSLAN	BIECINH → 0 IF00-07 → 0
(BB)	RSLBN	RA0-8 → 0	(TT)	RSLAN	BUSF → 0 DE → 0 FNU → 0 (DD) TMEF → 1 TMPF → 1 WRITFN → 0
(CC)	RSLAN	FTDEL → 0 KRY → 0			
	RSLBN	PUPN → 0			
(DD)	RSLBN	FSIG → 0 FSIGDIV → 0	RSLBN		BIOELN → 1 BSYCPU → 0 CHFEN → 1 FLOACT → 0 GBCPFN → 1 MADCPUN → 1 MADLCPUN → 1 MUBUSRFN → 1 OKVAL → 0 TMFN → 1 TMMN → 1 TMMU → 0 TMRF → 0
(EE)	RSLBN	T1, T3, T5 → 0 TC810 → 0			
	RSLAN	AP → 0 BP → 0 T6 → 0			
	RSLCN	BPQ → 0			
(PP) RSLF.PWF.LOCK set interrupt flip-flop PFF					
Resets by Master Clear or RSLN/RSLF					
(RR)	CPMCN or RSLAN	MCL → 1			
	MCL	CLEARN → 1 MCLN → 1 V24 resets			
(NN)	MCLN	PLR → level 63			

Logic	Sig.	Action
(PP)	MCLN	ENBF → 1 CPINTF → 0 RTCF → 0 PIF → 0 FU → 0 CPGFZ0N → 1
	CPGFZ0N	CPGF → 0 V24 resets

level 63 and set the Enable Interrupt flip-flop ENBF. The GF flip-flops CPINTF, RTCF, PIF, and RU are all reset, and the control panel interrupt flip-flop CPGF is reset.

Table 2-14 CPU Signal LIST (G-M)

GCRFNU, N	BB, DD	CMD/G	K00-15, N	AA	
GCRVMLN	BB	CMD/G	K0008	AA	
GCRVZ0, N	BB, DD	CMD/G	K05-08BUFN	AA	
GCSELN	BB	CMD/G	K08Q15	FF	
GCT	MM		K10R2E0	AA	
GCTLDN	BB	CMD/G	K10R2N	AA	
GFENBN	BB	CMD/G	K410N	AA	
GFETCH, N	BB	CMD/G	K415N	AA	
GFKYZ0N	BB	CMD/G	K567	AA	
GFLOT, N	BB	CMD/G	KFL0	AA	
GFPLRN	BB	CMD/G	KRY, N, DN	CC	
		to CLPR	KRYZ0N-Z1N	CC	
GFRZ0, N	BB	CMD/G			
GFSTOV, N	BB	CMD/G	L00-15	HH	
GFSYSN	BB	CMD/G	LOADMN	CC, LL	in from C. Panel
GIDLEN	BB	CMD/G	LOADRN	CC, LL	in from C. Panel
CMOVEN	BB	CMD/G	LSEL	FF	
CMULTI, N	BB, FF	CMD/G			
GOSH	DD		M00N-15N	GG	
GOTOECH	V24		MAD00-15, 64, 128	KK	
GOTOWST	V24		MADCPB	TT	
			MADCPUN	TT	
IEC0-2, N	SS		MADLCPUN	TT	
IECE	SS		MADLDN	TT	
IECGSN	SS		MADLS	TT	
IF00N-07N	SS		MADMMU, N, A	TT	
IN	V24		MADMUN	TT	
INRECH	V24		MADS	TT	
INSTN	--	not used (C. Pan)	MCL, N	RR	
		Interrupt address	MFAULTN	CC	in from MMU
INTAD0-5	SS		MMUABS	TT	in from MMU
			MOVE	CC	
INTEPL	SS		MOVIRN	CC	
INTGPL	SS		MSN	TT	BUS (in)
INTLPL	SS		MSR	TT	
INTSERN	V24		μA0-4	BB	CMD
INTVAL	V24		μA0N-4N	LL	
IPL, N	RR	in from C. Panel	μATN	RR	
			μADL0-4	BB	CMD
IPLDEL	RR		μBIOL, N	BB, TT	CMD
IPLF, N	RR		μBUSR, N	BB, TT	
IR, N	SS		μBUSRFN	TT	CMD
IS00N-07N	SS	in (Intern. Int's)	μC0-1	BB	CMD

Table 2-14 CPU Signal LIST (M-R)

μC0N	JJ		OSC, N	EE	
μCR0-2	BB	CMD	OSC90, N	EE	
μCT	BB	CMD	OSCFLO	EE	
μGP0-4	BB	CMD to CMD/G	OTINECH	V24	
μGP1N	BB	CMD to CMD/G	OTRECHN	V24	
μMLOAD	BB	CMD	OUT	V24	In/Out FF
μMSEL	BB	CMD	OVDIVA, B	NN	
μNA0N-8N	BB	CMD	OVF, N, D	NN	
μNA7	AA		OVFMU, N	NN	
MUNA7FUN	AA		OVSHA, B	NN	
μP0-1	BB	CMD			
μP0N	LL		P00-14	LL	
μQ0-1	BB	CMD	PAFDEN	EE	
μS0-1	BB	CMD	PAFN, Z0N	CC	
μS0N	KK		PARNB	V24	
μSEQ0-1	BB	CMD	PCO03, 07, 11N	LL	
μSEQ1N	BB	CMD	PE, PEB	V24	
μSNA0-1, N	BB	CMD	PFF, N	PP	
μTMRN	BB	CMD	PFFZ0N, Z1N	PP	
μWRITE	BB	CMD	PIF, N	PP	
MVREADN	CC		PIFZ1N	PP	
MVRML	CC		PLOADN	LL	
			PLA0-1, 0N	AA	
NACND	AA		PLAVALI	AA	
NACNDAD	AA		PLAVALN	AA	
NADRT	CC		PLR0-5	NN	
NADRTMV	CC		PMI	LL	
NAETAN	BB	CMD μSNA0.1	POVFN	FF	
NAETAT	CC		POWON	PP	
NAEXPL	CC		PREQN	PP	in from C.P.
NAFLAGN	CC		PREQT1	PP	
NAFLG	CC		PUPN, D	CC	
NAFLGX	CC		PWF, N	TT	in from P. Supply
NAPLINH	CC		PWFAN	PP	
NOFLON	AA				
NOJUMPN	AA		Q00	JJ	
NOVF	FF		Q00N-16N	JJ	
NXTCH, N	V24		Q15DN	JJ	
NXTCH0	V24				
ODEVPAR	V24		R1E0, N	AA	
OKO	TT	BUS	R1E15	AA	
OKVAL	TT		R2E0, N	AA	
OR, ORB	V24		R2E15	AA	
OROPE	V24		R110, N	V24	

Table 2-14 CPU Signal LIST (R-T)

R600,1200	V24		SCEIN	SS	BUS
R2400,4800	V24		SCEIT	SS	
R9600	V24		SEQINH	V24	
RA0-8	BB		SLOADN	KK	
RAD0-8	AA,CC	Control Addr-Bus	SPA0-3	MM	
			SPYC,A	TT	BUS
RADET6N-T8N	CC		SPYCOKN	TT	
RADETPLN	CC		SPYCADEL	TT	
RADL3-4	FF		SSST	V24	
RALUATN-2N	FF		STAFCL,A	RR	
RATESEL	V24		STAFZON	RR	
RCP0N-3N	LL	in from C. Panel	START,N	RR	in from C. Panel
RD08-15	V24		STARTF	RR	
RDA,N,1	V24		STOPNB	V24	
RDADEL,N	V24		STOV	GG	
RDARC	V24		SYNCT6	EE	
RDARN	V24		SYNIMN	EE	
READMN	CC,LL	in from C. Panel	SYNMEMN	TT	
READRN	CC,LL	in from C. Panel	T 1-10	EE	
READSTN	CC,LL	in from C. Panel	T1,3,5,6,8N	EE	
REPENDI	EE		T2,5D	EE	
RFBN	AA		T2DREP	EE	
ROMCR1-8	NN		T4T5N	EE	
ROMCREN	NN		T6DN,DEN	EE	
ROMENB	BB		TC810,N	EE	
RSLAN,BN,CN	RR		TDSN	V24	
RSL,N	RR	BUS (in from P.S.)	TE0C	V24	
RSLF	RR		TESTN	CC	in from C. Panel
RTCF,AN	PP		TIMEOUT,N	TT	
RTCFZ1N	PP	in from P. Sup.	TMEF	TT	
RUNF,N,A	PP		TMEN	TT	BUS (in/out)
RUNFZON,Z1N	PP		TMER,N	TT	
RUNN	PP	in from C. Panel	TMF,N	TT	
RUNT3	RR		TMMOFN	TT	
			TMMCY	TT	
			TMMN	TT	
			TMMU,D	TT	
			TMPF	TT	
			TMPN	TT	BUS (in/out)
S00-03	KK		TMPR,N	TT	
S00N-15N	KK		TMR,N	TT	BUS (in/out)
SCO03,07,11N	KK		TMRF,D	TT	

Table 2-14 CPU Signal LIST (T-W)

TMRR,N	TT	
TOUTZON	TT	
TPMDLN	TT	
TPMN	TT	BUS (in/out)
TPMR	TT	
TPMRDEL	TT	
TRMN	TT	BUS (in/out)
TRMR	TT	
TROHLTN	V24	
TROUBLEN	V24	
TYAC	V24	
TYAD0-5	V24	address code
TYARE,N	V24	
UNLOCK,N	PP	in from C. Panel
VALK08N	V24	
VALNA0	CC	
WRITE	TT	BUS
WRITEFN	TT	
WSP,N	RR	
WSPRSLI	RR	

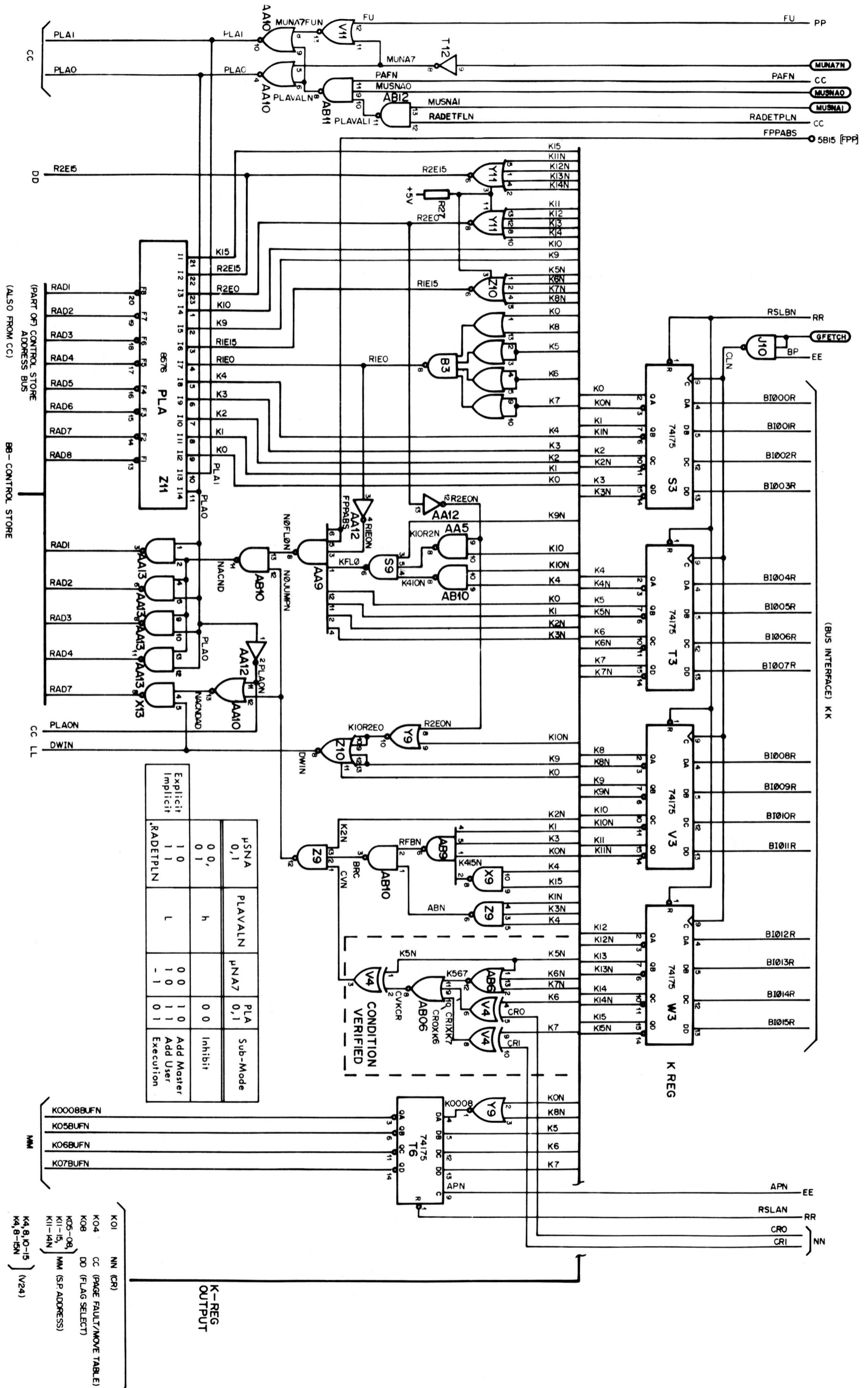


Figure 2-8AA Instruction Word Logic

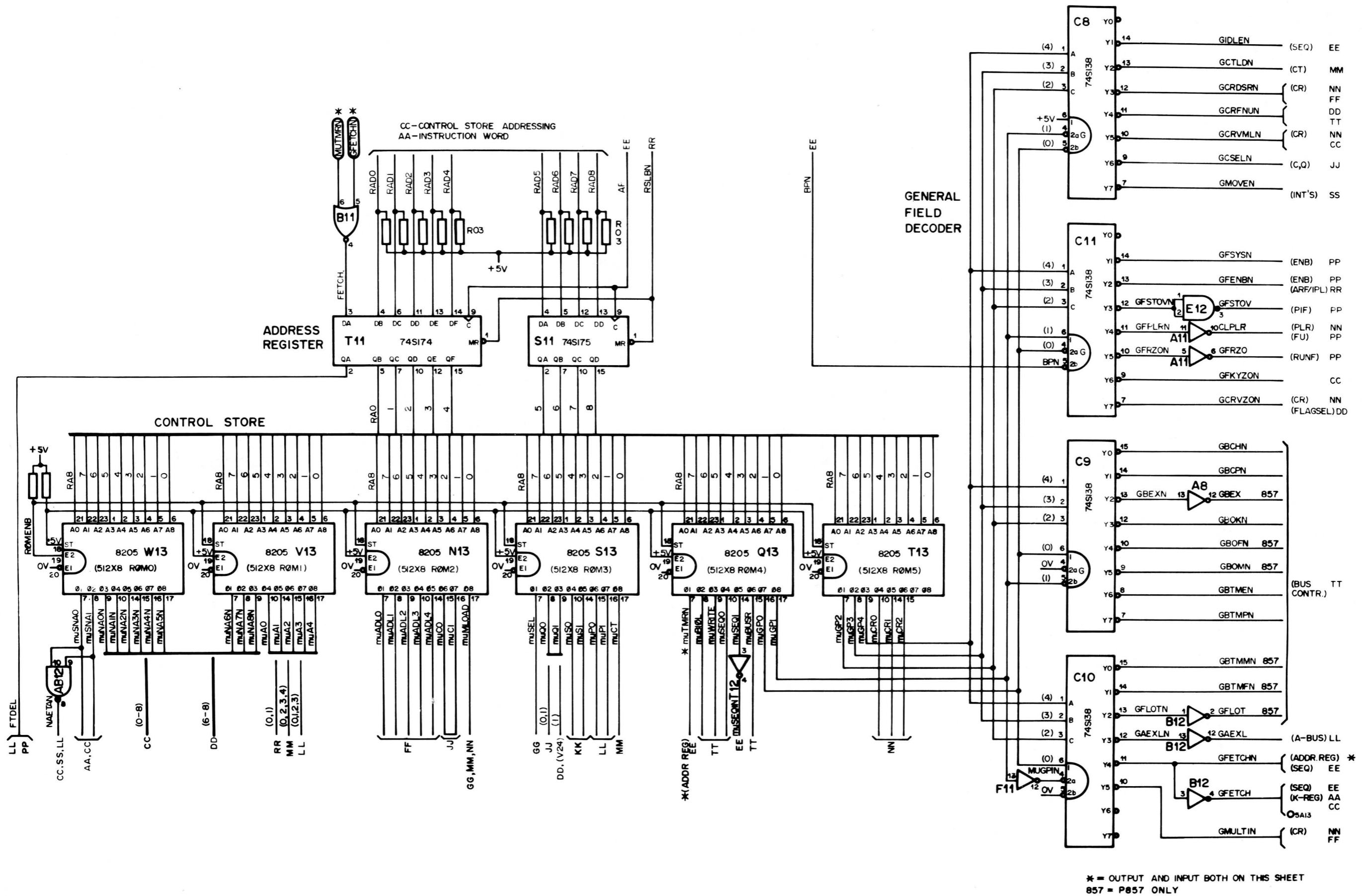
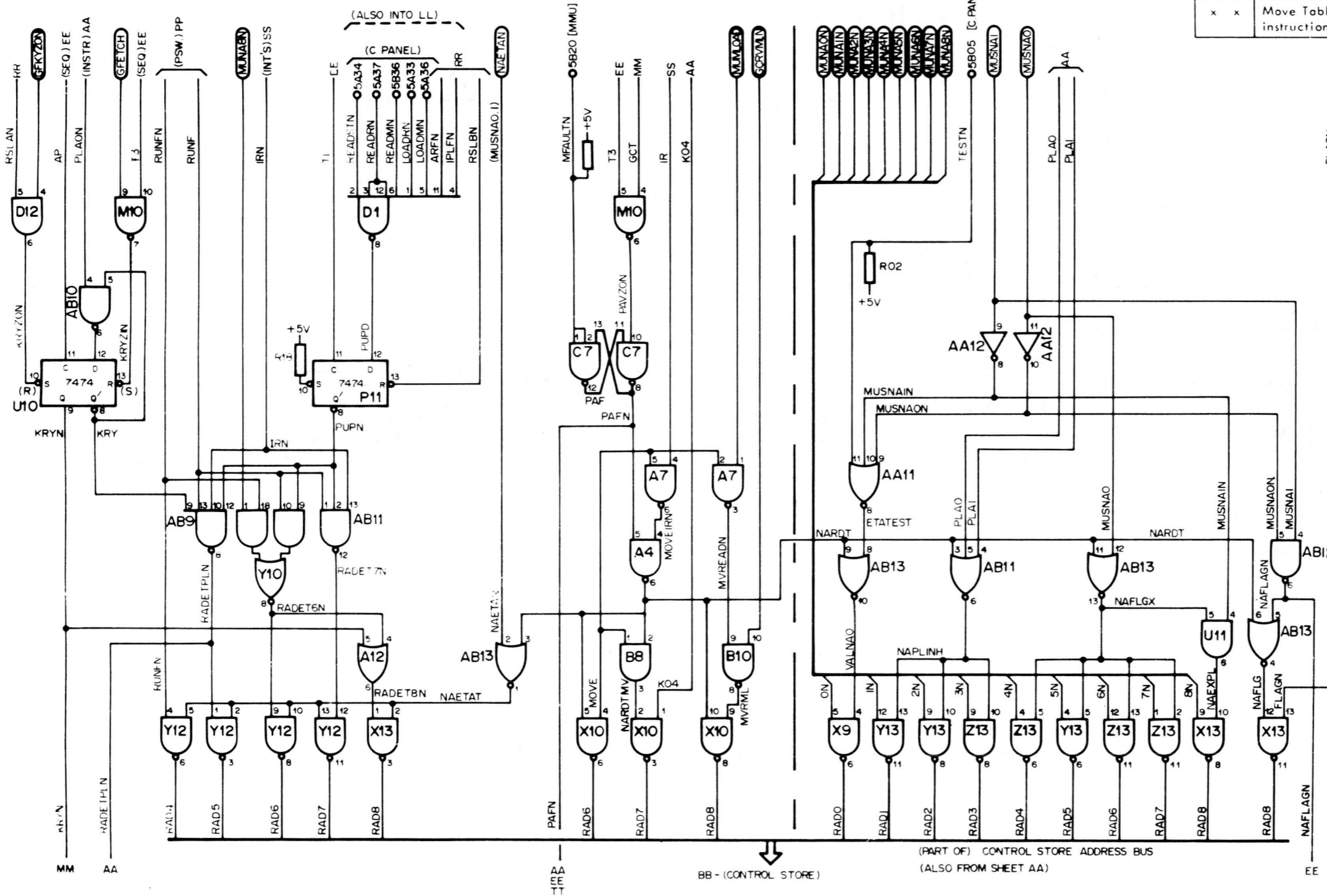


Figure 2-8BB Microcommand Control-Store Logic

MACHINE STATE POINTER

MOVE-TABLE/FAULT | EXPLICIT ADDRESS/TEST FLAG

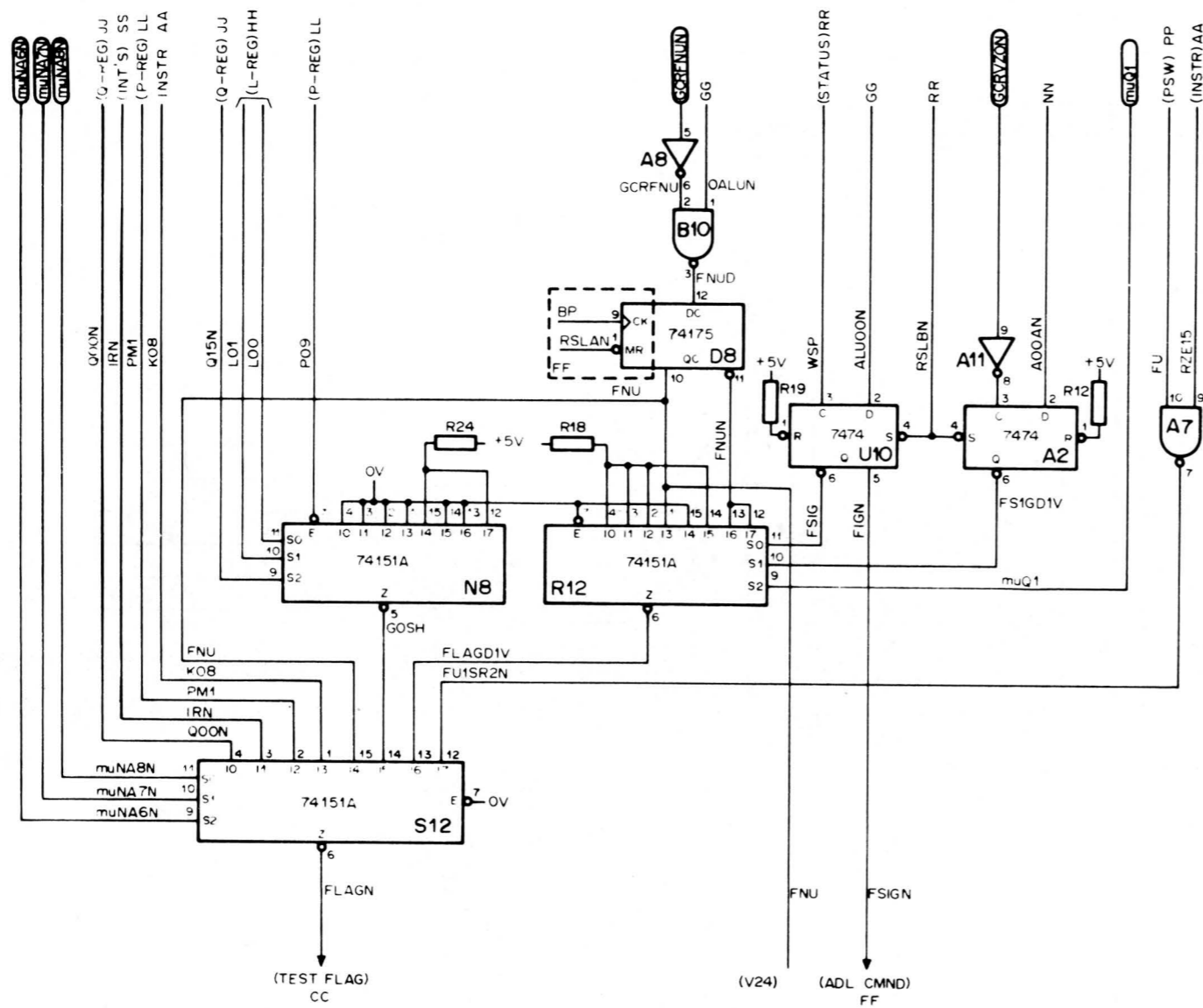
μ SNA 0 1	Mode	Function
0 0	Explicit	Explicit address is μ NA0-8.
0 1	Flag	Flag from execution-pointer tests (selected by μ NA6-8) sets RAD8; μ NA0-7 to RAD0-7.
1 0	Instruction Word (PLA)	Instruction word (K-reg, decoder) provides address RAD1-8; μ NA7 modifies the decoder for fetch or execution.
1 1	Machine-State	The machine-state pointer provides address RAD4-8; μ NA0-3 to RAD0-3.
x x	Move Table/Fault	an interrupt or a page fault during a Move Table instruction force a special microinstruction address with bits RAD6-8.



μ NA6,7,8N	Flag
0 0 0	Q00
0 0 1	IR
0 1 0	PMI
0 1 1	K08
1 0 0	FNU
1 0 1	NORM
1 1 0	DIV
1 1 1	FUI5R2

NARDT BLOCKS GATES,
FOR HIGH OUTPUTS.
M STATE 4-8
EXPL ADDR 0-7
8 (IF MUSNAI)
TEST FLAG 8

Figure 2-8CC Control Store Addressing



GOSH Select

Q15N S2	L01 S1	L00 S0		
L	0	0	0	0
L	0	1	1	0
L	1	0	2	0
L	1	1	3	0
h	0	0	4	+5
h	0	1	5	0
h	1	0	6	0
h	1	1	7	+5

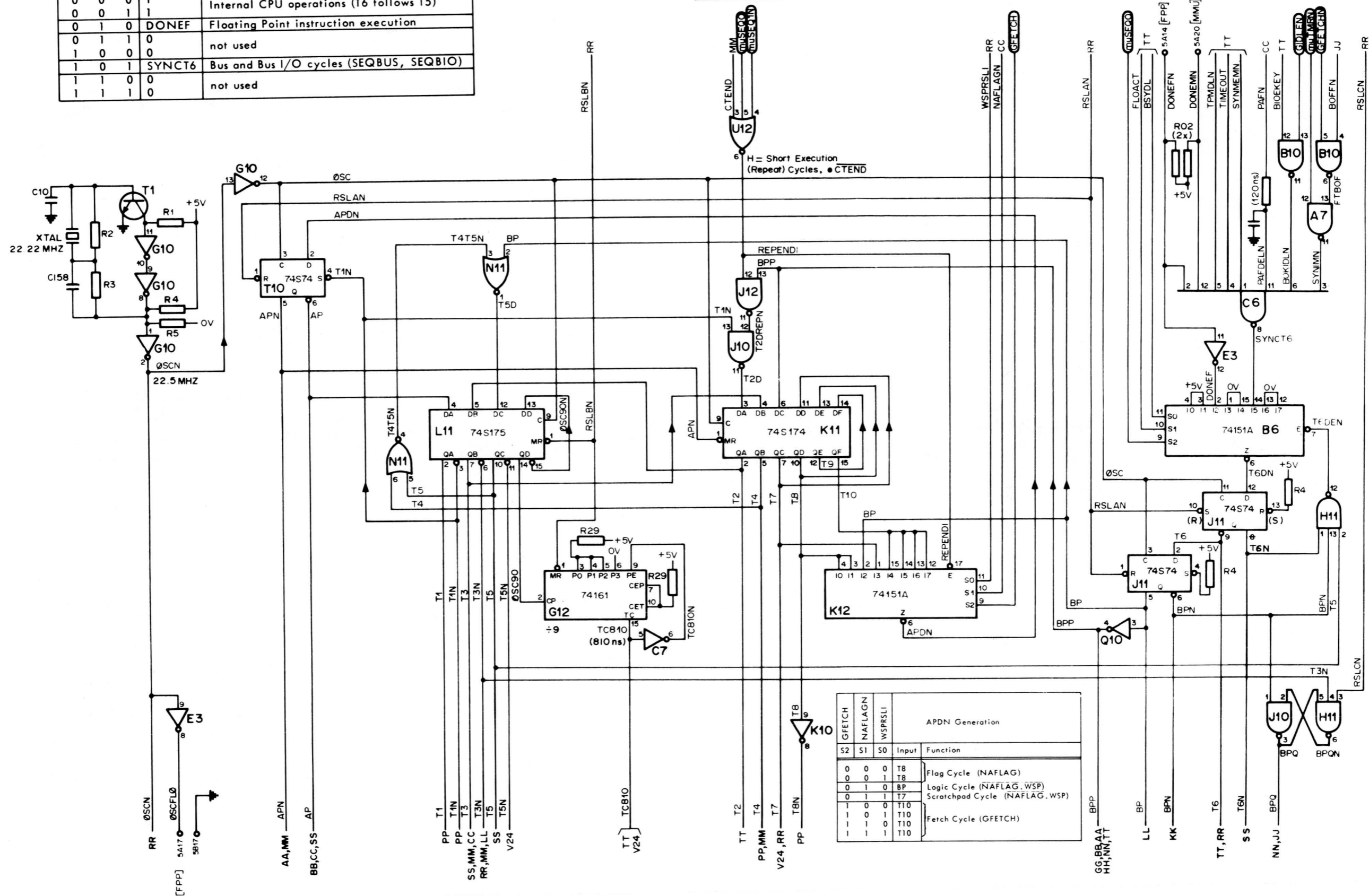
FLAGDIV Select

	Q1 S2	FSIGDIV S1	FSIG S0		
DIVB	0	0	0	0	+5
	0	0	1	1	+5
	0	1	0	2	+5
	0	1	1	3	FNU --- FSIGDIV.FSIG.FNUN
DIVA	1	0	0	4	0 --- FSIGDIV.FSIG
	1	0	1	5	+5
	1	1	0	6	FNUN
	1	1	1	7	FNUN --- FSIGDIV.FNU

Figure 2-8DD Flag Select

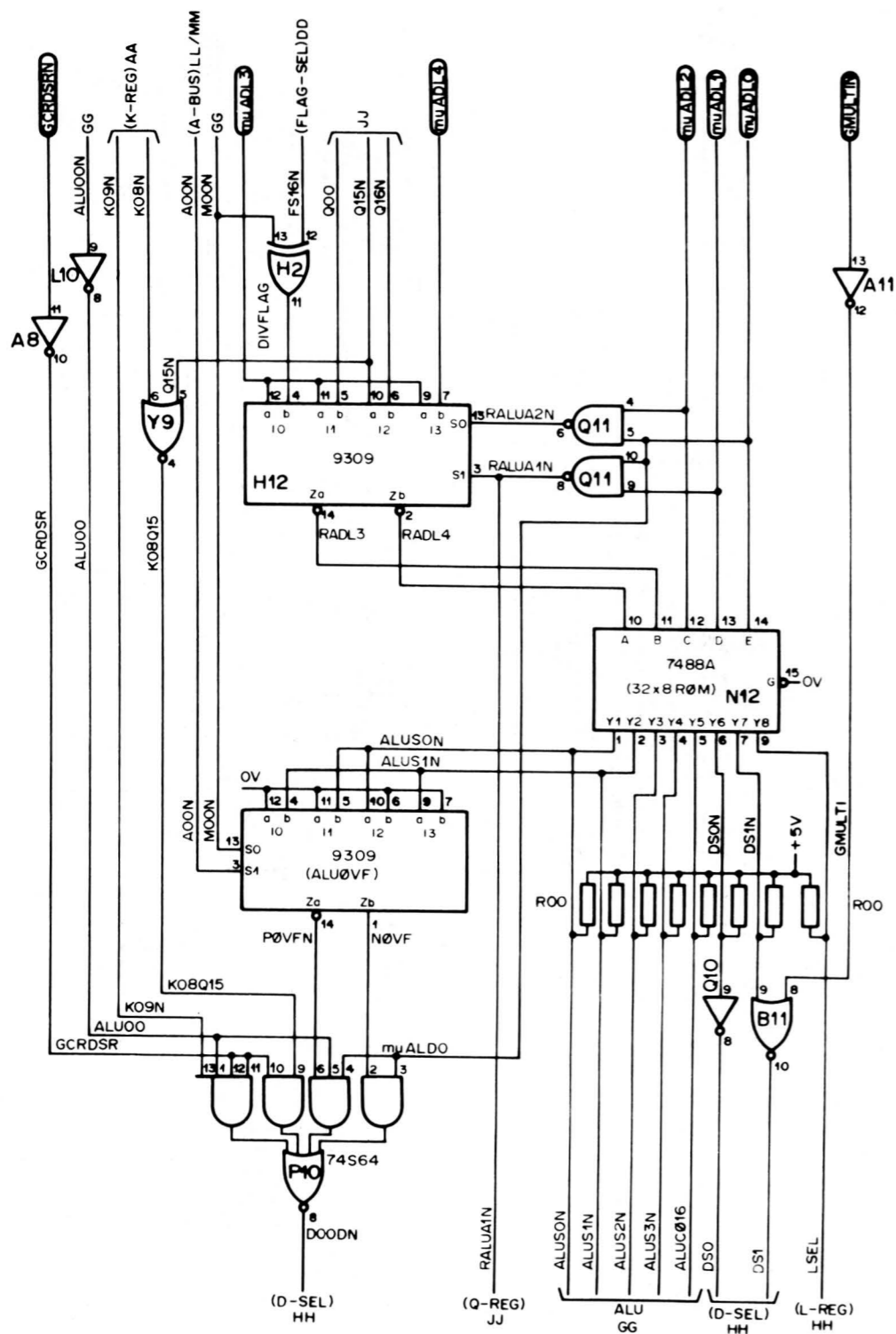
μSEQ0			T6DN Generation	
S2	S1	S0	Input	Function
0	0	0	1	Internal CPU operations (T6 follows T5)
0	0	1	1	Internal CPU operations (T6 follows T5)
0	1	0	DONEF	Floating Point instruction execution
0	1	1	0	not used
1	0	0	0	not used
1	0	1	SYNCT6	Bus and Bus I/O cycles (SEQBUS, SEQBIO)
1	1	0	0	not used
1	1	1	0	not used

mu SEQ 0,1	Mnemonic	Function
0 0	LOGIC	Internal CPU Operations
0 1	REPEAT	Short Execution Cycle
1 0	SEQBUS	GP-Bus Cycles
1 1	SEQBIO	I/O Cycles with Bus not released



APDN Generation			Input	Function
S2	S1	S0	Input	Function
0	0	0	T8	Flag Cycle (NAFLAG)
0	0	1	T8	Logic Cycle (NAFLAG.WSP)
0	1	0	BP	Scratchpad Cycle (NAFLAG.WSP)
0	1	1	T7	Scratchpad Cycle (NAFLAG.WSP)
1	0	0	T10	Fetch Cycle (GFETCH)
1	0	1	T10	Fetch Cycle (GFETCH)
1	1	0	T10	Fetch Cycle (GFETCH)
1	1	1	T10	Fetch Cycle (GFETCH)

Figure 2-8EE Sequensor (CPU Clock)



S1	S0			
DS0	1			
0	0	ALU 00-15N		ALU direct
0	1	ALU 08-15N	ALU 00-07N	Exchange ALU characters
1	0	ALU 00-14N		ALU shift right
1	1	BIO 00-15AN		BIO direct
D	00	07	08	15

LSEL	Data Source	Function
0	D 00-15	D direct
1	D 01-15	Q00 D shifted left
L	00 — 14 15	

Enable Clock	Select C or Q	Mnemonic	Data Source
μMLOAD	μMSEL		
0	-	No Op	Off; previously stored data available at output.
1	0	MYC	C 00 15
1	1	MYQ	Q 00 15
			M 00 15

μADL4	ADL-Control ROM				Operation		
	ALUC016	ALUS0-----3					
CE	CIN	S3	S2	S1	S0		
0	0	1	0	0	1	A plus B	ARITHMETIC
0	0	1	0	1	1	A or B	
0	0	1	1	0	0	A plus A	
0	1	0	0	0	1	A and B	
0	1	0	0	1	1	Zero	
0	1	0	1	1	0	A minus B	
1	-	0	1	0	1	B inverted	LOGIC
1	-	1	0	0	1	A ⊕ B	
1	-	1	0	1	0	B	
1	-	1	1	1	1	A	
1	-	1	0	0	1	A ⊕ B + 1	

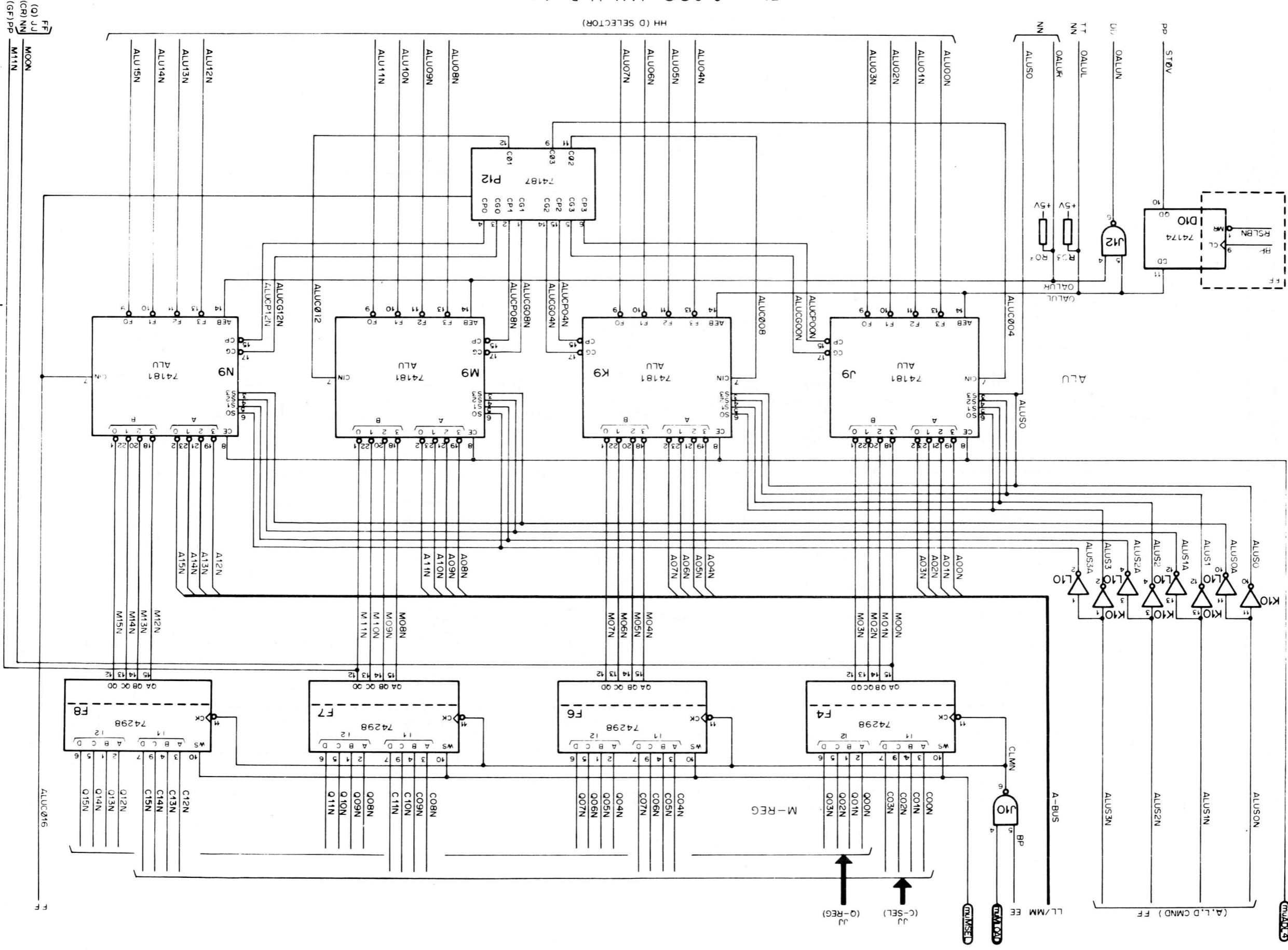
Control signals are all active high. Functions are for active-low data in and out.

Multiplexer Select Input		Multiplexer Output to ADL Command ROM	
μADL: 0, 1, 2	RALUA: 1N, 2N	RADL3 (B)	RADL4 (A)
0	0	μADL3N	μADL4N
0	0	μADL3N	μADL4N
0	1	μADL3N	μADL4N
0	1	μADL3N	μADL4N
0	1	μADL3N	μADL4N
1	0	Q15	Q16
1	1	μADL3N	FSIGN
1	1	μADL3N	DIVFLAGN

DIVFLAG = FSIGN ⊕ M00N

Figure 2-8FF A, D, L Command Logic

Figure 2-8GG ALU, M Register



FF (A, L, D CMD) FF
 LL/MM EE
 (M) (A) 3
 (M) (S) 4
 (M) (S) 5
 (M) (S) 6
 (M) (S) 7
 (M) (S) 8
 (M) (S) 9
 (M) (S) 10
 (M) (S) 11
 (M) (S) 12
 (M) (S) 13
 (M) (S) 14
 (M) (S) 15
 (M) (S) 16
 (M) (S) 17
 (M) (S) 18
 (M) (S) 19
 (M) (S) 20
 (M) (S) 21
 (M) (S) 22
 (M) (S) 23
 (M) (S) 24
 (M) (S) 25
 (M) (S) 26
 (M) (S) 27
 (M) (S) 28
 (M) (S) 29
 (M) (S) 30
 (M) (S) 31
 (M) (S) 32
 (M) (S) 33
 (M) (S) 34
 (M) (S) 35
 (M) (S) 36
 (M) (S) 37
 (M) (S) 38
 (M) (S) 39
 (M) (S) 40
 (M) (S) 41
 (M) (S) 42
 (M) (S) 43
 (M) (S) 44
 (M) (S) 45
 (M) (S) 46
 (M) (S) 47
 (M) (S) 48
 (M) (S) 49
 (M) (S) 50
 (M) (S) 51
 (M) (S) 52
 (M) (S) 53
 (M) (S) 54
 (M) (S) 55
 (M) (S) 56
 (M) (S) 57
 (M) (S) 58
 (M) (S) 59
 (M) (S) 60
 (M) (S) 61
 (M) (S) 62
 (M) (S) 63
 (M) (S) 64
 (M) (S) 65
 (M) (S) 66
 (M) (S) 67
 (M) (S) 68
 (M) (S) 69
 (M) (S) 70
 (M) (S) 71
 (M) (S) 72
 (M) (S) 73
 (M) (S) 74
 (M) (S) 75
 (M) (S) 76
 (M) (S) 77
 (M) (S) 78
 (M) (S) 79
 (M) (S) 80
 (M) (S) 81
 (M) (S) 82
 (M) (S) 83
 (M) (S) 84
 (M) (S) 85
 (M) (S) 86
 (M) (S) 87
 (M) (S) 88
 (M) (S) 89
 (M) (S) 90
 (M) (S) 91
 (M) (S) 92
 (M) (S) 93
 (M) (S) 94
 (M) (S) 95
 (M) (S) 96
 (M) (S) 97
 (M) (S) 98
 (M) (S) 99
 (M) (S) 100

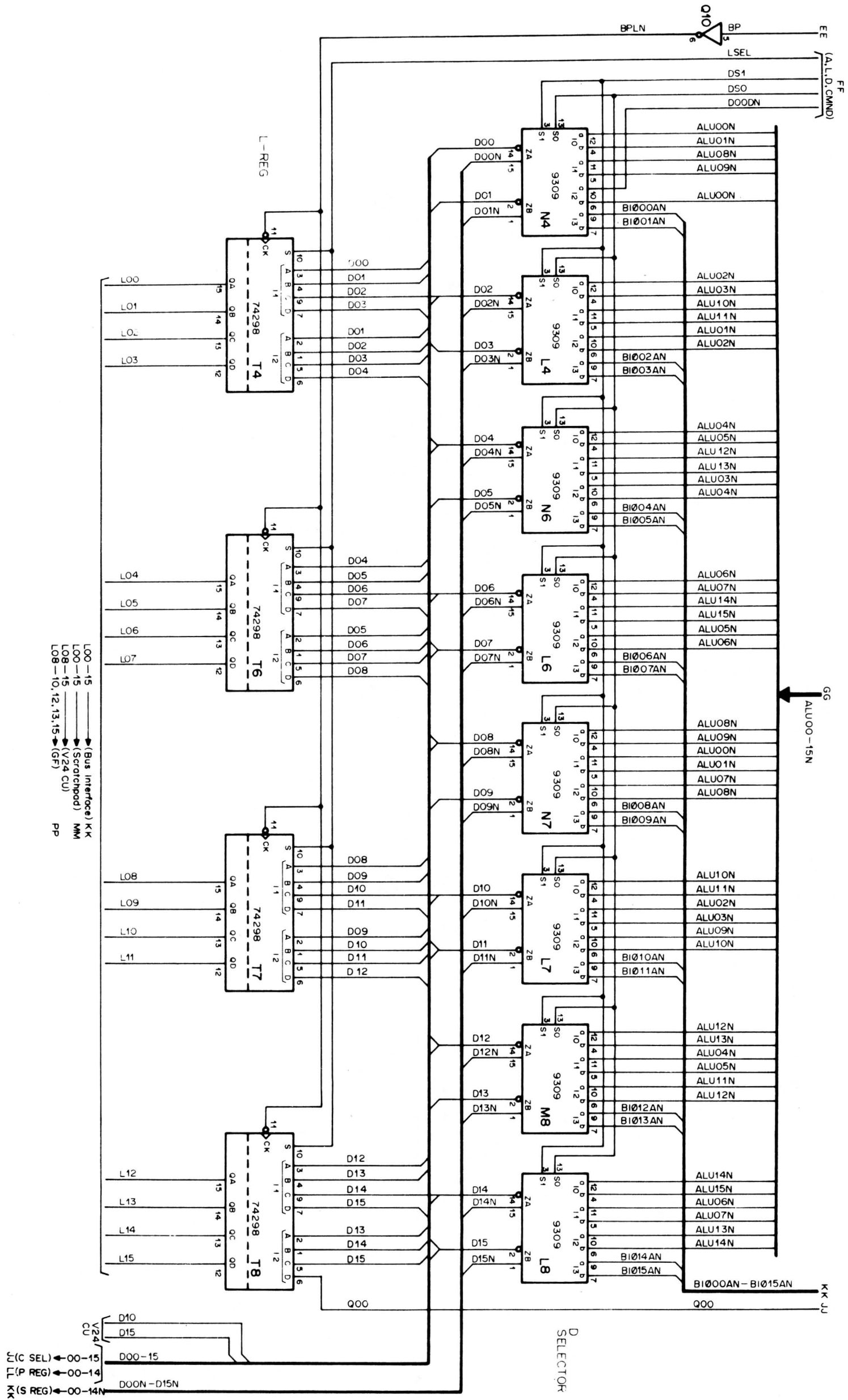


Figure 2-8HH D-Selector, L-Register

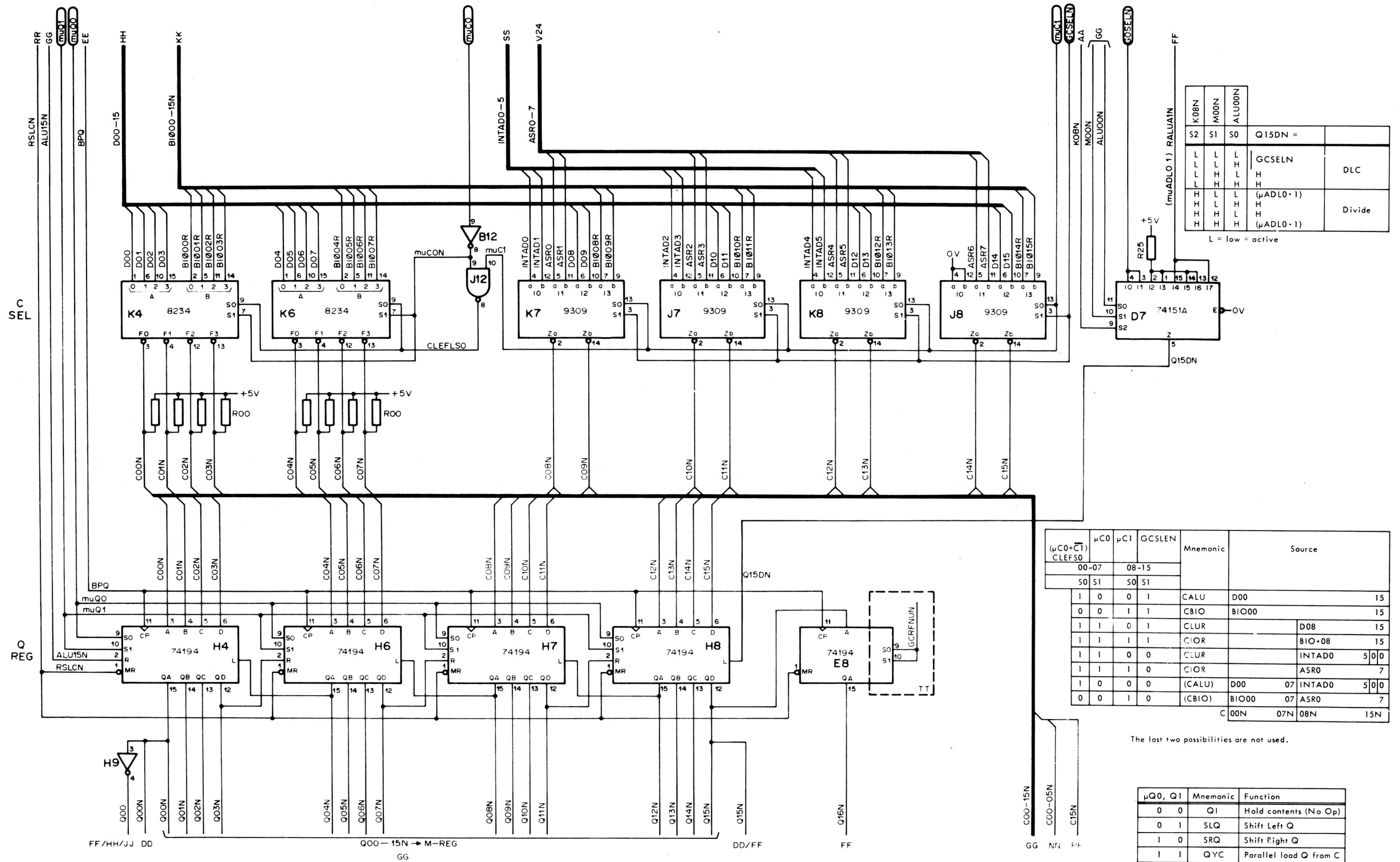
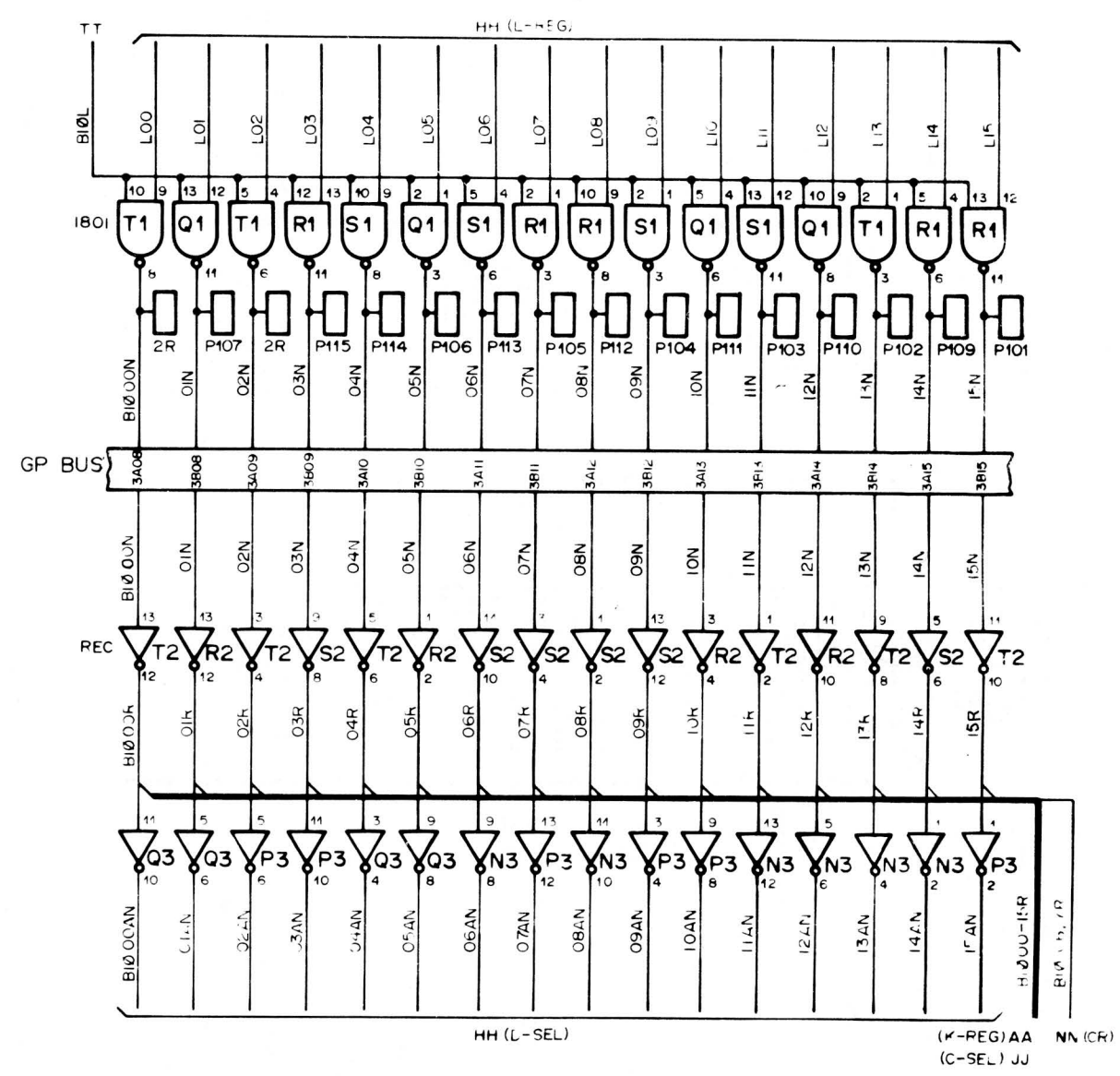
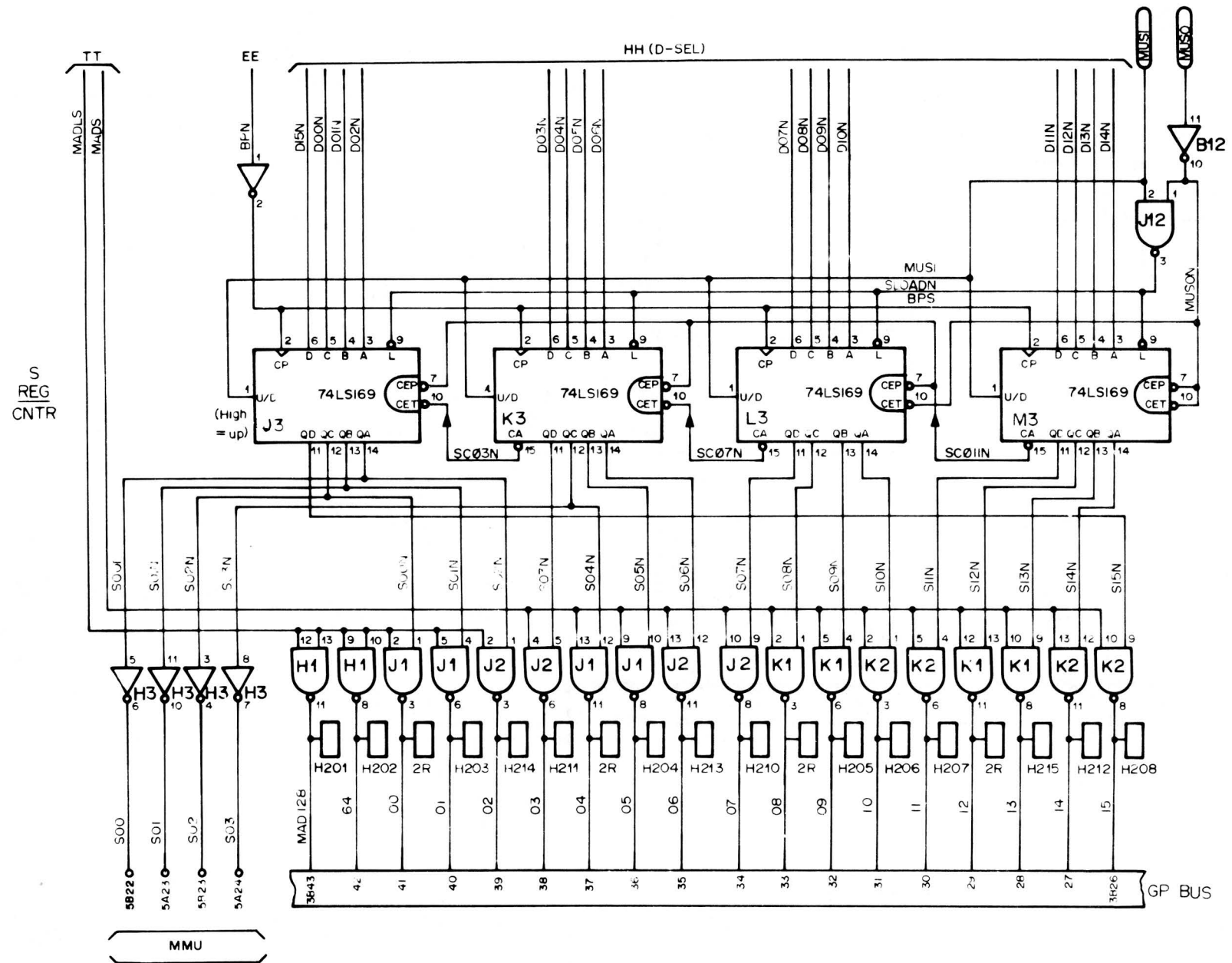


Figure 2-8JJ C-Selector, Q-Register



$\mu S0, S1$	Mnemonic	Function
0 0	SI	Inhibit counting or loading
0 1	SYD	Load S0-16N with D0-15N
1 0	SP2	S plus 2 (count down)
1 1	SM2	S minus 2 (count up)

Figure 2-8KK S-Register/Counter, Bus Interface

$\mu A0, A1, A2, A3$	9301 OUTPUT	Selected SOURCE
0 0 0 0	0 A Bus = 0	none
0 0 0 1	1 PLN	IPL
0 0 1 0	2 PN	P
0 0 1 1	3	PSW (K5-8)
0 1 0 0	4 NBN	
0 1 0 1	5 CONSTN	*
0 1 1 0	6 PUPN	
0 1 1 1	7 SYSH	
1 0 0 0	8 not connected	Scratchpad OFF
1 0 0 1	9 not connected	Scratchpad OFF
1 0 1 0	10 not connected	Scratchpad OFF

* $\mu A4-1: 00010-2_{10} (/002)$
 $\mu A4-0: 10010 18_{10} (/012)$

8234 CHIPS

S0	S1	
L	L	B
L	H	B
H	L	A
H	H	High Out (Off)

$\mu P0, P1$	Mnemonic	Function
0 0	PI	Inhibit counting or loading
0 1	PYD	Load P0-14 with D0-14
1 0	PM2	P minus 2 (count down)
1 1	PP2	P plus 2 (count up)

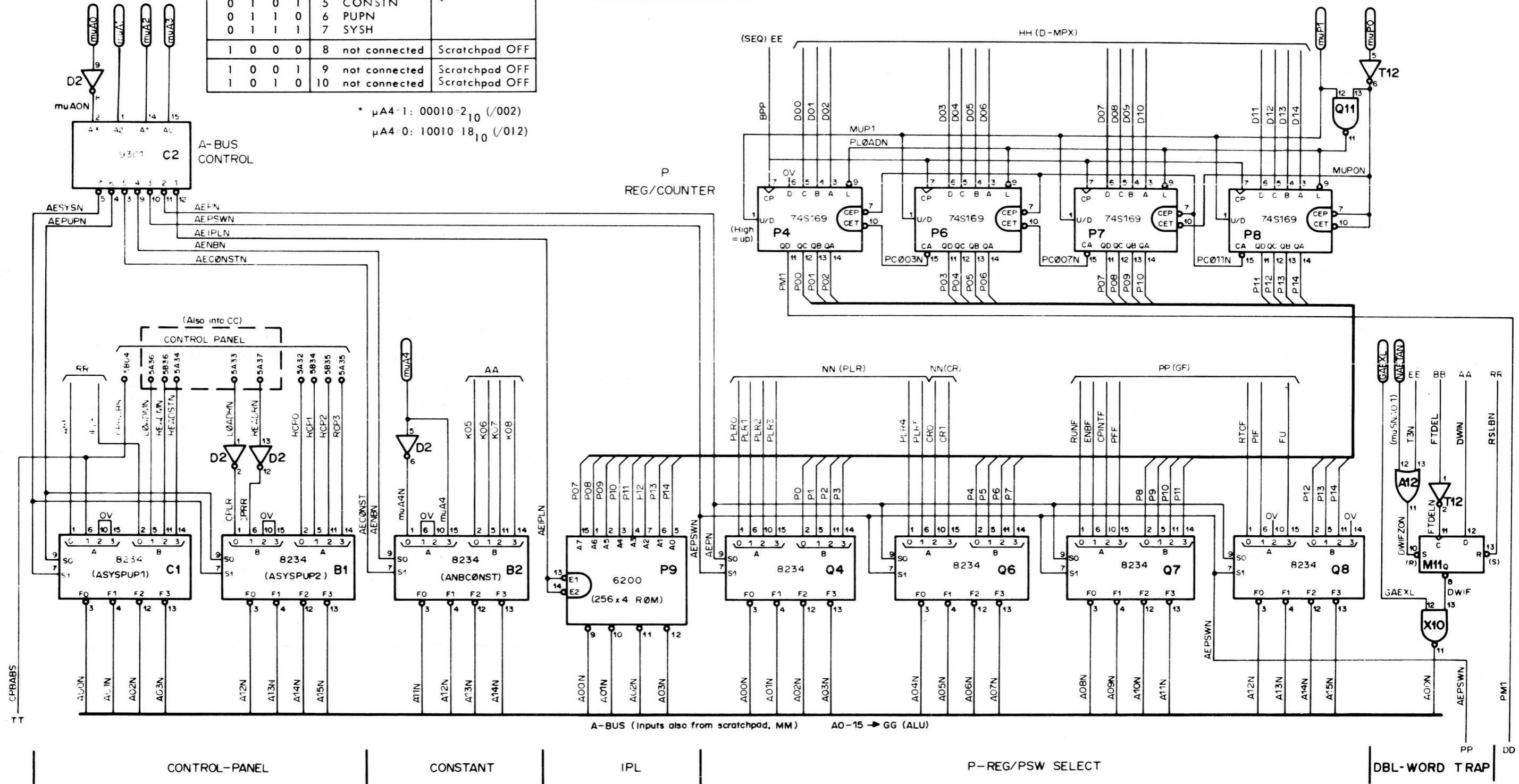


Figure 2-8LL A-Bus Selection, IPL, P-Register/Counter

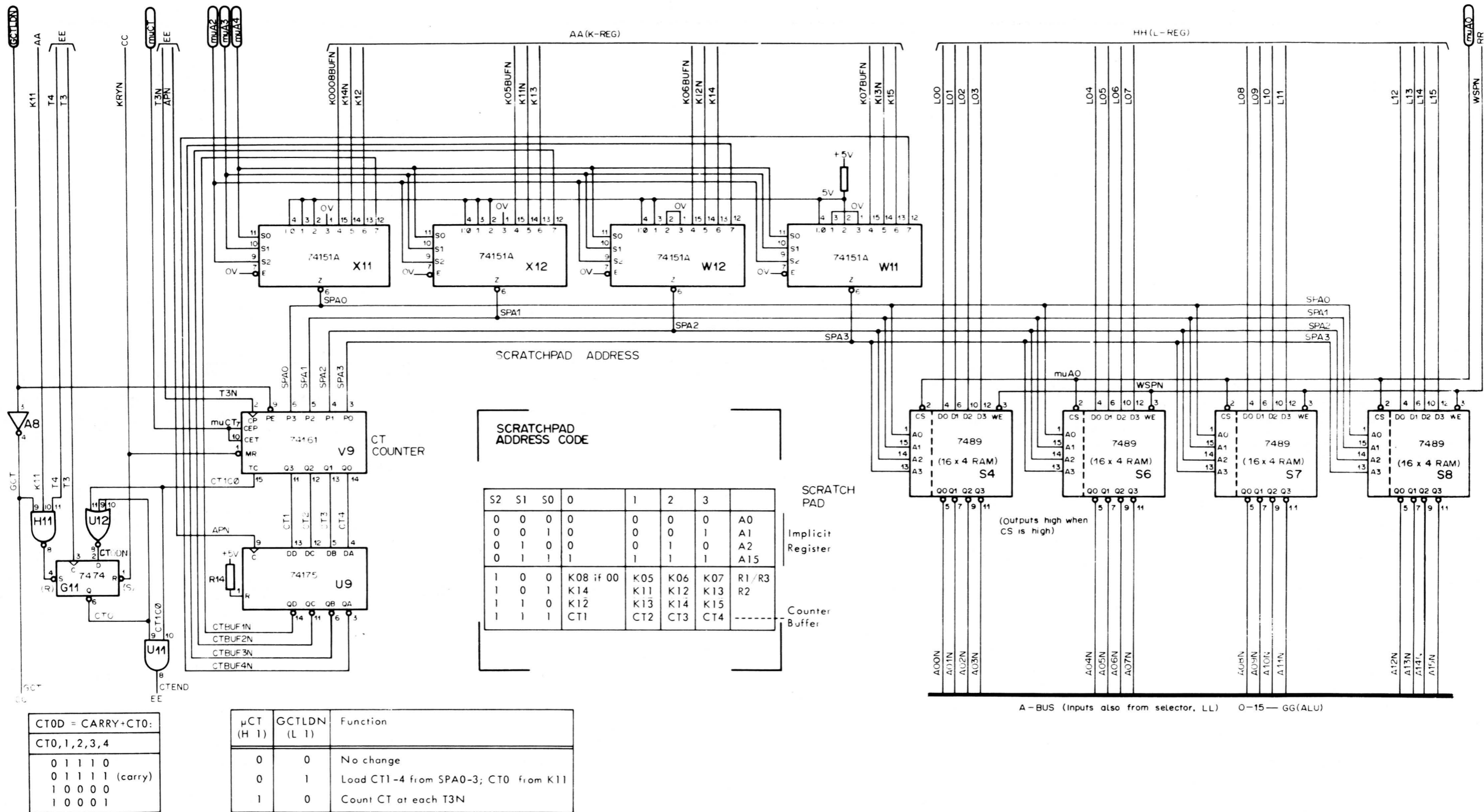


Figure 2-8MM Scratchpad

74157		74151				Function		
Input	Select A00	In	Select K01	ALU00	A01			
(+5V)	1	0 (11)	0	0	1	1	overflow OK	Shift
μMLOAD	1	1 (10)	2	0	0	1		
μMLOAD	0 (11)	1 (10)	1	0	1	0	overflow	Divide
GCRDSRN	1	1 (10)	3	0	0	0	OK	
GCRDSRN	0 (11)	1 (10)	4	1	1	1	overflow	
OVFMUN	1	1 (10)	5	1	1	0	OK	
(+5V)	1	0 (11)	6	1	0	1	OK	
GCRDSRN	1	1 (10)	7	1	0	0	overflow	

OVFMUN = GMULTI, ALUS0, M00

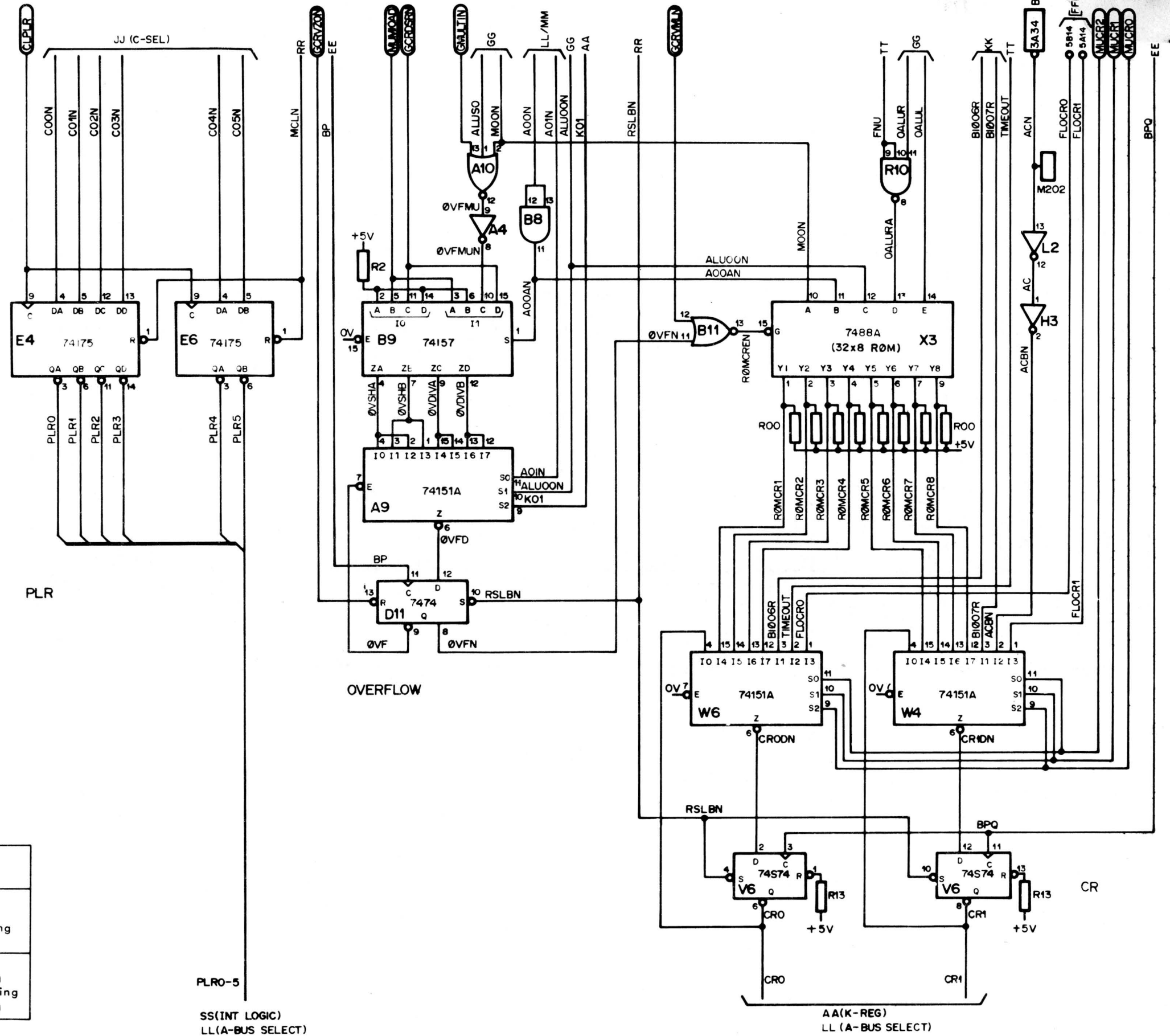
CR INPUT ROM CODE

ROM Addressing					ROM Output to CR Input Selector					
OALUL	OALUR	ALU00N	A00N	MOON	LOGIC	ADD	SUB	COMP		
					CR1,5	CR2,6	CR3,7	CR4,8		
0	0	0	0	0	1	0	1	0	1	0
0	0	0	0	1	1	0	1	0	1	0
0	0	0	1	0	1	0	1	1	0	1
0	0	0	1	1	1	0	1	1	0	1
0	0	1	0	0	0	1	1	0	1	0
0	0	1	0	1	0	1	0	1	1	1
0	0	1	1	0	0	1	0	1	1	1
0	0	1	1	1	0	1	0	1	1	1
0	1	0	0	0	1	0	1	0	1	0
0	1	0	0	1	1	0	1	0	1	0
0	1	0	1	0	1	0	1	1	0	1
0	1	0	1	1	1	0	1	1	0	1
0	1	1	0	0	0	1	0	1	1	1
0	1	1	0	1	0	1	0	1	1	1
0	1	1	1	0	0	1	0	1	1	1
0	1	1	1	1	0	1	0	1	1	1
1	0	0	0	0	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1	1
1	0	1	0	0	0	1	1	0	1	0
1	0	1	0	1	0	1	0	1	1	0
1	0	1	1	0	0	1	0	1	0	1
1	0	1	1	1	0	1	0	1	0	1

not used

CR INPUT SELECTION

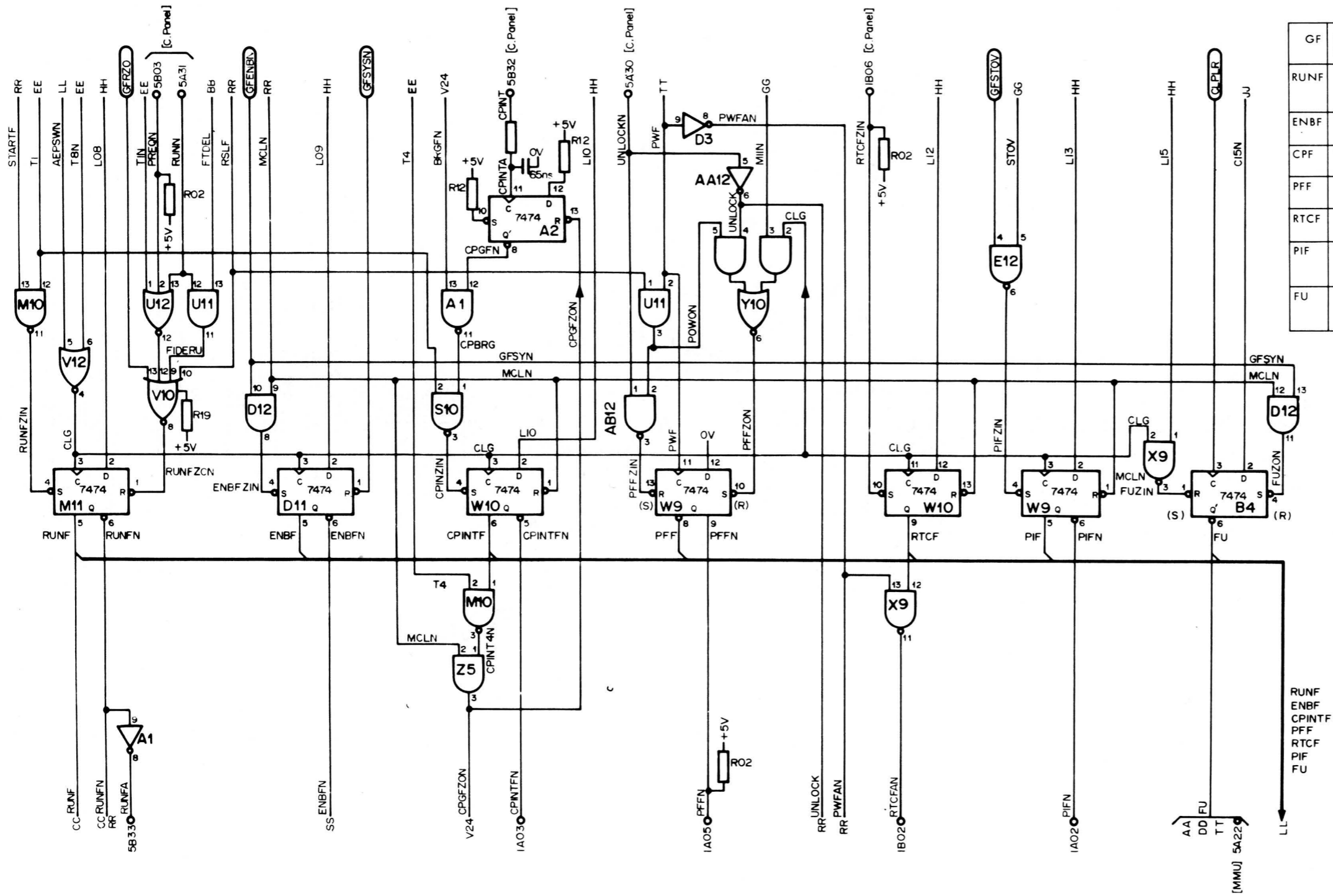
Selection μCR0,1,2	Mnemonic	Input to CR		Function
		CR0	CR1	
0 0 0	CRNC	CR0	CR1	No change
0 0 1	CRRTN	BIO06R	BIO07R	CR load during Return
0 1 0	CRIO	Timeout	ACBN	Programmed channel loading
0 1 1	CRFLO	FLOCRO	FLOCRI	Floating Point loading
1 0 0	CRLOG	ROMCR1	ROMCR5	Logic operation loading
1 0 1	CRADD	ROMCR2	ROMCR6	Addition operation loading
1 1 0	CRSUB	ROMCR3	ROMCR7	Subtraction operation loading
1 1 1	CRCMP	ROMCR4	ROMCR8	Compare operation loading



PLR0-5
SS(INT LOGIC)
LL(A-BUS SELECT)

AA(K-REG)
LL(A-BUS SELECT)

Figure 2-8NN PSW, PLR, CR



GF Bit	PSW Bit	Function	Loaded at CLG (set or reset)	Set by	Reset by
RUNF	8	Main CPU status pointer	L08	STARTF.T1	GFRZO+(PREO.RUN.T1)+(FTDEL.RUN)+.SLF
ENBF	9	Enable interrupts	L09	GFENBN.MCLN	GFSYSN
CPF	10	Operator's Interrupt	L10	(CPINT+V24INT)+T1	MCLN
PFF	11	Power Failure Interrupt	M11 resets	PWF	PWF.RSLF.UNLOCK
RTCF	12	Real Time Clock Interrupt	L12	RTCFZIN from ALU	MCLN
PIF	13	Op-Code (Program) Interrupt	L13	GSTOV.STOV from C Panel	MCLN
FU	15	User Mode (0=System Mode)	CLPLR.C15	L15.CLG	GFSYSN+MCLN

Figure 2-8PP PSW -- GF (General Flip-Flops)

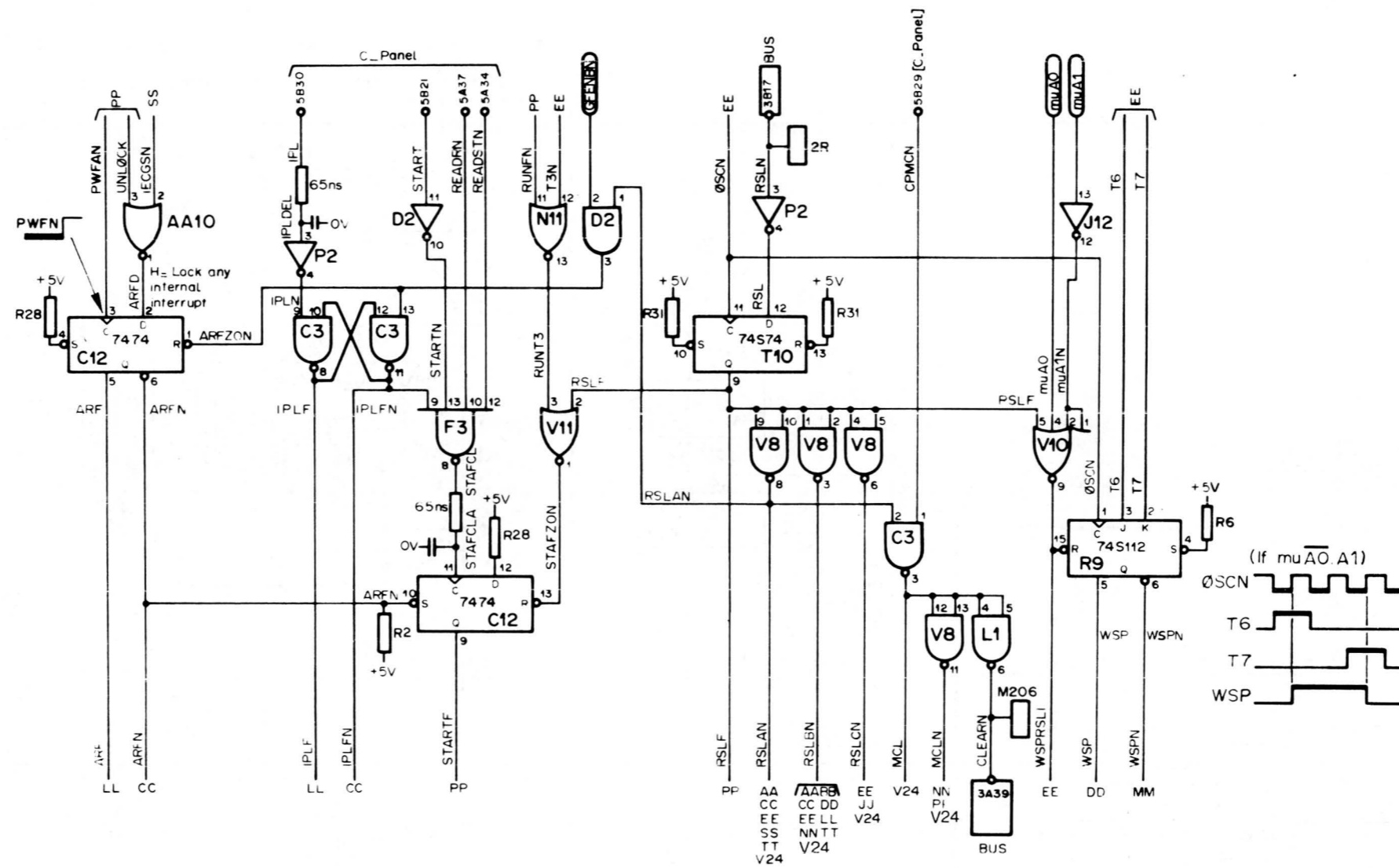


Figure 2-8RR Start, Reset

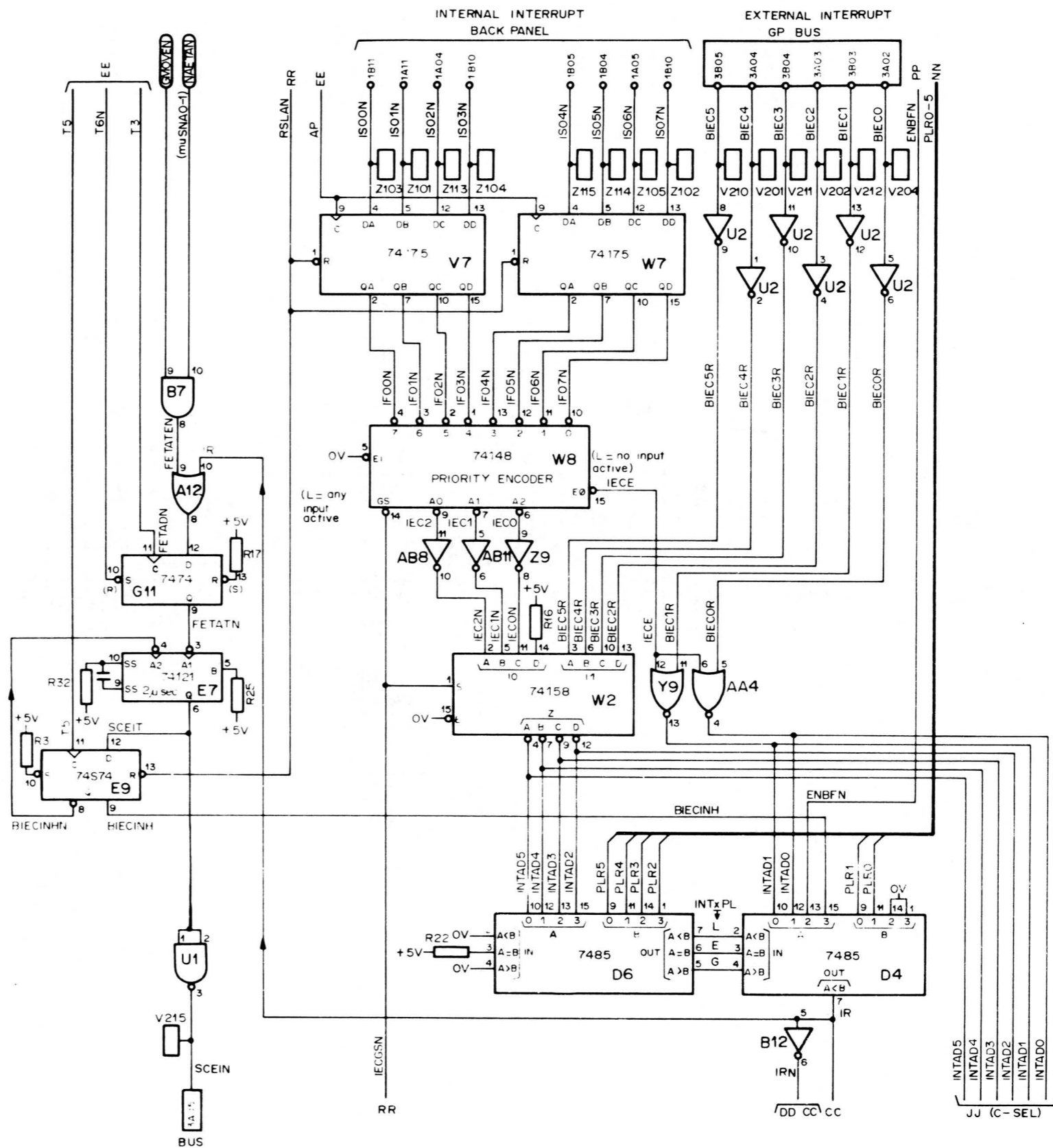
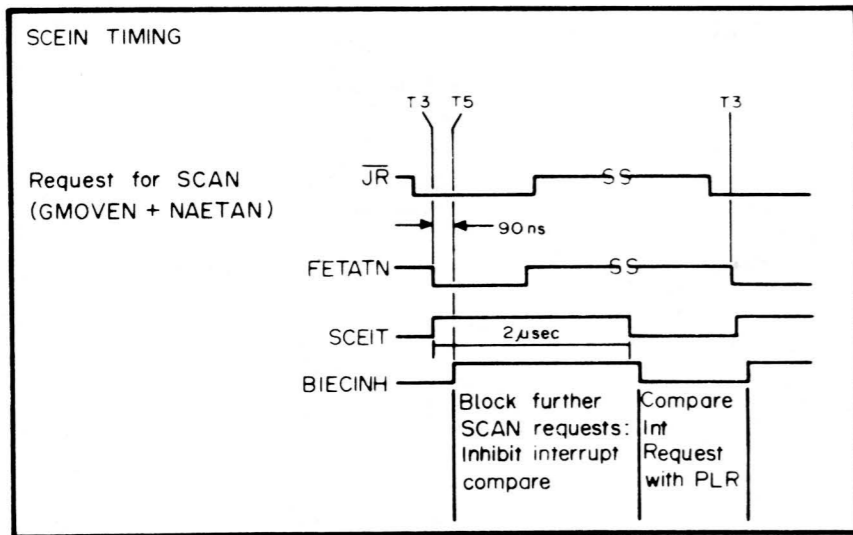


Figure 2-8SS Interrupt Logic

SECTION III

OPERATION AND TESTING

3.1 This section contains all control panel information and system testing information. The testing is divided into two parts : automatic testing via the control panel with the built-in hardware microdiagnostic, and test programs.

3.2 CONTROL PANEL

The P857M is provided with the Extended Control Panel (Figure 3-1). The top half of the panel contains ADDRESS controls; the bottom half, which is identical to the P852M/856M control panels, contains DATA controls and the CPU operating controls. When the computer is running, the BIO lines are displayed on the DATA lamps and the MAD lines are displayed on the ADDRESS lamps. When the computer stops, both the address and the contents of the next instruction are displayed. The ADDRESS half of the Extended Control Panel provides debugging facilities with the following functions :

- Display or load memory in the whole range of addresses (from 0 to 128K).
- Halt on a preset memory address.

All controls and indicators on the panel are described in Table 3-1.

3.3 The P856M is provided with the Standard Control Panel (Figure 3-1). This panel is identical to the P852M panel, with the exception of the TEST position for the key switch. The TEST function operates the microdiagnostic test routines. The P856M Standard Control Panel lacks the ADDRESS controls and the halt-on-preset-address feature.

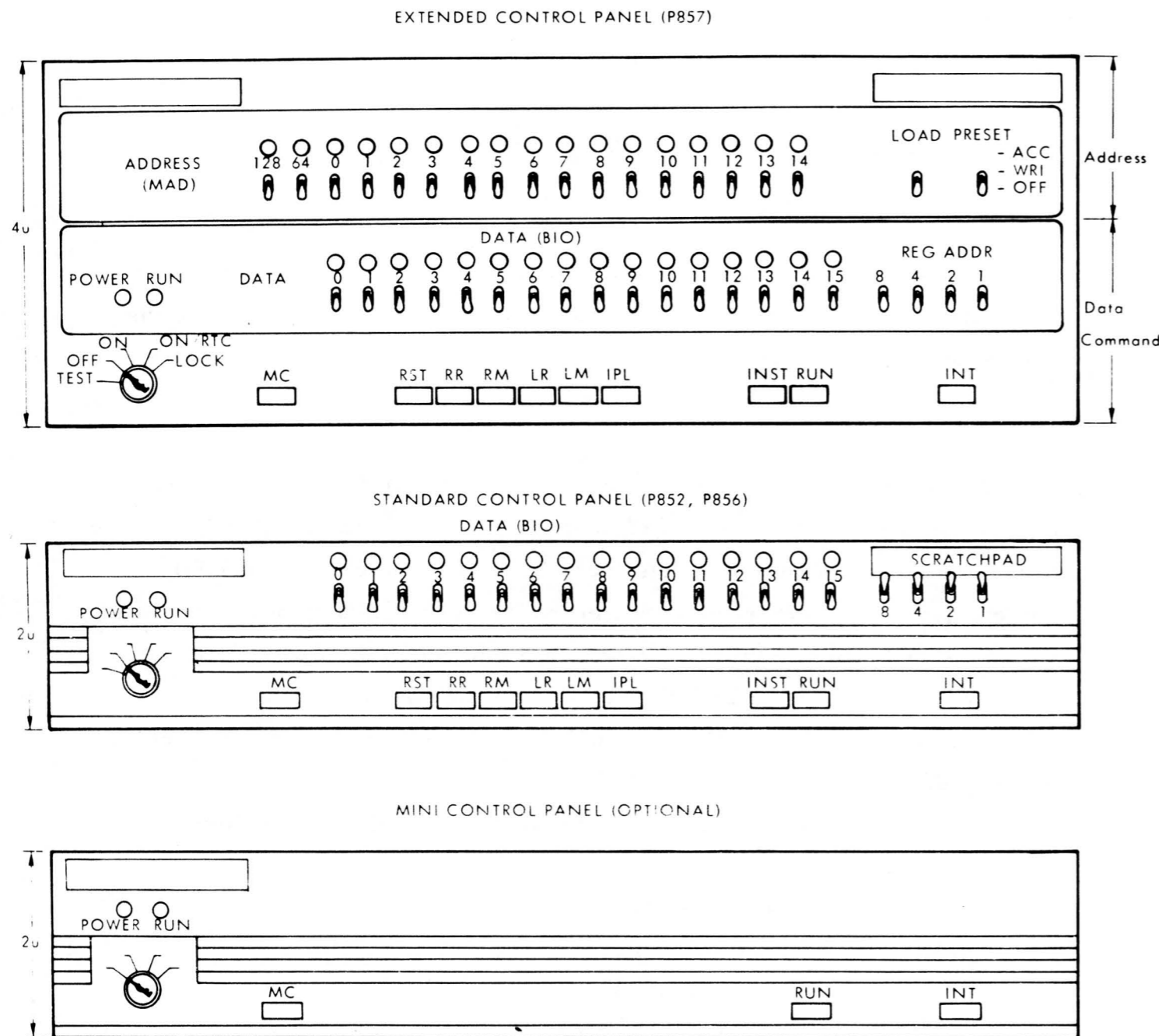


Figure 3-1 Control Panels

3.4 A mini control panel (Figure 3-1) is provided as an option for either the P856M or P857M systems.

3.5 Key and Command Operations

Control panel operations within the CPU operation sequence is shown in Figure 1-8 (Machine-State Pointer, Control Panel Operations). Each of the control-panel sequences shown on the flow diagram is described briefly in Table 3-1. The system is switched on by setting the key switch to ON, ON RTC, or LOCK. The LOCK position disables all control panel command switches except interrupt, which is still operable with the control panel locked. With the control panel switched ON or ON RTC, the CPU is set to the Run state by pressing RUN or IPL. Pressing any of the command switches MC, INST, RST, RR, RM, LR, LM will reset the Run state. When the key switch is set to TEST, the computer is set to Diagnostic mode, and the command switches then have different functions to those listed in Table 3-1.

3.6 Address Operations (P857 only)

The ADDRESS half of the control panel operates through an internal address register with incrementing logic. During normal running (PRESET at OFF), the ADDRESS lamps display the Bus MAD lines. When the computer stops, the ADDRESS lamps display the address of the next instruction, which is in the program counter (P), the address register (S), and on the MAD lines.

3.7 For either a read memory (RM) or load memory (LM) operation (Table 3-1), the ADDRESS lamps display the address placed in the control-panel address register by the panel switches. The address is first loaded by setting up the switches and then pressing LOAD ADDR. Thereafter, each time RM or LM is pressed, the address register is incremented and the ADDRESS lamps display the new address. The CPU program counter (P) is not used or affected by either of these operations.

Table 3-1 Control Panel Switches and Lamps

Key Switch	
OFF/ON	Main power switch connected directly to the power supply. The power is switched on (POWER lamp lighted) for positions ON, ON RTC, LOCK, and TEST.
ON	All panel controls are enabled.
ON RTC	All panel controls are enabled, and the Real Time Clock operates (Paragraph 5.6).
LOCK	All control-panel command switches except INT are disabled.
TEST	The automatic microdiagnostic test mode is selected (paragraph 3.16).
Command Switches	
MC	Master Clear : Clears or resets most hardware logic. Activates the GP Bus signal CLEARN, and the CPU signals MCL, MCLN.
RUN	Begins the program. The RUN switch sends the momentary signal START and the flip-flop signal RUNN to the CPU RUNF logic (paragraph 2.90).
INST	Instruction Step : Each time INST is pressed, the CPU performs the one instruction indicated by the program counter (P) and then halts. INST may be used to step the computer through a program (or part of one) instruction-by-instruction. The INST switch resets the control panel RUNN flip-flop, and sends a 113µsec START signal to the RUNF logic.
RST	Read Status. The contents of the program status word are displayed on the DATA lamps (paragraph 1.47).
RR	Read Register. The contents of the scratchpad register (A0-A15) selected by the SCRATCHPAD switches are displayed on the DATA lamps.
RM	Read Memory. The contents of memory are displayed on the DATA lamps. Consecutive words can be read by repeated pressing of the RM button.

	<p>P856 : memory address is selected by the program counter (P), and P is incremented with each RM.</p> <p>P857 : memory address is selected by the ADDRESS switches. The panel address register is automatically incremented; the program counter (P) is not used or affected.</p>
LR	Load Register. The word code set on the DATA switches is loaded into the scratchpad register (A0-A15) specified by the SCRATCHPAD switches.
LM	<p>Load Memory. The word code set on the DATA switches is loaded into memory. Consecutive words can be loaded by repeated pressing of LM.</p> <p>P856 : memory address is selected by the program counter (P), and P is incremented with each LM.</p> <p>P857 : memory address is selected by the ADDRESS switches. The panel address register is automatically incremented; the program counter (P) is not used or affected.</p>
IPL	Initial Program Loader. An initial bootstrap program located in a hardware read only memory is loaded into memory word locations 00 to 063 ₁₀ (characters 00 to 7E _{hex}).
INT	Interrupt. This button generates a level-1 Interrupt Request for the Operator's interrupt. The same interrupt can be set by the I/O console via the integral serial control unit. The interrupt may be used by the operator, for example, to change the running program with information supplied by the operator.
Data	
DATA	The 16 DATA switches are used to set a data word onto the Bus BIO lines during load register (LR) and load memory (LM) operations. For all computer operations, the DATA lamps display the contents of the Bus BIO lines. When a running computer stops, the DATA lamps display the contents of the next instruction. For RR and LR operations, the DATA lamps display the contents of the scratchpad register (A0-A15) selected by

	SCRATCHPAD. For RM and LM operations, the DATA lamps display the contents of the memory address selected by the panel address register (P857) or program counter (P856).
	Address Section (P857 only)
ADDRESS	The ADDRESS switches are used to select an initial memory address for read memory (RM) and load memory (LM) operations. For all computer operations except RM and LM, the ADDRESS lamps display the contents of the Bus MAD lines, via the panel address register. When a running computer stops, the ADDRESS lamps display the address of the next instruction. For RM and LM operations, the ADDRESS lamps display the contents of the panel address register, which is loaded from the ADDRESS switches and incremented by the RM and LM operations. No control is provided for bit 15 (character selector) because the panel accesses only memory word addresses.
LOAD ADDR	When this button is pressed, the code set on the ADDRESS switches is immediately loaded into the panel address register. This address is incremented by successive RM or LM operations; the address register is reloaded from the MAD lines for any other operation.
PRESET	This switch is used to select a Stop On Address mode. The stop will occur when the MAD-line address, via the panel address register, compares with the code set on the ADDRESS switches. OFF Normal operation. Do not stop on address. ACC Stop On Address, Access. Stop when any memory operation accesses the address set on the ADDRESS switches. WRITE Stop On Address, Write. Stop when any memory operation writes at the location set on the ADDRESS switches.

	Scratchpad Registers
SCRATCHPAD	These four switches select one of the scratchpad registers (A0-A15) to be accessed by the read register (RR) or load register (LR) operation.

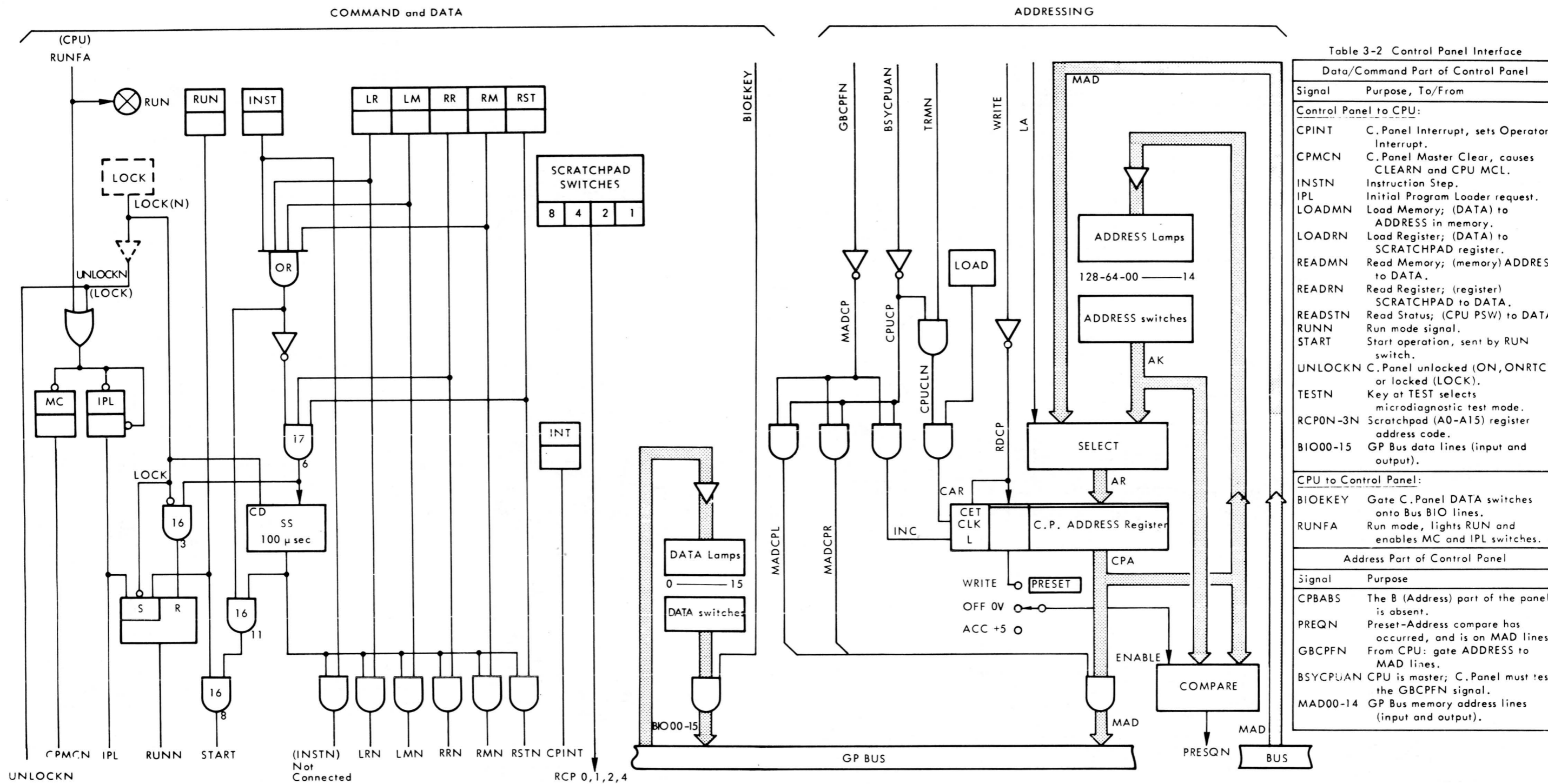
3.8 Halt on preset address is performed by setting the PRESET switch to ACC or to WRITE. The control-panel address register is loaded with the desired address set on the ADDRESS switches and LOAD ADDR pressed. In access mode (PRESET to ACC), the computer will halt whenever the memory address on the MAD lines is the same as the one set on the ADDRESS switches. In write mode (PRESET to WRITE), the computer will halt when the memory address on MAD compares with the ADDRESS switches during a memory write operation.

3.9 Interface Signals

The control panel interface signals are listed in Table 3-2, along with a brief description of each signal. All interface signals are also shown on the control panel block diagram, Figure 3-2. The control panel connector is included in the Wiring and Cabling part of Section III.

3.10 Control Panel Logic

A block diagram and a detailed schematic diagram of the control panel are provided in Figures 3-2 and 3-3.



Note 1 : With the key set to LOCK : the MC and IPL buttons are blocked ; INST, LR, LM, RR, RM, RST are blocked by the SS which is held reset ; and RUN is blocked from activating START by the reset SS.

Note 2 : Run state or Lock inhibit the MC and IPL buttons.

Figure 3-2 Control Panel Block Diagram

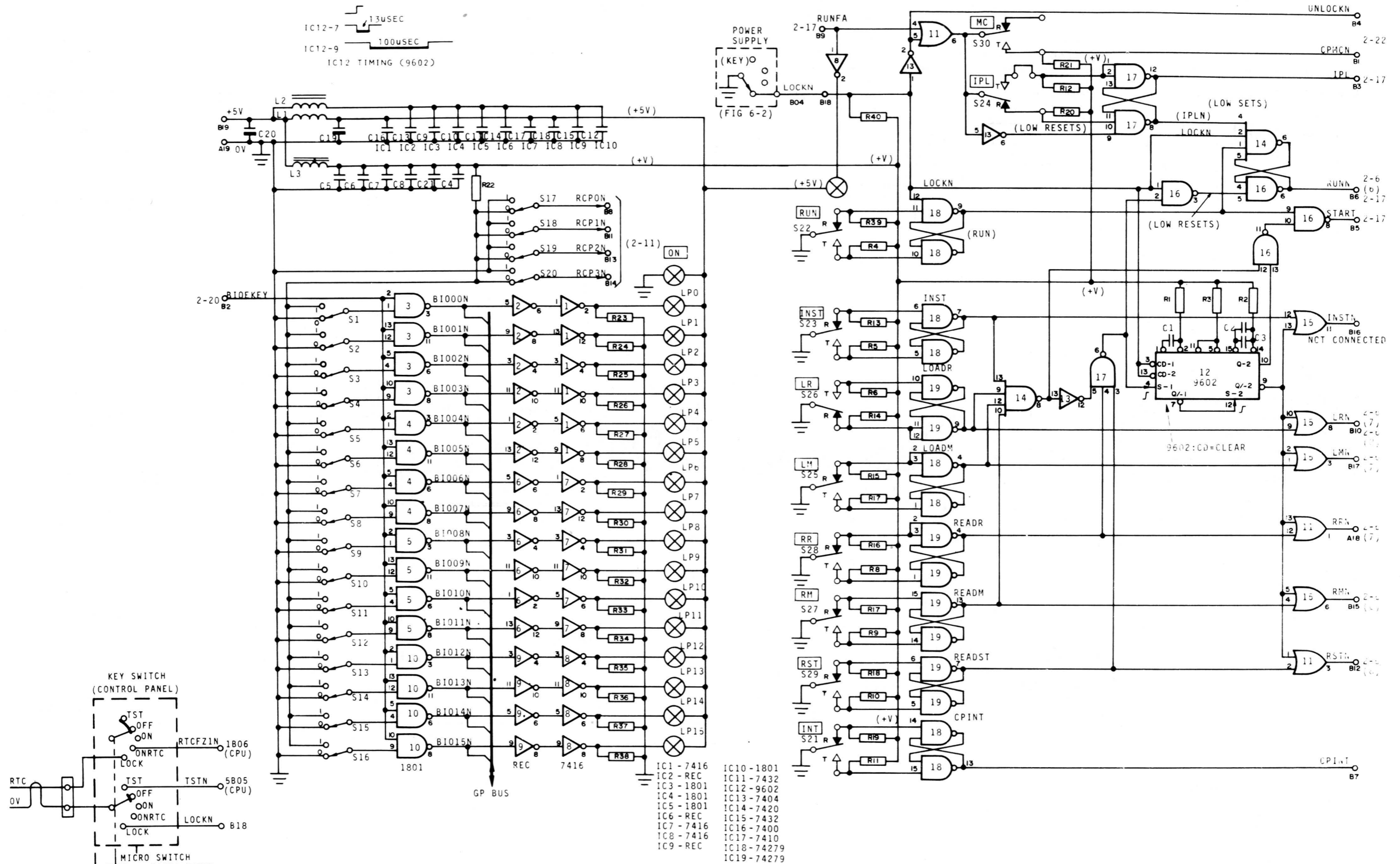


Figure 3-3A Control Panel Schematic (Data/Command Half)

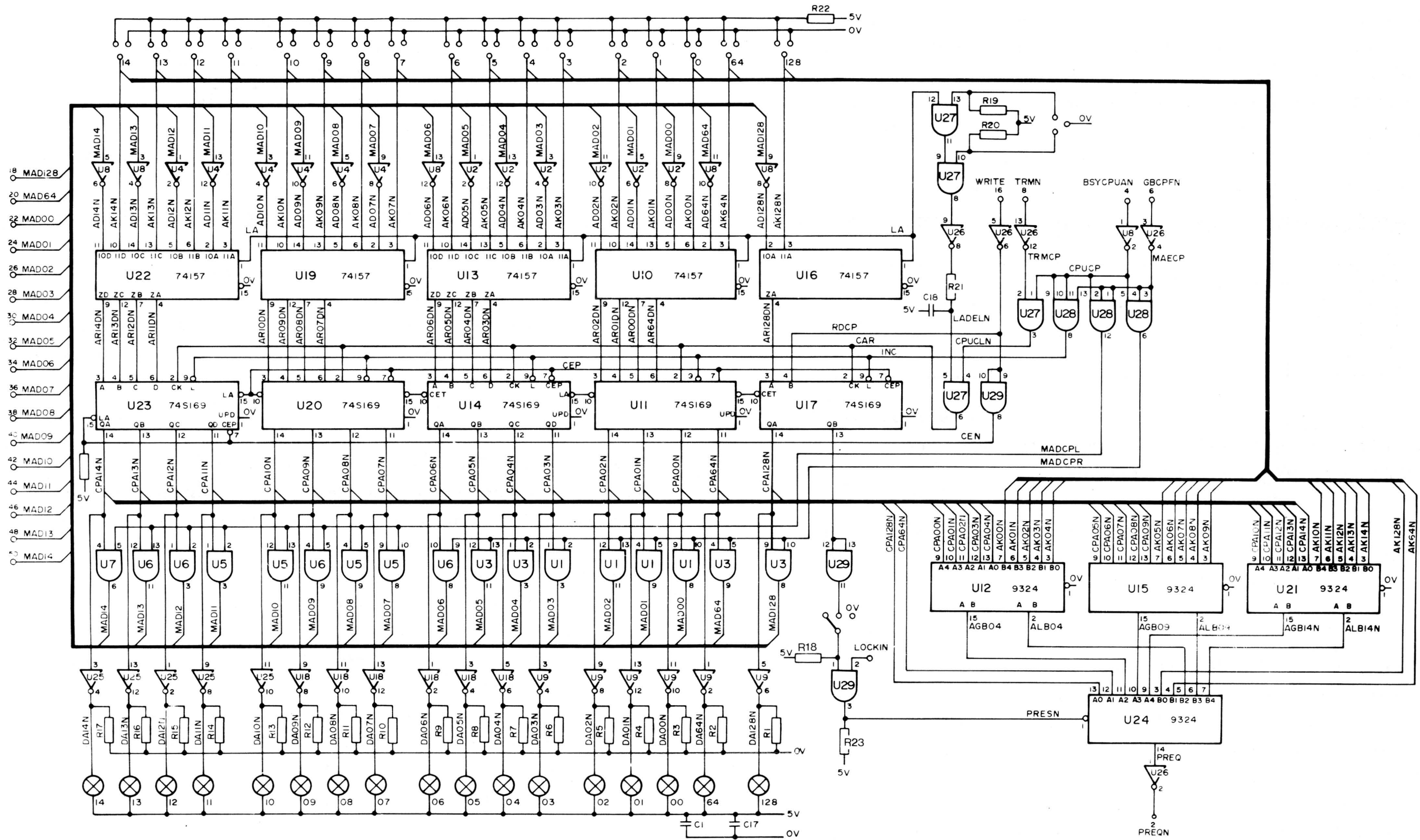


Figure 3-3B Control Panel Schematic (Address Half)

3.11 Key Switch. The Key Switch is shown on the power supply schematic, Figure 5-3. One contact of the switch controls the mains input power to the power supply. A second contact produces the enabling signal RTCFZ1N for the real time clock logic. A third contact of the Key Switch produces the test signal TSTN and the control-panel enable/inhibit signal LOCKN. The test signal is described in the Diagnostic Testing part of Section III. The Lock signal is used to disable all the control-panel command switches except INT. The following table is a review of the Key Switch functions.

Key Switch	Power	Real Time Clock	UNLOCKN Signal	Conditioning for :
OFF	off	--	--	RUN, INST, LR, LM, RR, MCL, RM, RST, IPL
ON	on	--	low	
ON RTC	on	on	low	
LOCK	on	on	h	

3.12 Scratchpad. The four Scratchpad switches are used to address a scratchpad register (A0-A15). The switches produce the true-logic (+5V = logic 1; 0V = logic 0) signals RCP0-4.

3.13 Data. The DATA switches are used to set the 16 bits of a data word to be placed on the GP-Bus BIO lines. The DATA switches are gated onto the BIO lines by CPU signal BIOEKEY. BIOEKEY is produced by BUSFN and BOKFN in the Bus Controller logic (paragraph 2.10).

3.14 INITIAL PROGRAM LOADING

A hardware bootstrap is provided which consists of a 256x4 bit ROM and an IPL-microprogram routine. This hardware routine organizes the bootstrap-ROM contents into 64 sixteen-bit words and loads them into memory locations 00-63. The system then has a bootstrap loaded and is able to accept a software IPL followed by a program. A basic procedure is given here for loading a program from paper tape into the system. This procedure assumes that the tape has been assembled with the software IPL, followed by the program.

Program Load Routine

- Load IPL/Program tape into paper-tape reader.
- Set DATA switches for I/O Bus and paper tape reader :

0 0 0 1 | 0 0 0 0 | 0 0 1 0 | 0 0 0 0

(The codes are shown at the top of Table 2-10, Bootstrap Listing).

- Press INST, MC, IPL
 - a. Hardware bootstrap organized by hardware IPL routine and loaded into memory 00-63.
 - b. Bootstrap then loads the low-core part of software IPL into memory 080 towards 000, partly overwriting bootstrap.
 - c. Low-core part of software IPL begins loading high-core IPL into memory nnn (decision of IPL based on memory size) towards 000.
 - d. High-core IPL begins loading program, deciding where to start and loading towards memory 000. Interrupt Start Addresses are loaded at this time in locations 63-00, overwriting bootstrap. Part of IPL is also overwritten.

3.15 Bootstrap Test

The hardware bootstrap can be loaded into memory for testing (comparing with table 2-10) without being overwritten by the software IPL. This is done by performing the Program Load Routine without a device connected (Paper-tape reader off, disc off, etc.), as follows :

- Set DATA switches for I/O Bus and reader.
- Press INST, MC, IPL.
 - a. Hardware bootstrap organized by hardware IPL routine and loaded into memory 00-63.
 - b. CPU stops after bootstrap is loaded.
- Load highest memory address into P-register (DATA + reg=0).
- Press INST which increments P to 000.
- Begin pressing RM, and copy bootstrap from DATA lamps.

3.16 MICRODIAGNOSTIC TESTS

Diagnostic test routines are included in the CPU microinstruction control ROM. Approximately 100 words of the CPU control store are allocated for microdiagnostic routines. The complete microdiagnostic tests about 70% of the CPU hardware (Figure 3-4). These microdiagnostics are performed, with the system off line, by setting the control-panel key switch to TEST. The microdiagnostics are arranged sequentially from very basic data-path and register tests to complex instruction operations (Figure 3-5). The different methods of running the tests, in the suggested order, are :

- Automatic Go/No-Go Mode. The instruction-tests DLA and RB are executed and a final display is given. The memory and CPU/CU test is executed and a final display is given. Correct results in this mode indicate correct machine operation.
- Test-By-Test Mode. The operator steps sequentially through logic tests T1, T2, T3, the memory and CPU/CU tests M, and the instruction tests R, and obtains a display indication at each stage. The Chained Tests mode provides further exercise of the same sequence of tests.
- Basic Tests. The operator steps sequentially through basic tests of the control panel, L register, M register, and Q register, exercising all DATA switches and lamps for each test. This checks the basic registers and data paths.

3.17 Test Procedure, Basic Tests

- Key switch to TEST. (Loops through control-panel test.)
- Operate each DATA switch.
- oo Corresponding DATA light should light.
- Press LR. (Loops through L-Register test.)
- Operate each DATA switch.
- oo Corresponding DATA light should light.
- Press LR. (Loops through M-Register test.)
- Operate each DATA switch.
- oo Corresponding DATA light should light.
- Press LR. (Loops through Q-Register test.)

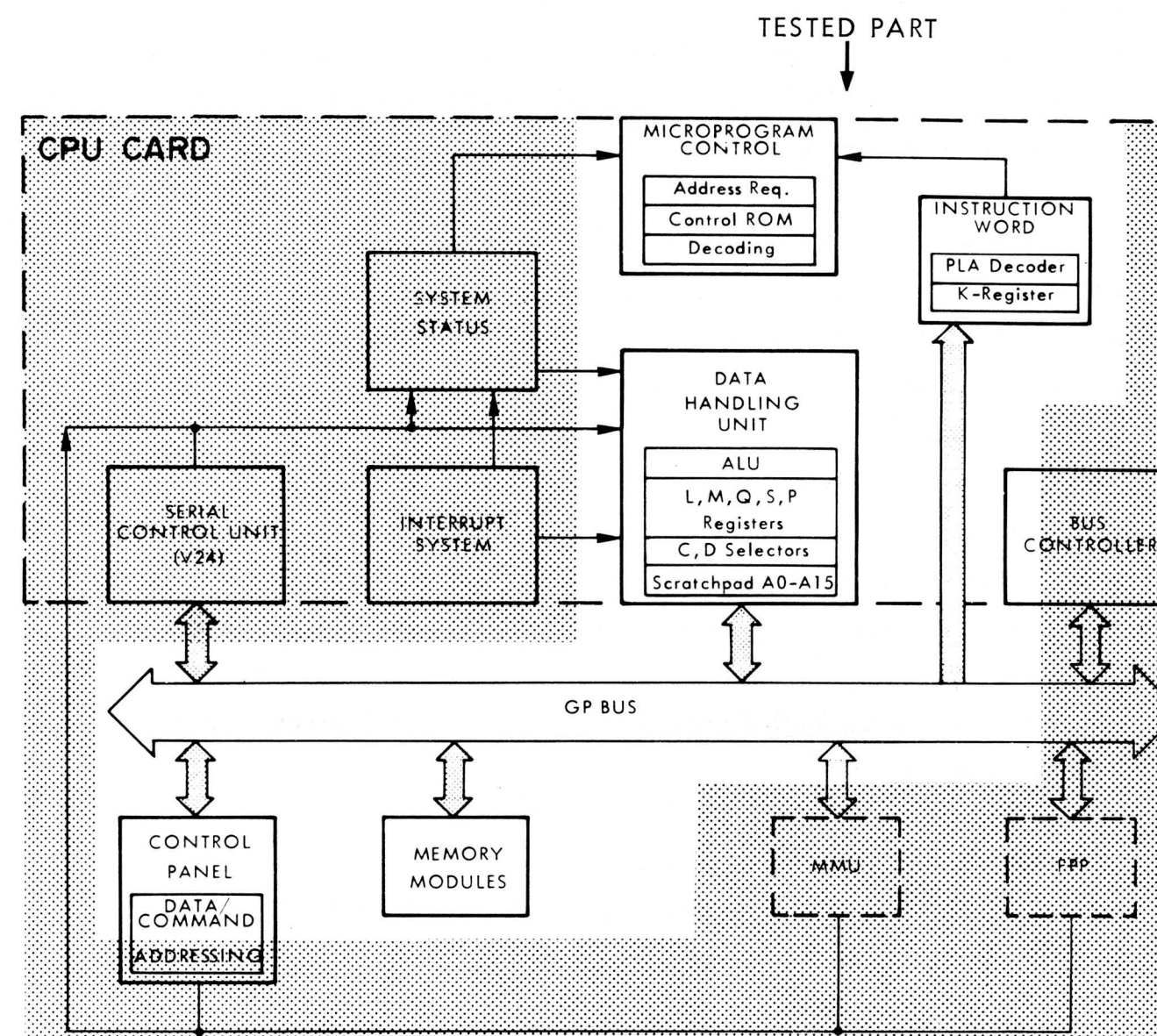


Figure 3-4 CPU Logic Tested by Microdiagnostics

- Operate each DATA switch.
- oo Corresponding DATA light should light.

The test is now looping through the Q-Register test. To continue to the next series of tests, set the DATA switches as required (refer to Automatic Mode or Test-by-Test Mode) and press LR or RUN.

3.18 Test Procedure, Automatic Mode (Go/No-Go)

- Key switch to TEST. (Loops through control-panel test.)
- DATA-switch 0 to 0 position. (Enables final display.)
- RUN. (Instruction test performs DLA and RB and then loops at end-of-test display.)
- oo Wait for display 000000000000●000. If error, do Test-by-Test mode to diagnose fault. If correct, do Memory and CU test.
 - Set address of existing CU device on DATA switches 02-07.
 - Press LM. (Executes memory tests 1 and 2, and CPU/CU test and then loops at end-of-test display.)
- oo Wait for display 0000000000000000. If error, do Test-by-Test mode to diagnose fault.

3.19 Test Procedure, Test-by-Test Mode

- Set address of existing CU device on DATA switches 02-07.
- Key switch to TEST. (Loops through control-panel test.)
- DATA-0 to 0 position. (Enables display at each stage.)
- DATA-15 to 1. (Selects test T1 first.)
- Press LR four times. (Steps through the basic tests, performs logic test T1, and loops at end-of-test display.)
- oo Wait for display 000000000000000●.
- Press LR. (Executes test T2, loops at end-of-test display.)
- oo Wait for display 000000000000000●0.
- Press LR. (Executes test T3, loops at end-of-test display.)
- oo Wait for display 000000000000000●00.
- Press LR or RUN. (Executes instruction test R, loops at end-of-test display.)

- oo Wait for display 000000000000●000.
- Press LR or LM. (Executes memory and CPU/CU test M, loops at end-of-test display.)
- oo Wait for display 0000000000000000.

Chained tests performs the same tests in the same sequence as above, but without halt on display after each stage.

- Key switch to TEST.
- Address of existing CU device on DATA 02-07.
- DATA-0 to 1 (chained tests, no display loop).
- DATA-15 to 1 (selects test T1 first).
- Press LR four times (steps through basic tests, then loops through the five operating tests without stopping.)
- Set DATA-0 to 0 to stop the chained tests.
- oo Wait for one of the five displays.
 - Press LR to step through the tests and obtain the display for each.
 - Set DATA-0 to 1 to restart the chained-tests loop.

To restart the tests at the beginning, do :

- DATA-0 to 1.
- Key switch away from TEST (OFF is best), then back to TEST.

To Start a test out of sequence (for example, to start test T3 after an error indication for test T2) :

- Key to TEST; CU address on DATA 02-07; DATA-0 to 0.
- Set DATA 08-15 to select desired test :
 - 00000001 (/01) for logic test T1.
 - 00000010 (/02) for logic test T2.
 - 00000100 (/04) for logic test T4.
 - 00001000 (/08) for instruction test R.
 - 00010000 (/10) for memory, CU test M.

3.20 Analysing Tested Functions

3.21 Basic Tests. In the control-panel test (Figure 3-6), the operator verifies the operation of all control-panel DATA switches and lamps and the BIO

Note: 0 = lamp on
● = lamp off

data path to and from the GP Bus. In the L-register test, the data path through the CPU D-selector and L-register is added to the control-panel test. The M-register test adds the C-selector, M-register, and ALU to the data-path loop. The Q-register test adds the Q-register to the data-path loop.

3.22 These tests exercise all bit positions of the control-panel switches and lamps and the CPU L, M, and Q registers as well as the data path. Not all the functions of the registers, the C and D selectors, or the ALU are tested.

3.23 Q-Register Test. This test is not only a part of the basic tests, but also the start and the error output of the following logic, memory, and instruction tests. While looping through the Q-register test, the DATA switches content are loaded into the CPU registers to select the type of Test-by-Test operation and the number of the next test. (Test selection and incrementing is shown in Figure 3-7.) If a test has been performed and an error occurred, the test branches back to the Q-test loop and the DATA lamps display the setting of the switches rather than the correct display.

3.24 Logic Test T1. The tested functions (Figure 3-8) are :

- Q-reg. shifted left
- A-Bus selection
- Constant Two
- Qo
- ALU=0
- A OR B
- A + B
- \bar{B}

3.25 Logic Test T2. The tested functions (Figure 3-8) are :

- Q-reg. shifted right
- Constant Ten
- ALUZERO
- P-register
- P - 2
- A - B
- A + B
- crossed A

3.26 Logic Test T3. The tested functions (Figure 3-8) are :

- A-operand shifted right
- Reading and writing scratchpad, address incrementation by CT counter (particular addresses A=A2 and A=A15).
- 4 X A

3.27 Instruction Tests R. The tested functions (Figure 3-10) are :

- The K-register is loaded with the instruction code for the DLA instruction.
- Scratchpad registers A1 and A2 are loaded with known values.
- Branch to the DLA microprogram and execute DLA instruction.
- Return to Test R via CPU Idle loop and verify A1 and A2. (The CR is verified by the next instruction.)
- The K-register is loaded with the instruction code for the RB instruction.
- Address the RB microprogram with the PLA, and execute RB instruction.
- Return to Test R via CPU Idle loop and verify P-register.
- A failure in CR positioning or in CV (condition verified) circuit, or PLA addressing function should cause wrong value of P counter.

3.28 Memory Test M1. The tested functions (Figure 3-9) are :

- Bit-15 is set to 1 in all addresses of a 16k block.
- The block is read and verified.
- The pattern is shifted left 1 position (to bit 14) and the block is read and verified.
- Shifting the pattern and reading the block is continued until all bit positions have been read and verified.

3.29 Memory Test M2. The tested functions (Figure 3-9) are :

- Every word of a 16k memory block is written with its own address value.
- Each word is read and verified.

3.30 CPU/CU Dialogue Test M3 (part of Memory-test loop). The tested functions (Figure 3-9) are :

- An existing CU device address (from DATA switches 02-07) and TMP are sent on the GP-Bus MAD lines.
- The CU response TPM is verified by means of condition register (CR \neq 3).

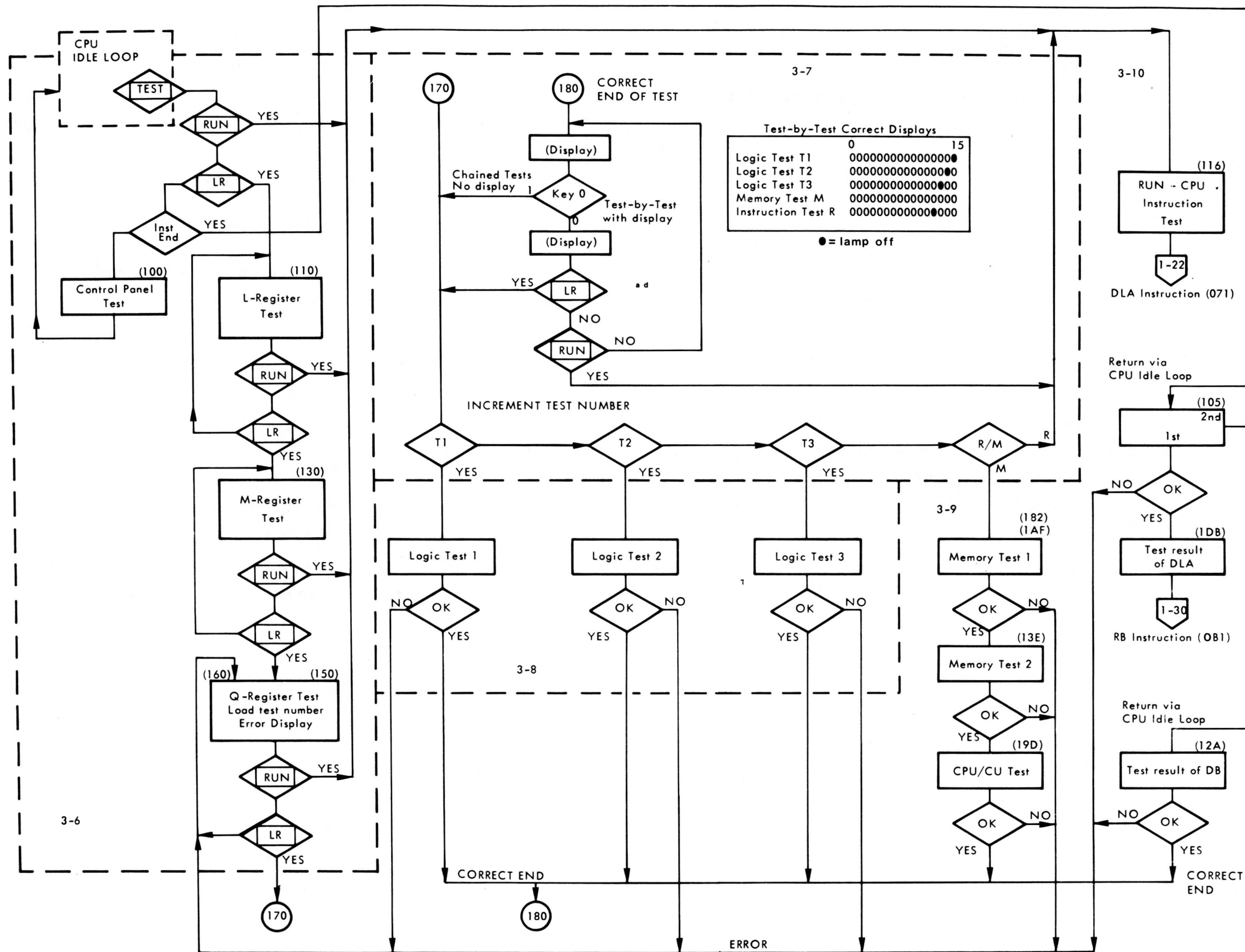


Figure 3-5 Microdiagnostics Block Diagram

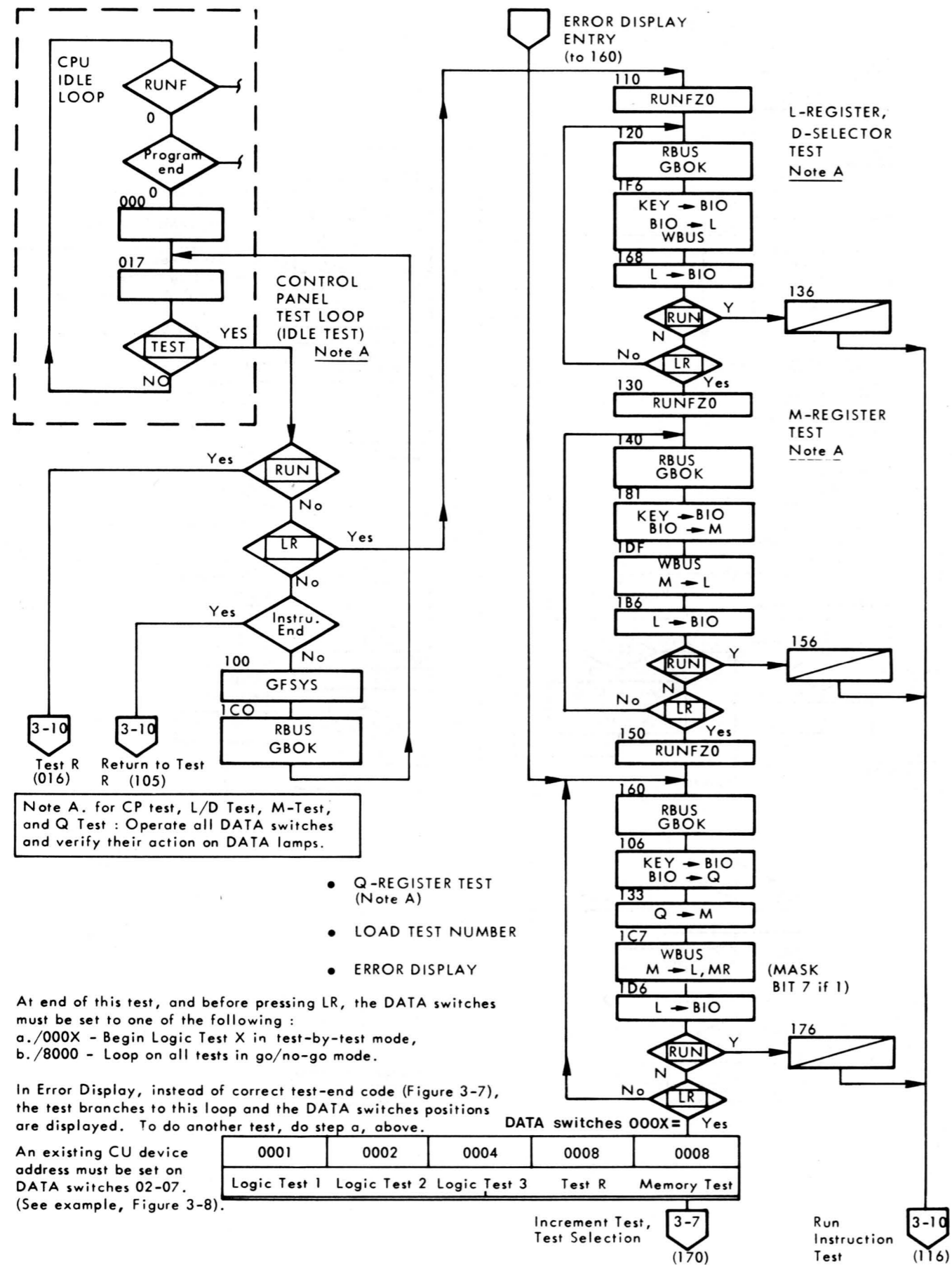


Figure 3-6 Start and Basic Tests

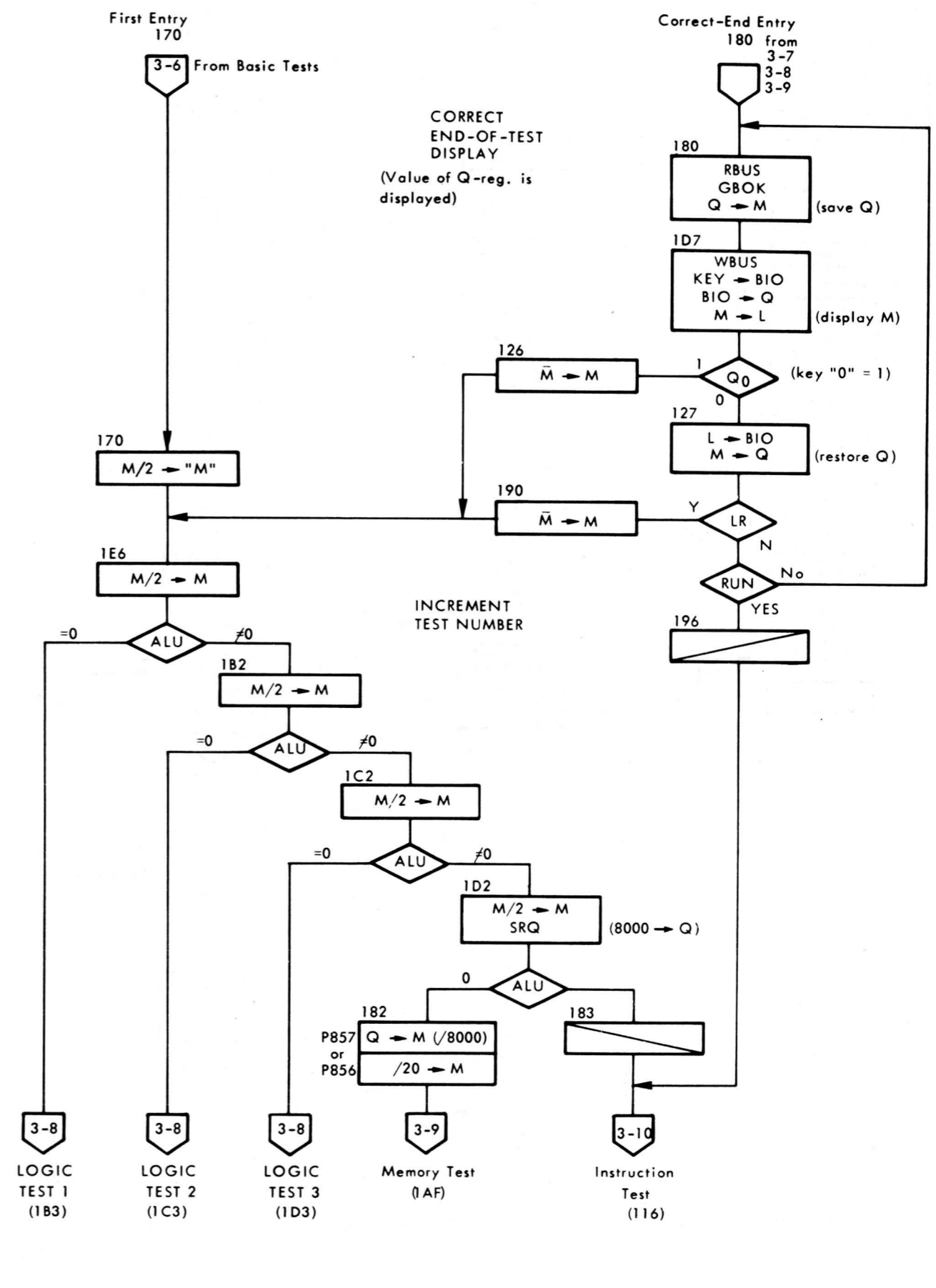


Figure 3-7 Increment Test, Test-End Display

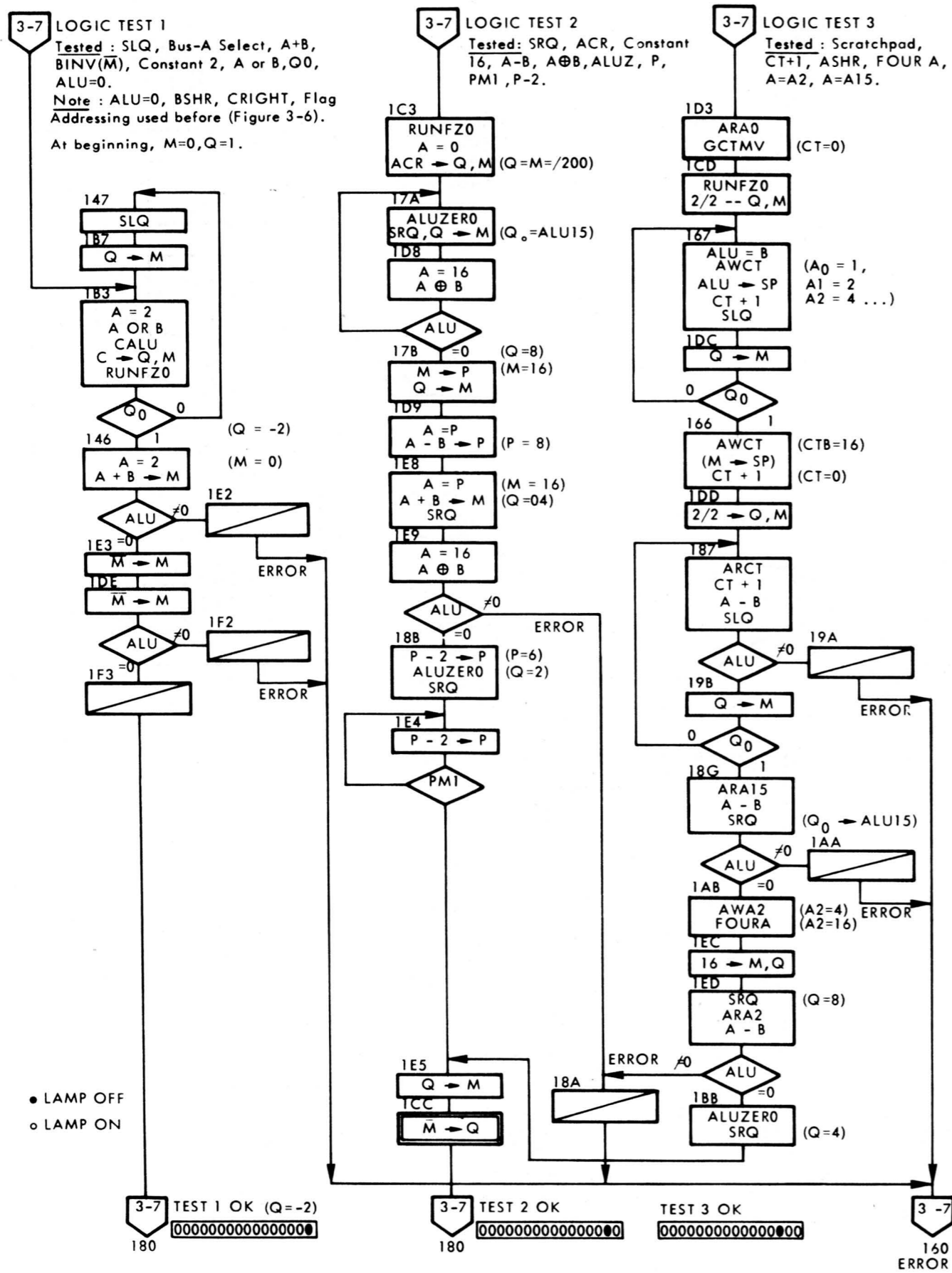


Figure 3-8 Logic Tests (T1, T2, T3)

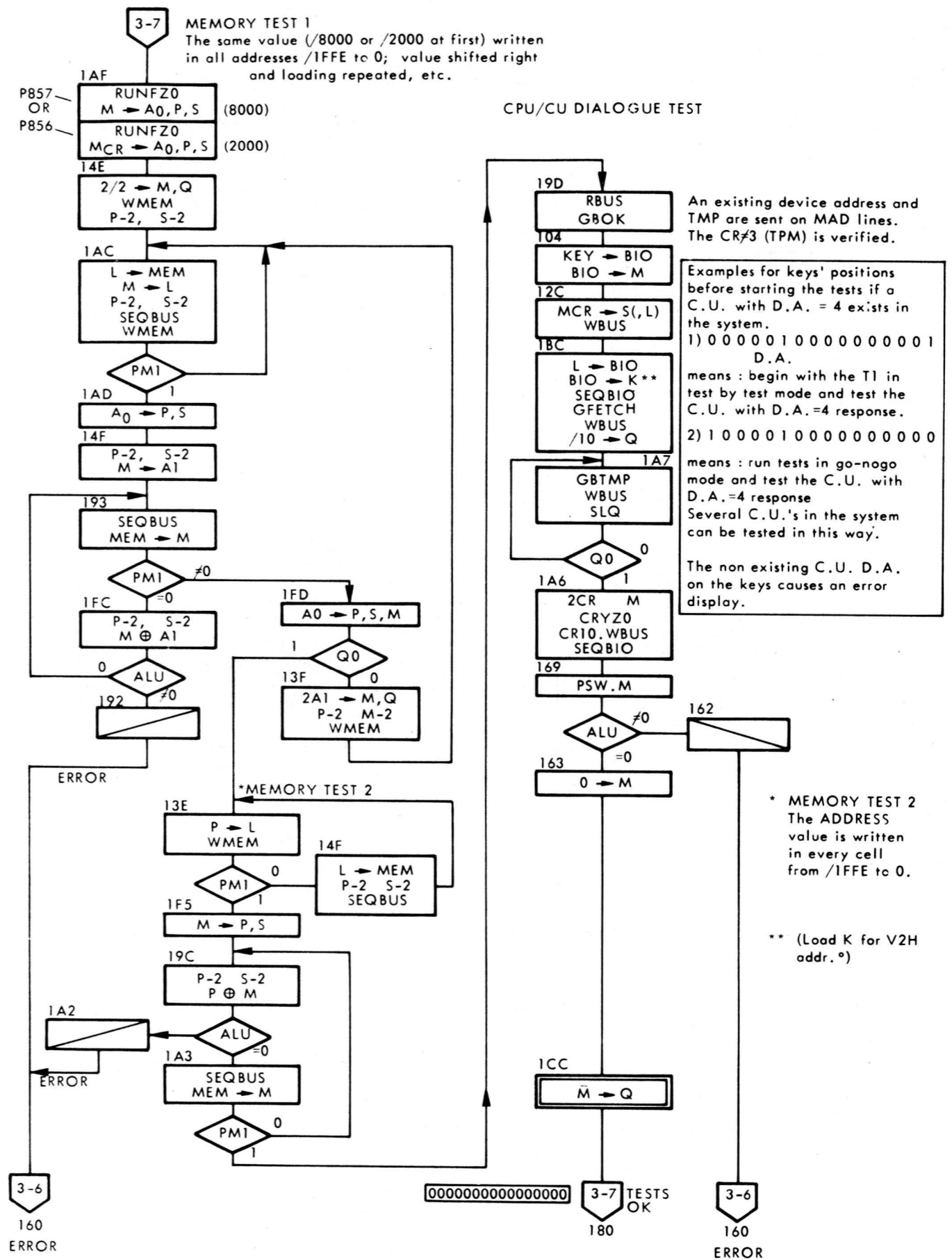


Figure 3-9 Memory and CPU/CU Dialogue Tests (Test M)

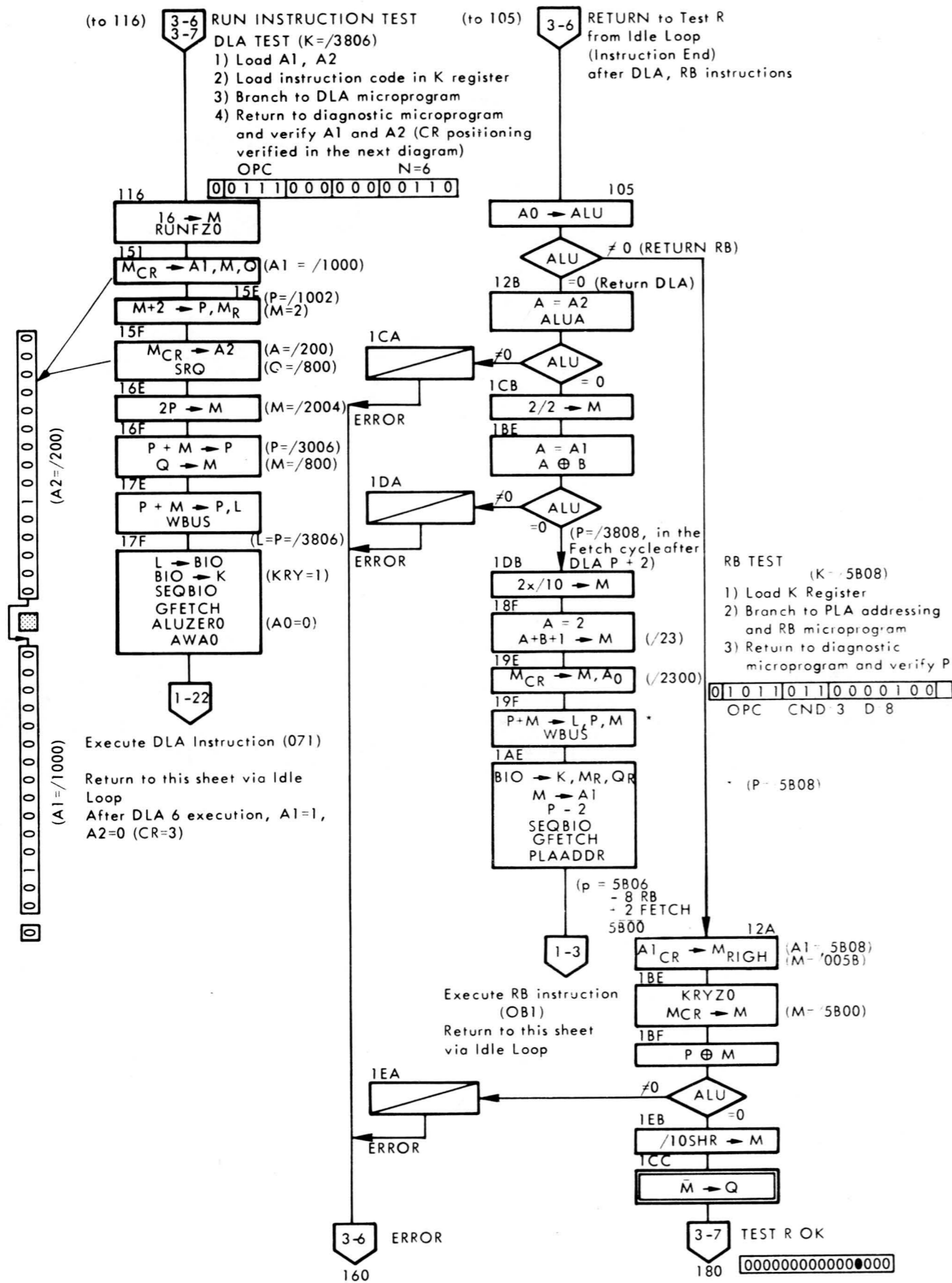


Figure 3-10 Run CPU Instruction Tests (Test R)

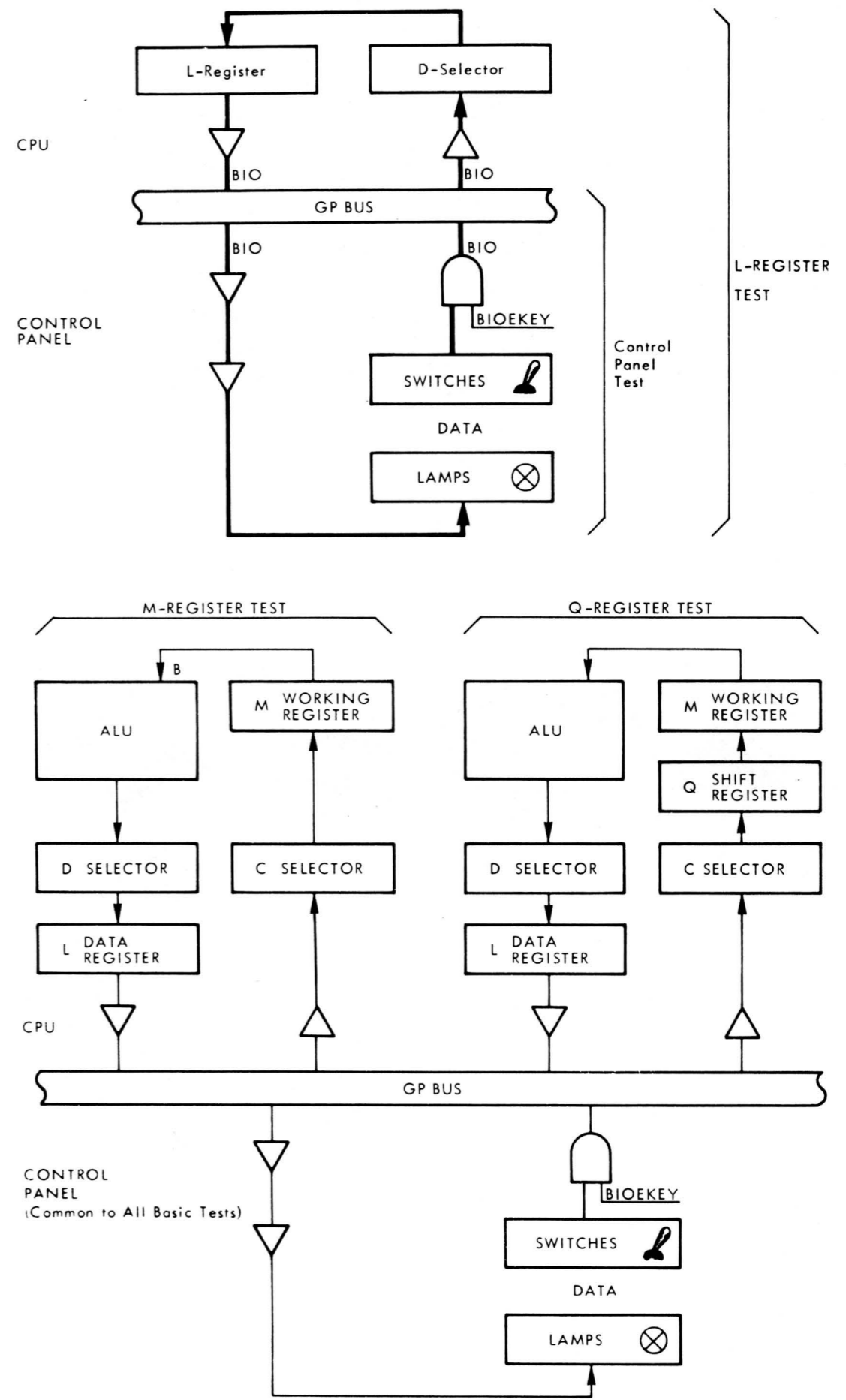
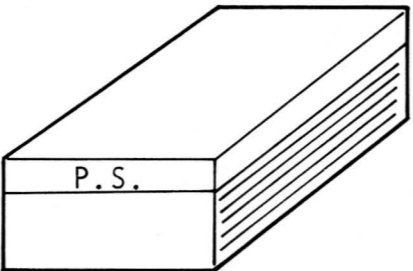
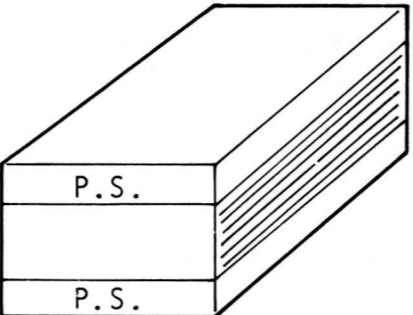


Figure 3-11 Logic Tested by Basic Tests

Chassis	Card Slots	Models	Maximum Memory Size (16-bit words)	Width - 19 inches (483mm) Depth - 21.65 in. (550mm)	Height
M4	10	P856	32K		6 U 267mm
		P857 MMU FPP optional	32K		
		P857 MMU standard FPP optional	64K		
OPTIONAL					
M5	17	P856	32K		11 U 489mm
		P857 MMU, FPP	128K		

- The Memory Management Unit (MMU) is required for all systems with more than 32k words of memory.
- The Floating Point Processor (FPP) is available with any P857M system.
- 1 U = 44.45 mm (1.75 inches)

Figure 4-1 P856M/P857M Chassis Configurations

SECTION IV

MECHANICAL

4.1 GENERAL

The P856M/P857M System is available in either of two basic chassis (M4 or M5), with two different extension chassis (E1, E2) available for additional control-unit cards. The different basic-chassis configurations are shown in Figure 4-1. Each chassis contains a power supply, ventilation, printed circuits, and I/O cable connectors, as well as the logic-card slots. Each chassis is slide mounted in a 19-inch (483mm) rack. Each basic chassis includes a control panel (either complete or simplified) mounted on the front face. The chassis dimensions and installation data are provided in Figure 4-2.

4.2 WIRING AND CABLING

4.3 General

The cable connections on the P857M basic chassis and the extension chassis, are shown on Figure 4-3. The uses of the different connectors on the cards are shown on Figure 4-4. Lists of signals and pin numbers for the card connectors are provided in the following tables :

Table	Connector
4-1	GP Bus connector, IOM-IOB
4-2	CPU-A connector-1 (V24 CU)
4-3	Connector-3 (CPU, Memory, IOP, CU)
4-4	CPU-A connector-5
4-5	Control Panel Connector
4-6	IOP Connectors 4, 5 (Break)
4-7	Extension Connectors AIE/TAIE

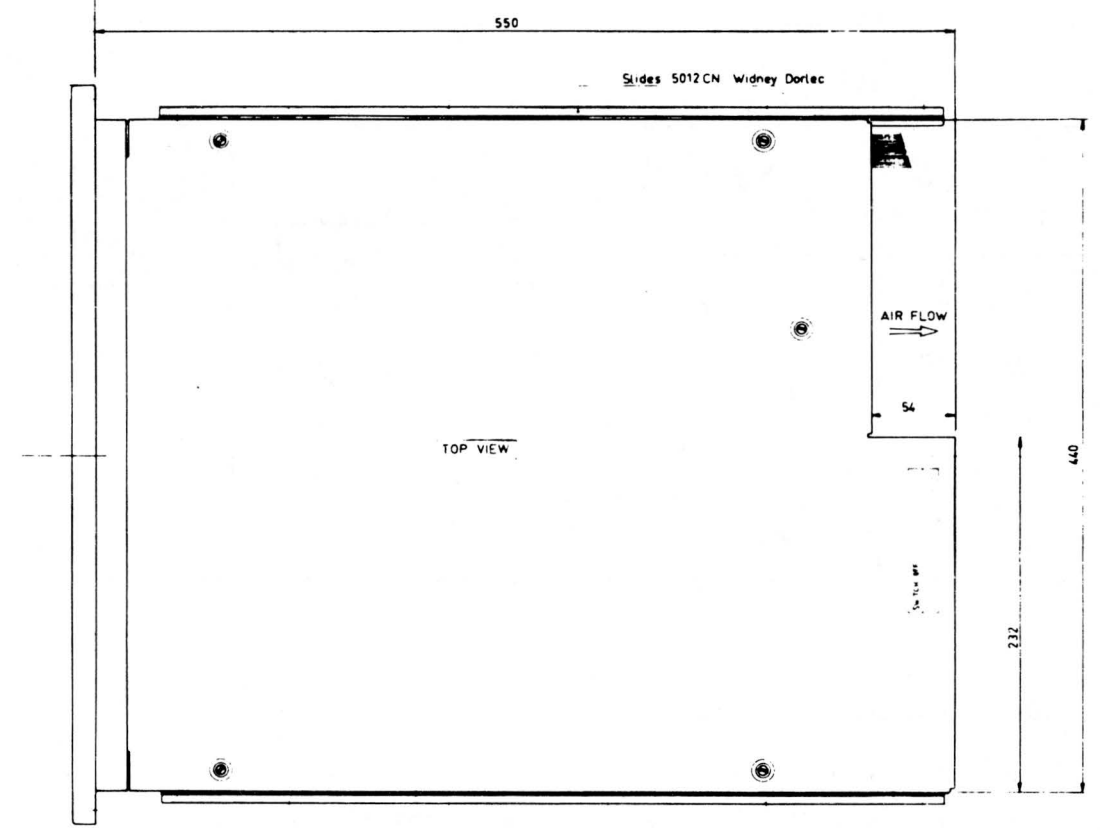
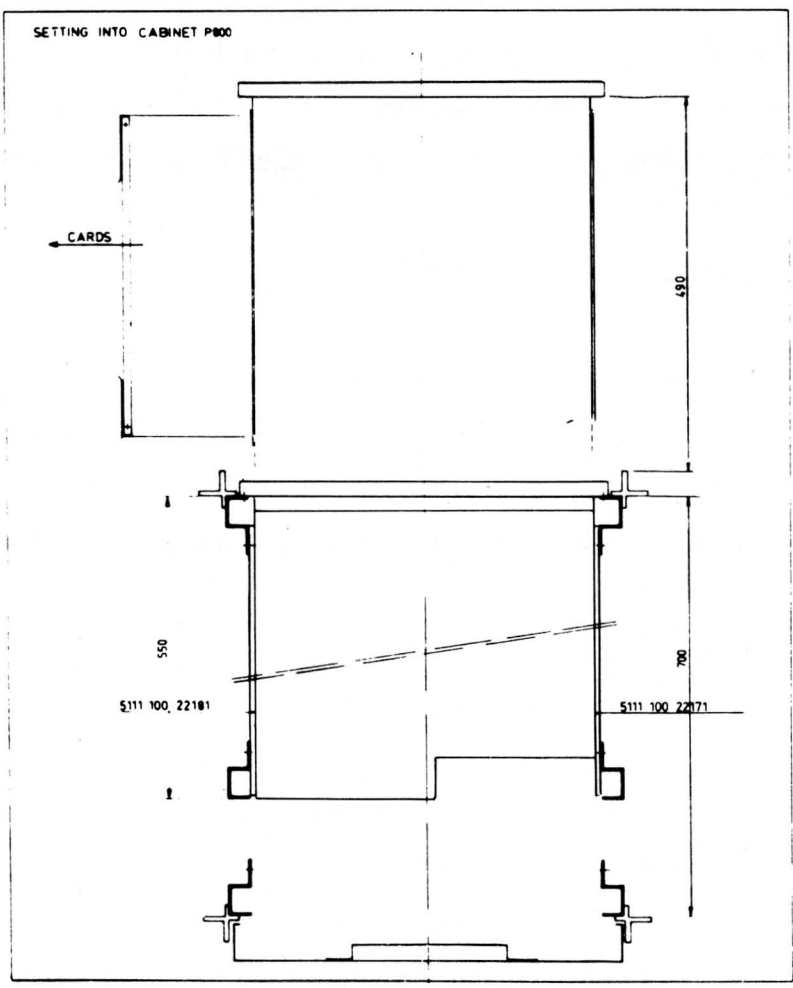
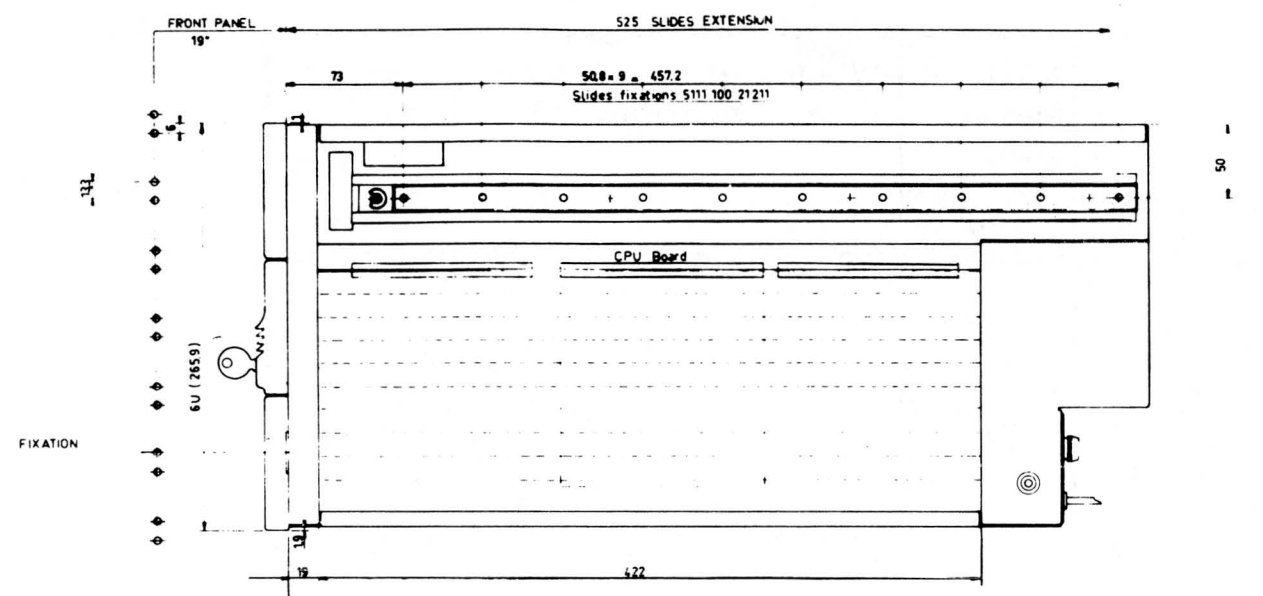
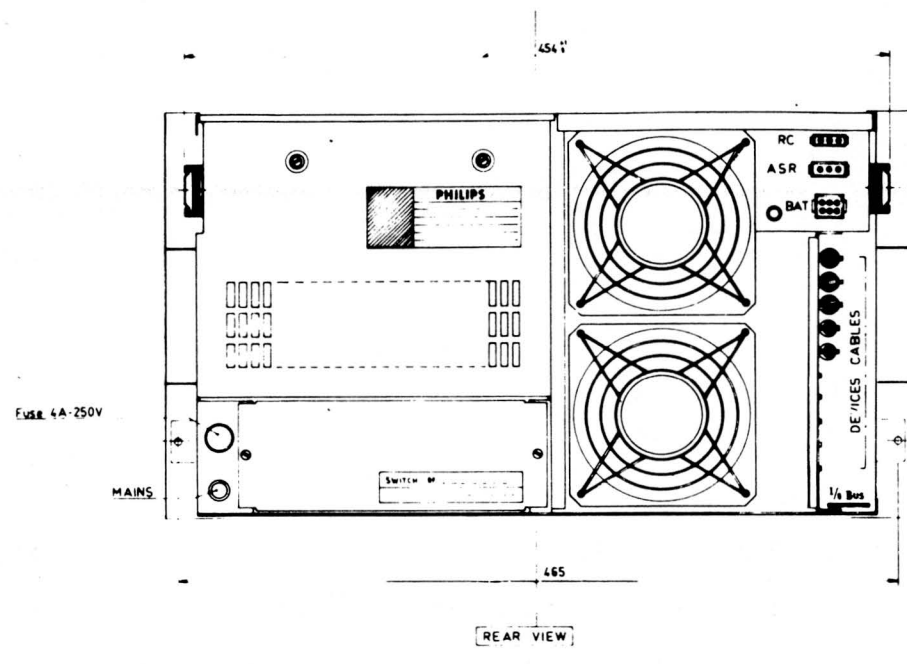


Figure 4-2A M4 Chassis Installation Data

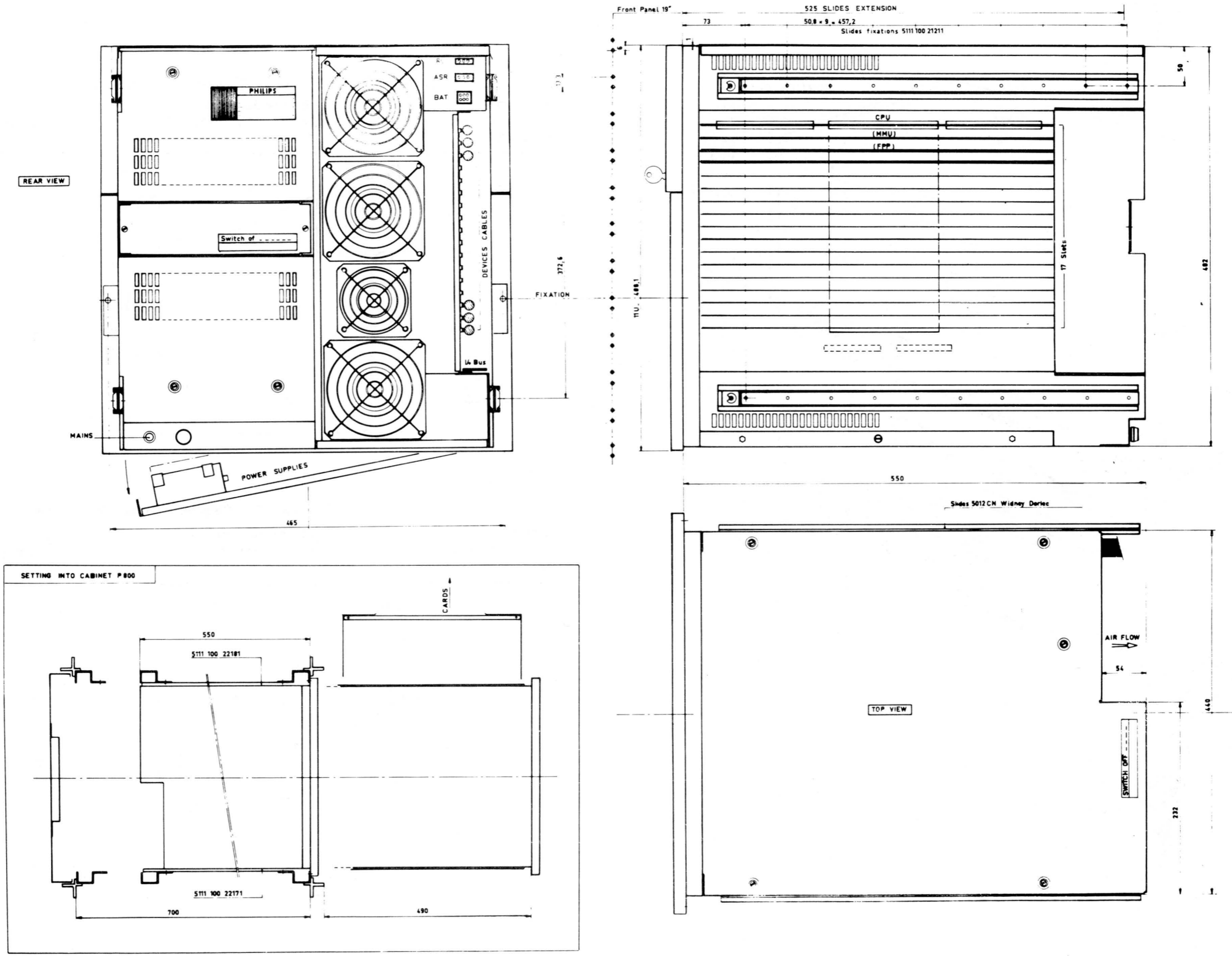


Figure 4-2B M5 Chassis Installation Data

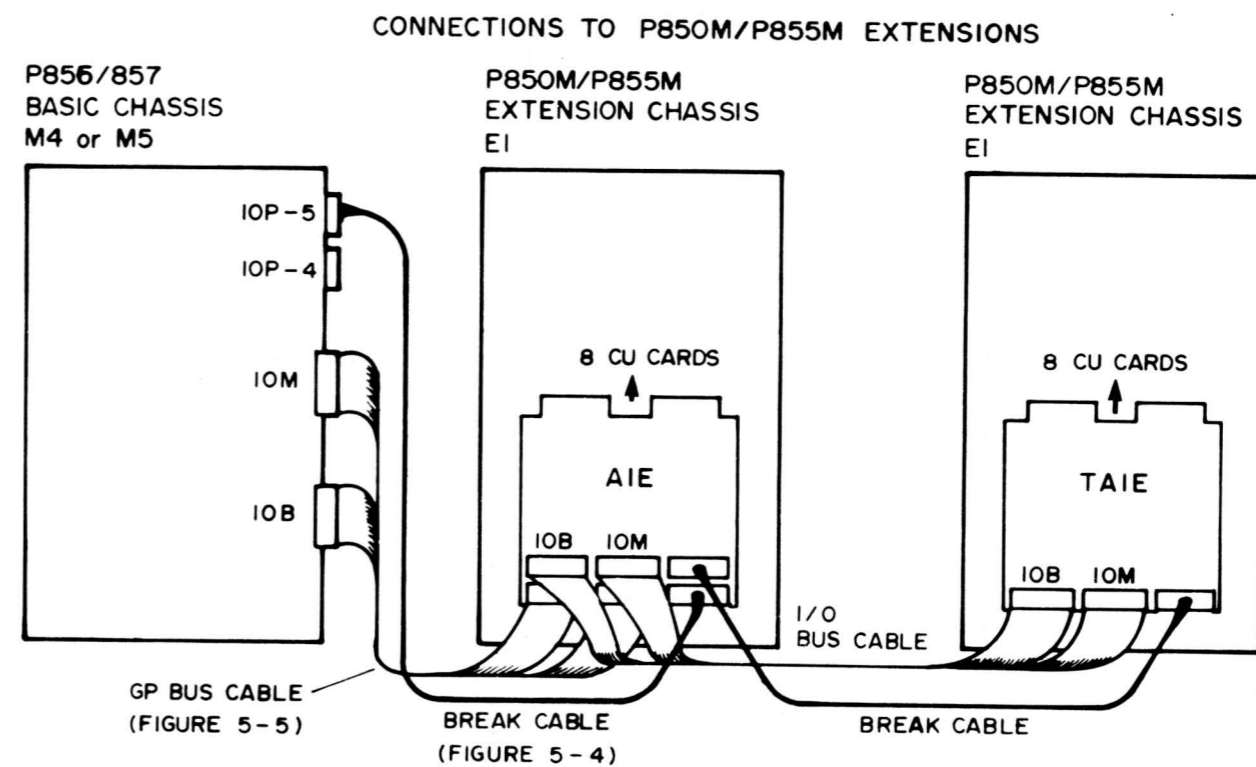
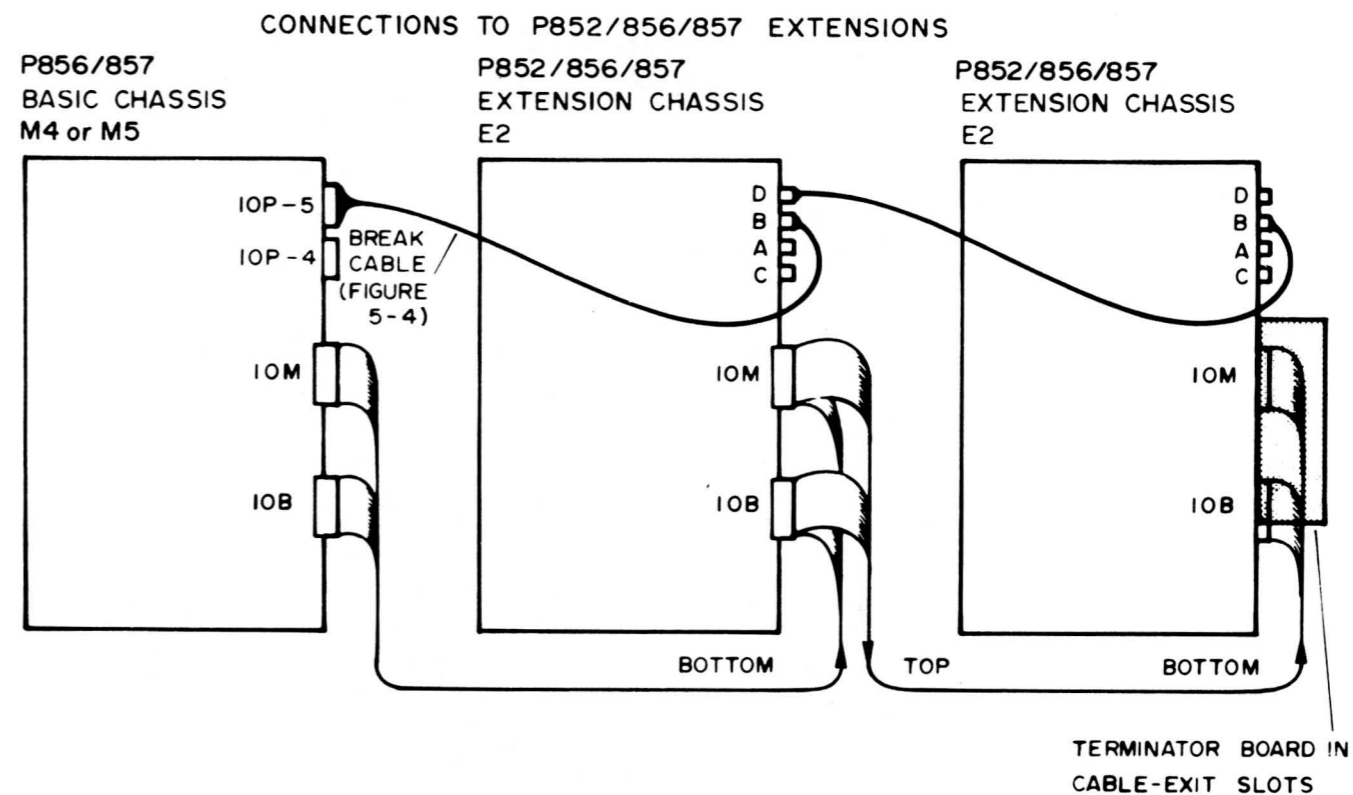
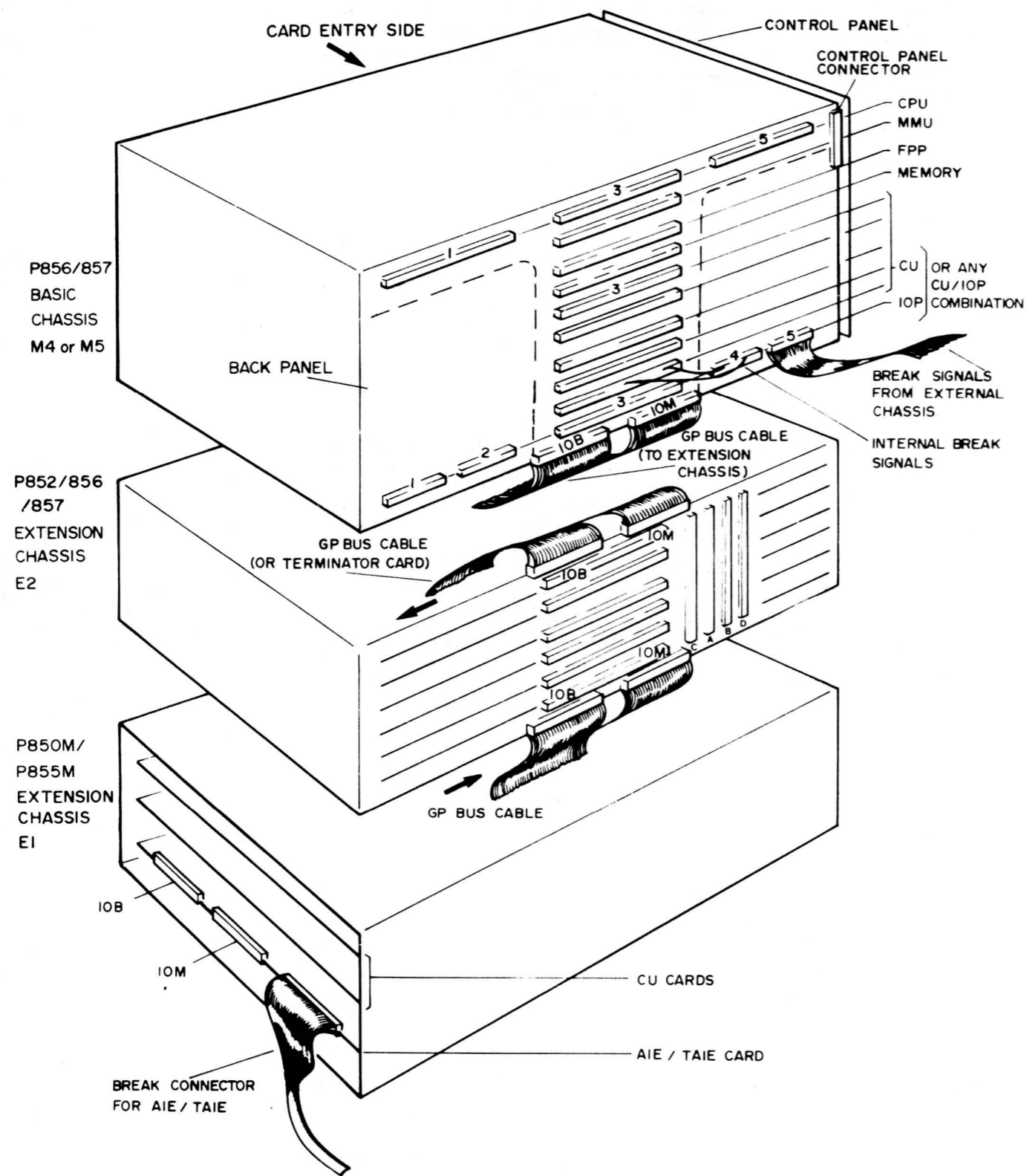


Figure 4-3 P856M/857M Basic/Extension Chassis Connections

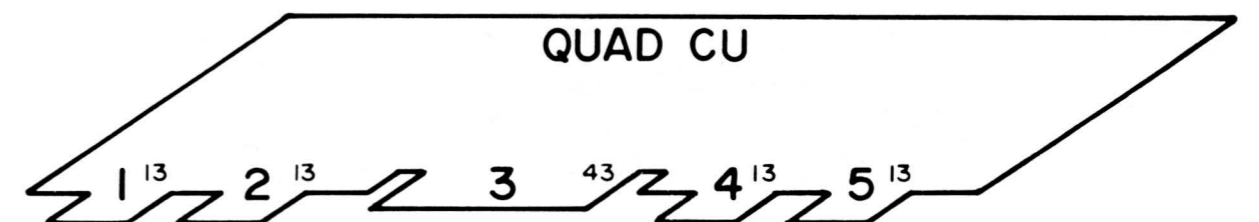
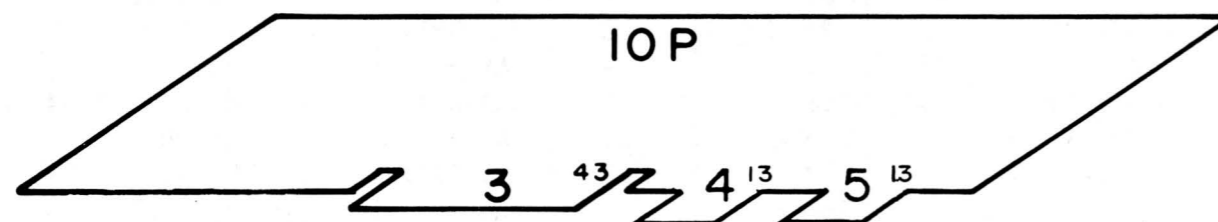
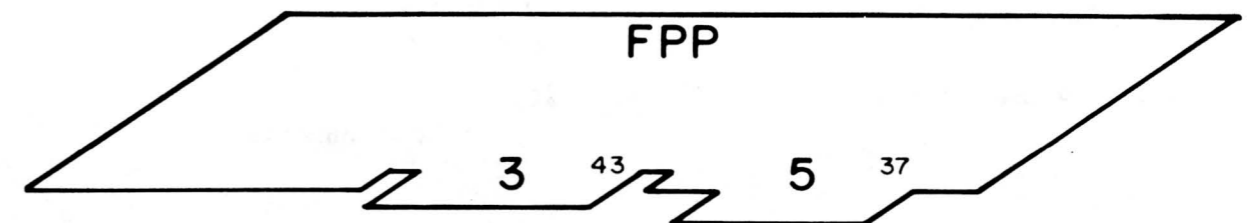
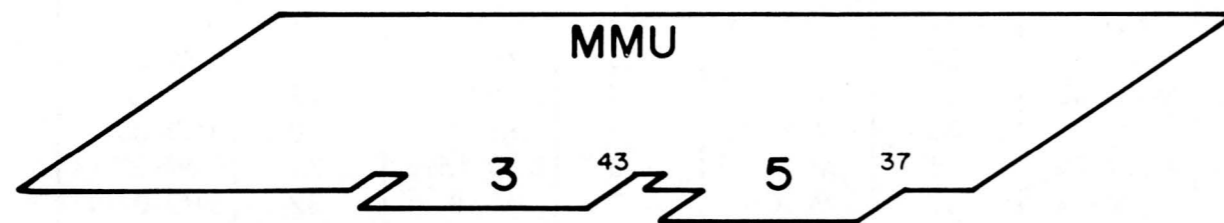
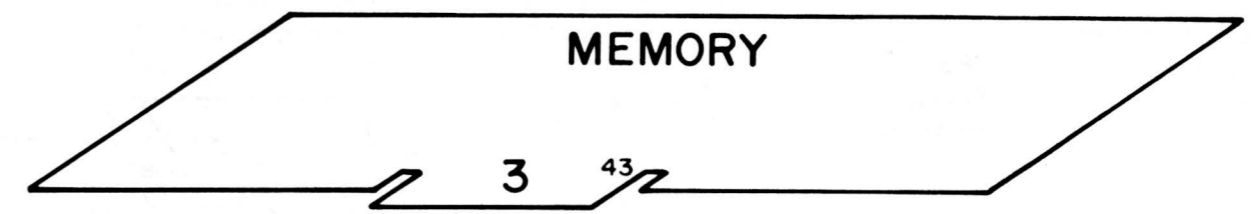
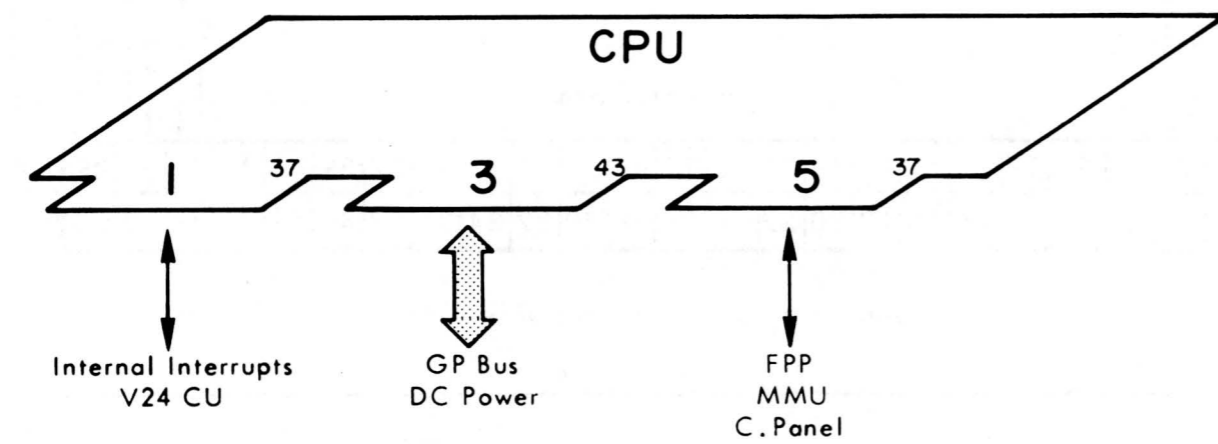
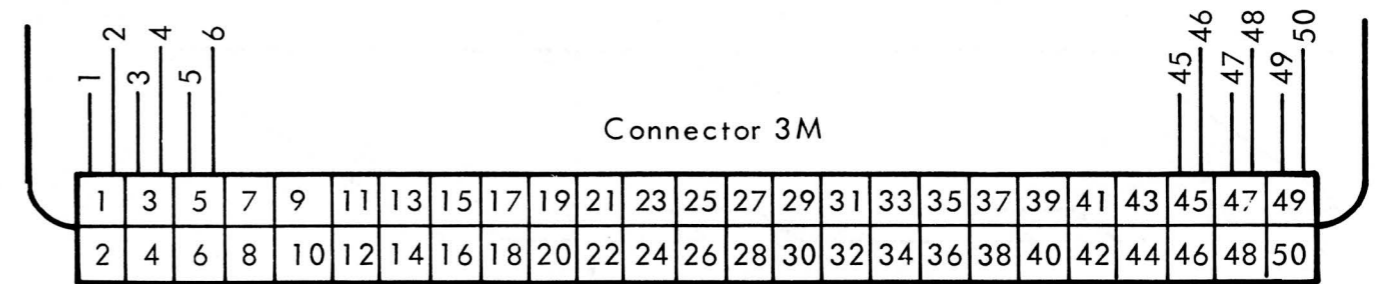
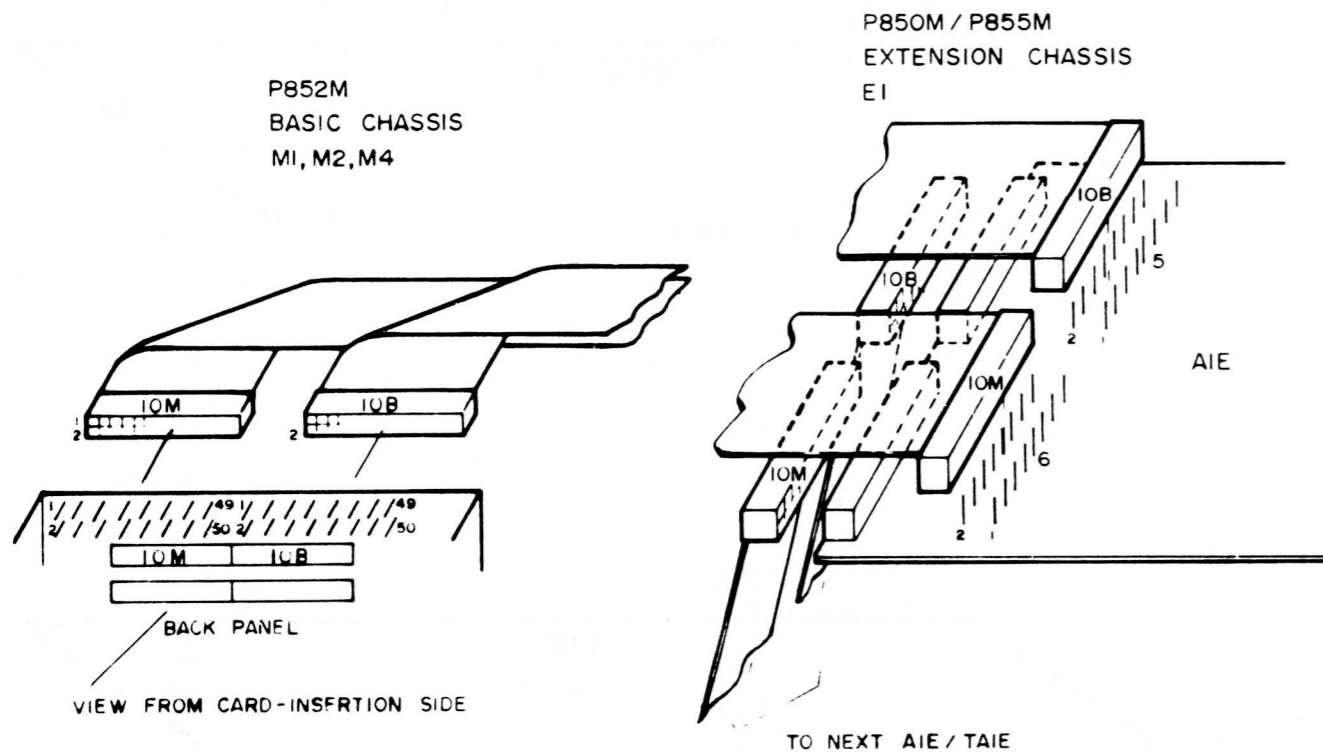


Figure 4-4 Circuit-Card Connector Uses

Table 4-1 GP Bus Connector IOM,IOB



Connector viewed from cable side

Connector I O M				Connector I O B			
N° Pin	Signal	N° Pin	Signal	N° Pin	Signal	N° Pin	Signal
1	M A	26	M C	1	M C	26	BIO 05N
2	MAD 04	27	CLEARN	2	RSLN	27	M 3
3	M A	28	M C	3	M C	28	BIO 04N
4	MAD 03	29	M C	4	PWFN	29	M B
5	M A	30	TPMN	5	M B	30	BIO 03N
6	MAD 08	31	M C	6	BIO 15N	31	M B
7	M A	32	M C	7	M B	32	BIO 02N
8	MAD 09	33	TMPN	8	BIO 14N	33	M B
9	M A	34	M C	9	M B	34	BIO 01N
10	MAD 10	35	M C	10	BIO 13N	35	M B
11	M A	36	TMEN	11	M B	36	BIO 00N
12	MAD 11	37	M C	12	BIO 12N	37	M B
13	M A	38	M C	13	M B	38	BIEC5
14	MAD 12	39	TRMN	14	BIO 11N	39	M C
15	M A	40	M C	15	M B	40	SCEIN
16	MAD 13	41	M C	16	BIO 10N	41	M C
17	M A	42	Spare	17	M B	42	BIEC3
18	MAD 14	43	M C	18	BIO 09N	43	M C
19	M A	44	Spare	19	M B	44	BIEC4
20	MAD 15	45	M C	20	BIO 08N	45	M C
21	M A	46	Spare	21	M B	46	BIEC1
22	ACN	47	M C	22	BIO 07N	47	M C
23	M C	48	Spare	23	M B	48	BIEC2
24	ARN	49	M D	24	BIO 06N	49	M C
25	M C	50	5V	25	M B	50	BIEC0

4.4 Operator I/O Device

The operator's input/output device is connected to the CPU-integral Serial Control Unit via CPU connector 1. The maximum cable length between the device and the CPU is 20 meters. The operator's interrupt does not use the encoded BIEC lines, but is connected from the Serial Control Unit directly to the internal-interrupt inputs (see Figure 1-3A).

4.5 Interface Signals

All interface signals between the CPU and the units (Serial CU, MMU, FPP, and interrupts) are listed in Table 4-8. The GP Bus signals are described in Section II. All these signals are also included in the signal lists for the connectors where they are used.

4.6 Interrupts and Breaks

The use of interrupts and breaks, and their interconnections, are described in Section I and shown in Figure 1-2.

4.7 CARDS

The complete CPU and the V24 Serial Control Unit are mounted on a single, multi-layer printed circuit card. The circuit-card locations within the chassis are shown in Figure 4-3. The CPU card fits in a dedicated slot at the top of the chassis. The MMU and FPP (P857 only), and memory cards fit in the next three dedicated slots. The MMU and FPP must be close to the CPU card. Both the MMU and the FPP have some discrete wiring connections to the CPU which are used to increase operating speed. If either the MMU or FPP are not used, a memory card may be placed in that dedicated slot. Card connector uses are shown in Figure 4-4. Circuit locations and parts lists are provided at the end of this section.

4.8 INTEGRATED CIRCUITS

A complete list of integrated circuits (ICs) and a guide which shows the IC symbols, input and output polarities, control codes, and pin numbers is provided in Appendix A.

4.9 Read Only Memories (ROMS) and PLA

The ROMs and the Programmable Logic Array (PLA) circuits are pre-loaded with special contents, or codes, which cannot be shown on the general IC-diagram drawing. The following list is a directory to the logic descriptions that use these circuits :

Circuit	Logic		Paragraph
6200 1024-bit ROM	LL	IPL	2.66
7488A 256-bit ROM	FF	A,D,L Command	2.47
	NN	CR code selection	2.76
8205 4096-bit ROM	BB	Microcommand Control Store	2.37
8576 96-code PLA	AA	Instruction Decode	2.42

4.10 PARTS LISTS

The following figures and tables give the parts layout and parts lists for the circuit cards and the control panel.

Table 4-2 CPU-A, Connector 1 (V24 CU)

1A01	0V	1B01	ASR LINE
1A02	FIFN	1B02	RTC AN
1A03	CPFN	1B03	SCEIN
1A04	IS02N	1B04	IS05N
1A05	FFFN	1B05	IS04N
1A06	BIEC4	1B06	RTCFZ1N
1A07	BIEC2	1B07	BIEC1
1A08	BIEC5	1B08	BIEC3
1A09	IS06N	1B09	BIEC0
1A10	IS03N	1B10	IS07N
1A11	IS01N	1B11	IS00N
1A12	INTASRN (INTSERN)	1B12	
1A13		1B13	
1A14		1B14	
1A15		1B15	
1A16		1B16	
1A17		1B17	
1A18		1B18	
1A19		1B19	
1A20		1B20	
1A21		1B21	
1A22		1B22	
1A23		1B23	
1A24		1B24	
1A25		1B25	
1A26		1B26	
1A27	0V	1B27	0V
1A28	5V	1B28	5V
1A29		1B29	
1A30	Mech. Ground	1B30	
1A31	CT103	1B31	0V
1A32	CT104	1B32	0V
1A33	CT106	1B33	0V
1A34	CT107	1B34	0V
1A35	CT1082	1B35	0V
1A36	CT109	1B36	0V
1A37	CT133	1B37	0V

Table 4-3 Connector 3 (CPU, Mem, IOP, CU)

3A01	+18V		c	3B01	-18V		c
3A02	BIEC0	a	c	3B02	Chassus Ground		c
3A03	BIEC2	a	c	3B03	BIEC1	a	c
3A04	BIEC4	a	c	3B04	BIEC3	a	c
3A05	SCEIN	a	c	3B05	BIEC5	a	c
3A06	+16V		b	3B06	+16V		b
3A07	0V			3B07	0V		
3A08	BIO 00N			3B08	BIO 01N		
3A09	BIO 02N			3B09	BIO 03N		
3A10	BIO 04N			3B10	BIO 05N		
3A11	BIO 06N			3B11	BIO 07N		
3A12	BIO 08N			3B12	BIO 09N		
3A13	BIO 10N			3B13	BIO 11N		
3A14	BIO 12N			3B14	BIO 13N		
3A15	BIO 13N			3B15	BIO 15N		
3A16	OKO	*	a c d	3B16	OKI	*	c d
3A17	PWFN		a c	3B17	-5V		a b c
3A18	0V			3B18	-5V		b
3A19	+5V			3B19	+5V		
3A20	+5V			3B20	+5V		
3A21	0V			3B21	0V		
3A22	0V			3B22	0V		
3A23	BR (CU - 4)		c	3B23	+5V Battery		
3A24	0V			3B24			
3A25	0V			3B25	+16VM		b
3A26	WRITE	*		3B26	MAD 15		
3A27	CHA	*		3B27	MAD 14		
3A28	TRMN	*		3B28	MAD 13		
3A29	TMRN		a b c	3B29	MAD 12		
3A30	TMEN		a c d	3B30	MAD 11		
3A31	TMPN		a c d	3B31	MAD 10		
3A32	TPMN		a c d	3B32	MAD 09		
3A33	0V			3B33	MAD 08		
3A34	ACN		a c	3B34	MAD 07		*
3A35	SPYC	*	a c d	3B35	MAD 06		*
3A36	BUSRN	*	a c d	3B36	MAD 05		*
3A37	MSN	*	a c d	3B37	MAD 04		
3A38	BSYN	*	a c d	3B38	MAD 03		
3A39	CLEARN		a c d	3B39	MAD 02		*
3A40	0V			3B40	MAD 01		*
3A41	BR (CU-2)		c	3B41	MAD 00		*
3A42	BR (CU-3)		c	3B42	MAD 64		*
3A43	BR (CU-1)		c	3B43	MAD 128		

a- CPU only
b- Memory only

c- Control Unit only
d- IOP only

* CU use only on DMA
(main chassis)

Table 4-4 CPU-A Connector-5

5A01		5B01	* SP05
5A02		5B02	* GBCPFN
5A03		5B03	* PREQN
5A04		5B04	* CPBABS
5A05		5B05	* TESTN
5A06		5B06	
5A07		5B07	
5A08		5B08	
5A09		5B09	
5A10	* SP03	5B10	* SP04
5A11	* FLOACTN	5B11	* SP01
5A12	* BSYCPUAN	5B12	* TMFN
5A13	* GFETCH	5B13	* BOFFN
5A14	* DONEFN	5B14	* PLOCRO
5A15	* FLOCRI	5B15	* FPPABS
5A16		5B16	
5A17	* OSCF0	5B17	* MOSCFLO
5A18		5B18	
5A19	* MMUABS	5B19	
5A20	* DONEMN	5B20	* MFAULTN
5A21	* BOMFN	5B21	
5A22	* FU	5B22	* S00
5A23	* S01	5B23	* S02
5A24	* S03	5B24	* TMMM
5A25	* SP02	5B25	* TMMU
5A26		5B26	
5A27	* 0V	5B27	* 0V
5A28	* 5V	5B28	* 5V
5A29	BIOEKEY	5B29	CPMCN
5A30	UNLOCKN	5B30	IPL
5A31	RUNN	5B31	START
5A32	RCP00N	5B32	CPINT
5A33	LOADRN	5B33	RUNFA
5A34	READSTN	5B34	RCP01N
5A35	RCP03N	5B35	RCP02N
5A36	LOADMN	5B36	READMN
5A37	READRN	5B37	INSTN

Table 4-5 Control Panel Connector

A01	BIO15N	B01	CPMN
A02	BIO14N	B02	BIOEKEY
A03	BIO13N	B03	IPL
A04	BIO12N	B04	UMLOCKN
A05	BIO11N	B05	START
A06	BIO10N	B06	RUNN
A07	BIO09N	B07	CPINT
A08	BIO08N	B08	RCP0N
A09	BIO07N	B09	RUNFA
A10	BIO06N	B10	LOADRN
A11	BIO05N	B11	RCP1N
A12	BIO04N	B12	READSTN
A13	BIO03N	B13	RCP2N
A14	BIO02N	B14	RCP3N
A15	BIO01N	B15	READMN
A16	BIO00N	B16	INSTN
A17		B17	LOADMN
A18	READRN	B18	LOCK
A19	0V	B19	+5V

Table 4-6 IOP Connectors 4, 5 (Break)

4A01	IR07N
4A02	IR05N
4A03	IR03N
4A04	IR01N
4A05	
4A06	BREX07N
4A07	BREX06N
4A08	BREX05N
4A09	BREX04N
4A10	BREX03N
4A11	BREX02N
4A12	BREX01N
4A13	BREX00N

4B01	IR06N
4B02	IR04N
4B03	IR02N
4B04	IR00N
4B05	
4B06	BR07N
4B07	BR06N
4B08	BR05N
4B09	BR04N
4B10	BR03N
4B11	BR02N
4B12	BR01N
4B13	BR00N

5A01	
5A02	
5A03	
5A04	
5A05	
5A06	BREX07N
5A07	BREX06N
5A08	BREX05N
5A09	BREX04N
5A10	BREX03N
5A11	BREX02N
5A12	BREX01N
5A13	BREX00N

5B01	
5B02	
5B03	
5B04	
5B05	
5B06	Ground (0V)
5B07	Ground (0V)
5B08	Ground (0V)
5B09	Ground (0V)
5B10	Ground (0V)
5B11	Ground (0V)
5B12	Ground (0V)
5B13	Ground (0V)

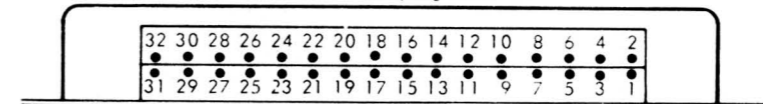
Table 4-7 Extension Connectors AIE/TAIE

TAIE Connectors

1A01	- 5V	1B01	0V	2A01	BOF02N	2B01	IR04N
1A02		1B02	BIN00N	2A02	EOR	2B02	IR02N
1A03	BIN00N	1B03	BIN01N	2A03	BOF01N	2B03	IR06N
1A04	BIN01N	1B04	BIN02N	2A04	BOF00N	2B04	IR07N
1A05	BIN02N	1B05	BIN03N	2A05	BOU14	2B05	IR00N
1A06	BIN03N	1B06	BIN04N	2A06	BOU13	2B06	IR01N
1A07	BIN04N	1B07	BIN05N	2A07	BOU12	2B07	IR03N
1A08	BIN05N	1B08	BIN06N	2A08	BOU09	2B08	IR05N
1A09	BIN06N	1B09	BIN07N	2A09	BOU10	2B09	
1A10	BIN07N	1B10		2A10	BOU11	2B10	
1A11	BR00	1B11	ACCN	2A11	BOU15	2B11	
1A12	BR01	1B12	ACCN	2A12	MCN	2B12	
1A13	ACCN	1B13	AREN	2A13	BAD03N	2B13	
1A14	AREN	1B14	AREN	2A14	DAVN	2B14	
1A15		1B15	SP4	2A15	BAD04N	2B15	
1A16		1B16	SP3	2A16	BAD05N	2B16	
1A17	BR02	1B17	SP2	2A17	BOU08	2B17	
1A18	BR03	1B18	SP1	2A18	BOU07	2B18	
1A19	BR06	1B19		2A19	BOU06	2B19	
1A20	BR07	1B20		2A20	BOU05	2B20	
1A21	BIN08N	1B21	BIN08N	2A21	BOU04	2B21	
1A22	BIN14N	1B22	BIN14N	2A22	BAD00N	2B22	
1A23	BR04	1B23		2A23	BOU03	2B23	
1A24	BR05	1B24	S	2A24	BAD01N	2B24	
1A25		1B25		2A25	BAD02N	2B25	
1A26	BIN13N	1B26	BIN13N	2A26	BOU02	2B26	
1A27	BIN15N	1B27	BIN15N	2A27	BOU01	2B27	
1A28	BIN09N	1B28	BIN09N	2A28	BOU00	2B28	
1A29	BIN10N	1B29	BIN10N	2A29	- 5V TELEC	2B29	PWF MOD1
1A30	BIN11N	1B30	BIN11N	2A30		2B30	PWF MOD2
1A31	BIN12N	1B31	BIN12N	2A31	- 5V	2B31	0V

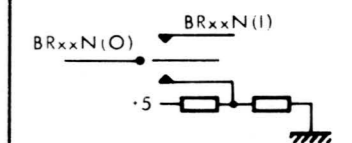
Break Connector for AIE TAIE Cards

(View from plug side)



1	BR00N (I)
2	BR00N (O)
3	M
4	M
5	BR01N (I)
6	BR01N (O)
7	M
8	M
9	BR02N (I)
10	BR02N (O)
11	M
12	M
13	BR03N (I)
14	BR03N (O)
15	M
16	M

17	BR04N (I)
18	BR04N (O)
19	M
20	M
21	BR05N (I)
22	BR05N (O)
23	M
24	M
25	BR06N (I)
26	BR06N (O)
27	M
28	M
29	BR07N (I)
30	BR07N (O)
31	M
32	M



Reference Berg 65268 - 005

Table 4-8 Interface Signal List

GP BUS (Section II)		
Input to CPU	Input/Output	Output from CPU
BUSRN MSN PWFN RSLN BIEC0-5	ACN BIO00-15N BSYN TMEN TMPN TMRN TPMN TRMN	CHA WRITE CLEARN MAD128,64,00-15 OK0/OK1 SCEIN SPYC
CONTROL PANEL (Section III)		
Input to CPU	Output from CPU	
* CPBABS CPINT CPMCN INSTN IPL LOADMN LOADRN * PREQN	RCP0-3N READMN READRN READSTN RUNN START TESTS UNLOCKN	BIOEKEY * BSYCPUAN * GBCPFN RUNFA * = Extended (Address) half of panel, P857 only
V24 SERIAL CONTROL UNIT (Section VI)		
Input from Device	Output from Device	Ground Signals
CT103 CT1082 CT133	CT104 CT106 CT107 CT109	CT101 CT102
CPU/V24 SERIAL CU (CPU Logic)		
Input to CPU	Output from CPU	
AREDELA (TT) ASR0-7 (JJ) BRGFN (PP) TYAC (TT)	BUSFDET (TT) K04,08,10-15 (AA) μ Q1 (BB) CPGFZ0N(PP) K04,08-15 (AA) RSLAN (RR) D10,15 (HH) L08-15 (HH) RSLCN (RR) FNU (DD) MCL, MCLN (RR) T5N,T7 (EE) TC810 (EE)	
V24 CU to CPU Interrupt		
INTSERN		

MEMORY MANAGEMENT UNIT (MMU)		
Input to CPU	Output from CPU	
DONEMN MFAULTN MMUABS	BOMFN BSYCPUAN FU GFETCH	OSCFLO S00-03 TMMN TMMU
FLOATING POINT PROCESSOR (FPP)		
Input to CPU	Output from CPU	
DONEFN FPPABS FLOCRO, 1	BOFFN BSYCPUAN FLOACT	GFETCH OSCFLO TMFN
INTERNAL INTERRUPTS (LEVELS 0-7)		
Input to CPU	Output from CPU	Input to CPU
(P. Supply) PWFN (C. Panel) CPINT (CPU program) (P. Supply) RTCFZ1N (V24 CU) INSERTN - - - - -	PFFN CPINTFN PIFN RTCFAN	ISO0N ISO1N ISO2N ISO3N ISO4N ISO5N ISO6N ISO7N

Table 4-9A P856M CPU Parts List

Reference	Description	12NC Code
	Printed Circuit	5111 100 05701
	Heat Sink	
	Right Spring	
	Left Spring	
	Integrated Circuit 7400	
	Integrated Circuit 7402	
	Integrated Circuit 7403	
	Integrated Circuit 7404	
	Integrated Circuit 7408	
	Integrated Circuit 7410	
	Integrated Circuit 7420	
	Integrated Circuit 7425	
	Integrated Circuit 7427	
	Integrated Circuit 7428	
	Integrated Circuit 7430	
	Integrated Circuit 7432	
	Integrated Circuit 7437	
	Integrated Circuit 7450	
	Integrated Circuit 7453	
	Integrated Circuit 7474	
	Integrated Circuit 7485	
	Integrated Circuit 7486	
	Integrated Circuit 7489	
	Integrated Circuit 74121	
	Integrated Circuit 74132	
	Integrated Circuit 74148	
	Integrated Circuit 74151A	
	Integrated Circuit 74157	
	Integrated Circuit 74158	
	Integrated Circuit 74161	
	Integrated Circuit 74174	
	Integrated Circuit 74175	
	Integrated Circuit 74181	
	Integrated Circuit 74182	
	Integrated Circuit 74194	
	Integrated Circuit 74298	
	Integrated Circuit 74500	
	Integrated Circuit 74510	
	Integrated Circuit 74511	
	Integrated Circuit 74520	
	Integrated Circuit 74530	
	Integrated Circuit 74504	
	Integrated Circuit 74564	

Table 4-9A Contd.

Reference	Description	12NC Code
	Integrated Circuit 74574	
	Integrated Circuit 745112	
	Integrated Circuit 745138	
	Integrated Circuit 745169	
	Integrated Circuit 745174	
	Integrated Circuit 745175	
T3.	Integrated Circuit 7812 (TO22O)	
T2.	Integrated Circuit 7912 (TO22O)	
	Integrated Circuit 1488	
	Integrated Circuit 1489A	
	Integrated Circuit 1801	
	Integrated Circuit 1891 (8576)	
	Integrated Circuit 8234	
	Integrated Circuit 9301	
	Integrated Circuit 9309	
	Integrated Circuit ADL 2011	
	Integrated Circuit REC 0612	
	Integrated Circuit 2502 (UART)	
	Integrated Circuit CR 2021	
	Integrated Circuit ROM 2032 (8205)	
	Integrated Circuit ROM 2042 (8205)	
	Integrated Circuit ROM 2052 (8205)	
	Integrated Circuit ROM 2062 (8205)	
	Integrated Circuit ROM 2072 (8205)	
	Integrated Circuit ROM 2083 (8205)	
	Integrated Circuit SN 74LS169N	
C10.	Capacitor 1nF, 100V, 10%, ceramic.	
C68.	Capacitor 200pF, 500V, 1%.	
C8,76.	Capacitor 390pF, 250V, 1%.	
C157.	Capacitor 510pF, 250V, 1%.	
C1,5,14.	Capacitor 620pF, 250V, 1%.	
C9.	Capacitor 1000pF, 125V, 1%.	
C7,6.	Capacitor 1.3nF, 63V, 1%.	
C2,4.	Capacitor 430pF, 250V, 1%.	
C13.	Capacitor 2 nF, 63V, 1%	
C158.	Capacitor 150pF, 63V, 2%.	
C21,24.	Capacitor 10µF, 63V, FITCO.	
C15,18,20,22,23,25,27,29,31-35,38,41,42.	Capacitor 10µF, 25V, FITCO.	
C3.	Capacitor 1,8nF, 125V, 1%.	

Table 4-9A Contd.

Reference	Description	12NC Code
C12, 46, 49-64, 67, 69, 71, 74, 79, 80, 85, 86, 88, 90, 92, 94, 97, 100, 102, 104-106, 109, 110, 112, 113, 115, 117, 120-122, 124, 128-130, 133, 134, 138, 142, 146, 147, 149, 151, 152, 155.	Capacitor 10nF, ceramic.	
R3, 8, 37, 54, 74, 75.	Resistor 100n, 1 8W, 1%.	
R27, 36, 65-72.	Resistor 470n, 1 4W, 5%.	
R53.	Resistor 560n, 1 4W, 5%.	
R14, 19-26, 28-35, 39-52, 55-62, 64, 76, 80-109, 130.	Resistor 1Kn, 1 4W, 5%.	
R04-06, 07, 38, 63, 77, 79, 131, 132.	Resistor 10Kn, 1 4W, 5%.	
R16.	Resistor 464n, 1 8W, 1%.	
R17.	Resistor 681n, 1 8W, 1%.	
R12.	Resistor 2.15Kn, 1 8W, 1%.	
R11, 13.	Resistor 3.16Kn, 1 8W, 1%.	
R110-115, 122, 123, 126, 127.	Resistor 220n, 1 4W, 5%.	
R116-121, 124, 125, 128, 129.	Resistor 390n, 1 4W, 5%.	
R15.	Resistor 1.47Kn, 1 8W, 1%.	
R1, 2, 10, 73.	Resistor 110n, 0.125W, 1%.	
R9, 78.	Resistor 147n, 0.125W, 1%.	
CR1, 2.	IC. TERNET Resistors.	
Y 1.	Quartz QA 55A22.22 Mhz.	
T 1.	Transistor BSX20	
	U-Link DCW06.	
	Mica 56325.	

Table 4-9B P857M CPU Parts List

Reference	Description	12 NC Code
	Printed Circuit	5111 100 05701
	Heat Sink	
	Right Spring	
	Left Spring	
	Integrated Circuit 7400	
	Integrated Circuit 7402	
	Integrated Circuit 7403	
	Integrated Circuit 7404	
	Integrated Circuit 7408	
	Integrated Circuit 7410	
	Integrated Circuit 7420	
	Integrated Circuit 7425	
	Integrated Circuit 7427	
	Integrated Circuit 7428	
	Integrated Circuit 7430	
	Integrated Circuit 7432	
	Integrated Circuit 7437	
	Integrated Circuit 7450	
	Integrated Circuit 7453	
	Integrated Circuit 7474	
	Integrated Circuit 7485	
	Integrated Circuit 7486	
	Integrated Circuit 7489	
	Integrated Circuit 74121	
	Integrated Circuit 74132	
	Integrated Circuit 74148	
	Integrated Circuit 74151A	
	Integrated Circuit 74157	
	Integrated Circuit 74158	
	Integrated Circuit 74161	
	Integrated Circuit 74174	
	Integrated Circuit 74175	
	Integrated Circuit 74181	
	Integrated Circuit 74182	
	Integrated Circuit 74194	
	Integrated Circuit 74298	
	Integrated Circuit 74500	
	Integrated Circuit 74510	
	Integrated Circuit 74511	
	Integrated Circuit 74520	
	Integrated Circuit 74530	
	Integrated Circuit 74504	
	Integrated Circuit 74564	
	Integrated Circuit 74574	

Table 4-9B Contd.

Reference	Description	12NC Code
	Integrated Circuit 74S112	
	Integrated Circuit 74S138	
	Integrated Circuit 74S169	
	Integrated Circuit 74S174	
	Integrated Circuit 74S175	
T3.	Integrated Circuit 7812 (TO22O)	
T2.	Integrated Circuit 7912 (TO22O)	
	Integrated Circuit 1488	
	Integrated Circuit 1489A	
	Integrated Circuit 1801	
	Integrated Circuit 1891 (8576)	
	Integrated Circuit 8234	
	Integrated Circuit 9301	
	Integrated Circuit 9309	
	Integrated Circuit 2502 (UART)	
	Integrated Circuit ADL 2011	
	Integrated Circuit REC 0612	
	Integrated Circuit CR 2021	
	Integrated Circuit ROM 2442 (8205)	
	Integrated Circuit ROM 2452 (8205)	
	Integrated Circuit ROM 2462 (8205)	
	Integrated Circuit ROM 2472 (8205)	
	Integrated Circuit ROM 2482 (8205)	
	Integrated Circuit ROM 2493 (8205)	
	Integrated Circuit SN74LS169N	
C10.	Capacitor 1nF, 100V, 10%, ceramic.	
C68.	Capacitor 200pF, 500V, 1%.	
C8,76.	Capacitor 390pF, 250V, 1%.	
C157.	Capacitor 510pF, 250V, 1%.	
C1,5,14.	Capacitor 620pF, 250V, 1%.	
C9.	Capacitor 1000pF, 125V, 1%.	
C6,7.	Capacitor 1.3nF, 63V, 1%.	
C2,4.	Capacitor 430pF, 250V, 1%.	
C13.	Capacitor 2nF, 63V, 1%.	
C12,46,49-64,67,69,71,74,79,80,85, 86,88,90,92,94,97,100,102,104-106, 109,110,112,113,115,117,120-122,124, 128-130,133,134,138,142,146,147,149, 151,152,155.	Capacitor 10nF, ceramic.	
C21,24.	Capacitor 10μF, 63V, FITCO.	
C15,18,20,22,23,25,27,29,31-35,38, 41,42.	Capacitor 10μF, 25V, FITCO.	
C158	Capacitor 150pF, 63V, 2% ceramic.	
C3.	Capacitor 1.8nF, 125V, 1%.	

Table 4-9B Contd.

Reference	Description	12NC Code
R3,8,37,54,74,75.	Resistor 100n, 1/8W, 1%.	
R27,36,65-72.	Resistor 470n, 1/4W, 5%.	
R53.	Resistor 560n, 1/4W, 5%.	
R14,19-26,28-35,39-52,55-62,64,76, 80-109,130.	Resistor 1Kn, 1/4W, 5%.	
R4,5,6,7,38,63,77,79,131,132.	Resistor 10Kn, 1/4W, 5%.	
R16.	Resistor 464n, 1/8W, 1%.	
R17.	Resistor 681n, 1/8W, 1%.	
R12.	Resistor 2.15Kn, 1/8W, 1%.	
R11,13.	Resistor 3.16Kn, 1/8W, 1%.	
R110-115,122,123,126,127.	Resistor 220n, 1/4W, 5%.	
R116-121,124,125,128,129.	Resistor 390n, 1/4W, 5%.	
R15.	Resistor 1.47Kn, 1/8W, 1%.	
R1,2,10,73.	Resistor 110n, 0,125W, 1%.	
R9,78.	Resistor 147n, 0,125W, 1%.	
CR1,2.	Diode AAZ 18.	
Y1.	Quartz QA 55A 22.22 Mhz.	
T1.	Transistor BSX 20.	
	IC TERNET Resistors.	
	U-Link DCW06.	
	Mica 56325.	

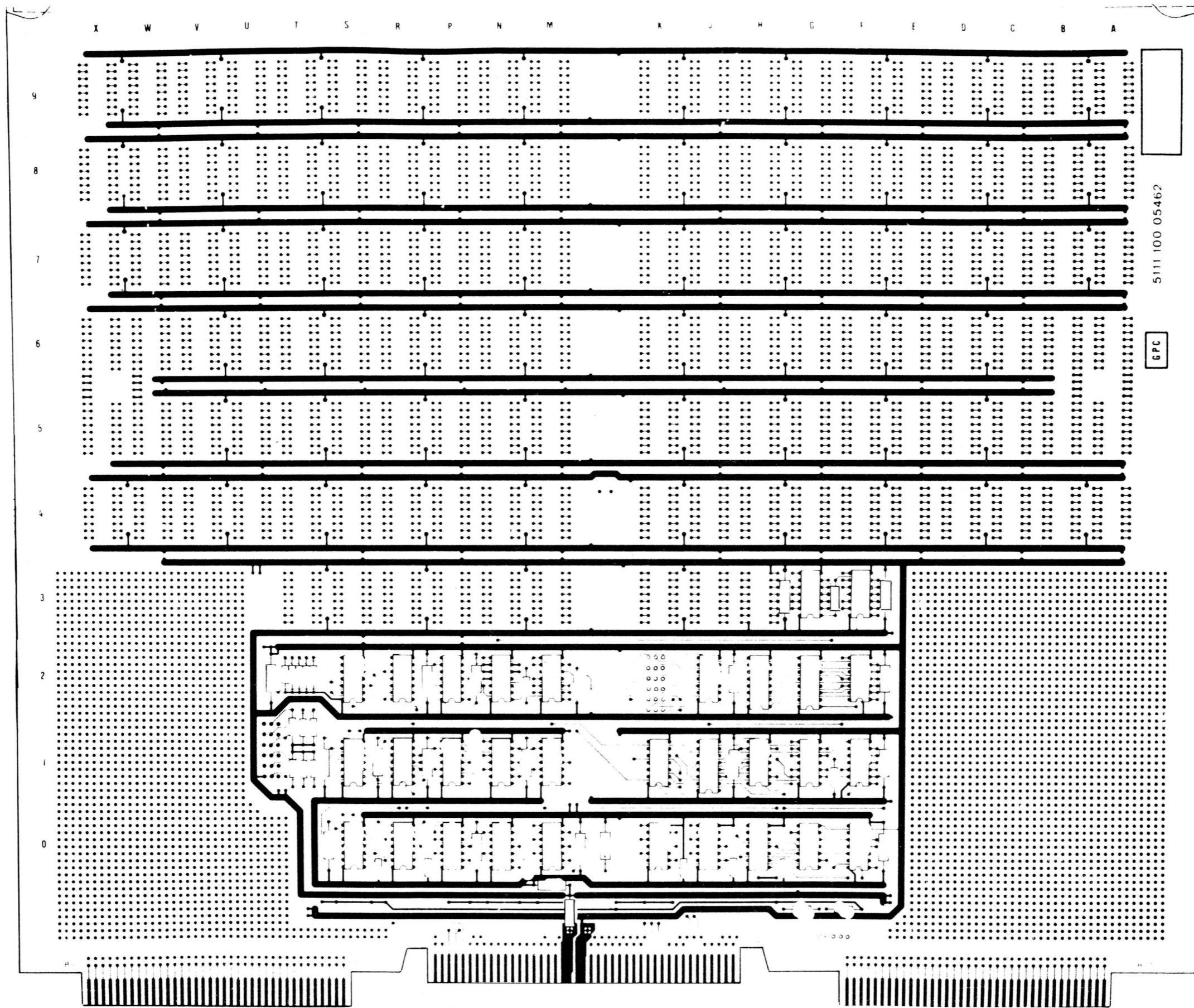


Figure 4-6 General Purpose Card Layout

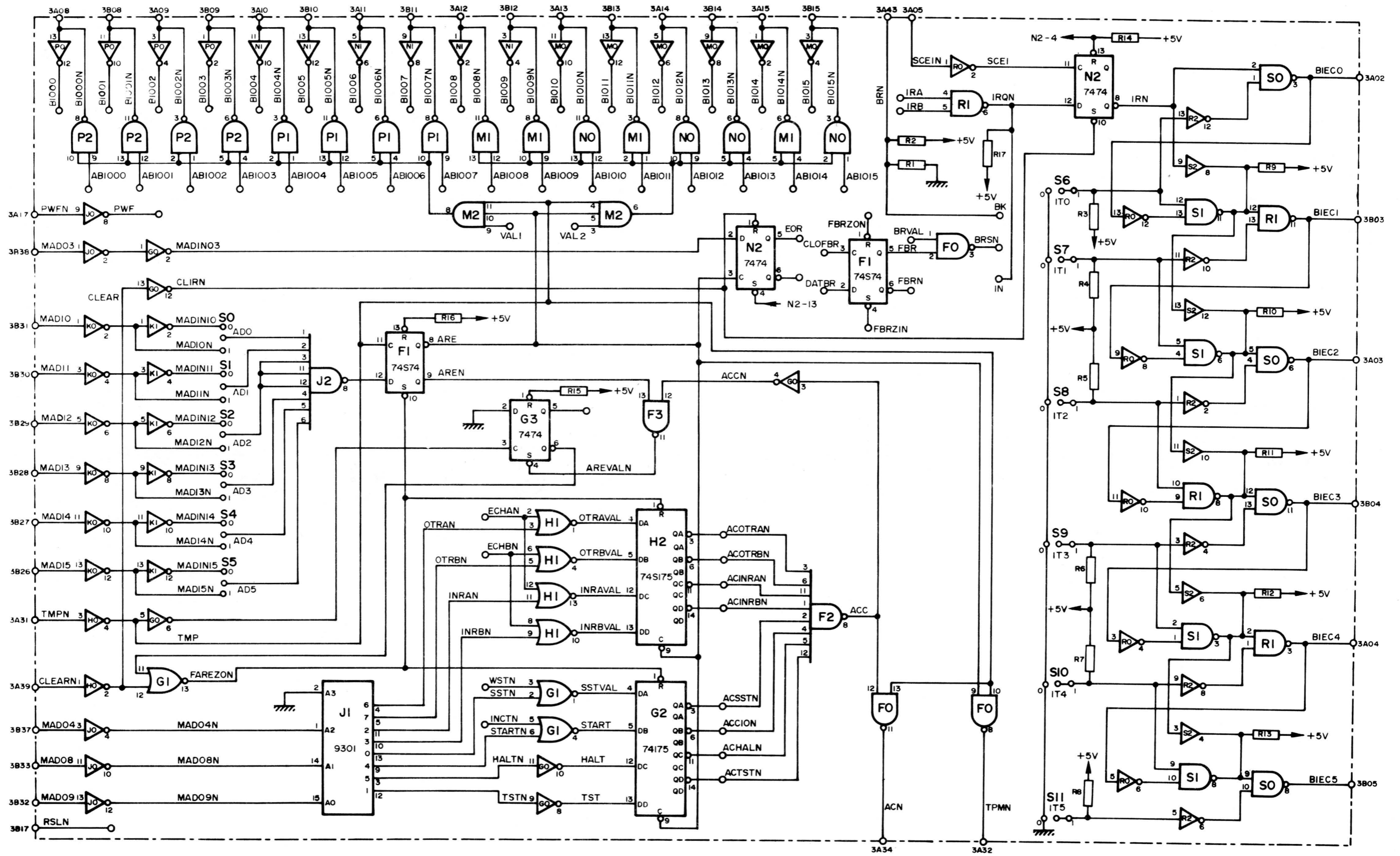


Figure 4-7 General Purpose Card Schematic

Table 4-10 General Purpose Card Parts List

Reference	Description	12NC Code
IC 1,4,10,12,13,15,23. IC 9,16,22. IC 11. IC 18,19. IC 20. IC 21. IC 24,30. IC 25. IC 26,29. IC 28. IC 31. IC 27. IC 17. IC 2,3,5,6,7,8,14.	Printed circuit Integrated Circuit 1801 Integrated Circuit 7404 Integrated Circuit 7438 Integrated Circuit 7402 Integrated Circuit 74574 Integrated Circuit 7417 Integrated Circuit 7474 Integrated Circuit 74511 Integrated Circuit 7430 Integrated Circuit 74175 Integrated Circuit 7400 Integrated Circuit 745175 Integrated Circuit 9301 Integrated Circuit REC 0613	5111 100 05462
C 7,8. C 2,5,12-15,17-26. C 3,4,6,9,10,11,16.	Capacitor 68μF, 16V, CTS. Capacitor 3.3μF, 16V, CTS. Capacitor 10μF, ± 20%, ceramic.	
R 1. R 2,9,10,11,12,13. R 3-8,14-17.	Resistor 220Ω, 0.250W, ± 5%. Resistor 390Ω, 0.250W, ± 5%. Resistor 1KΩ, 0.250W, ± 5%.	
L 1,2. S 0-12.	Inductance. U-Link.	

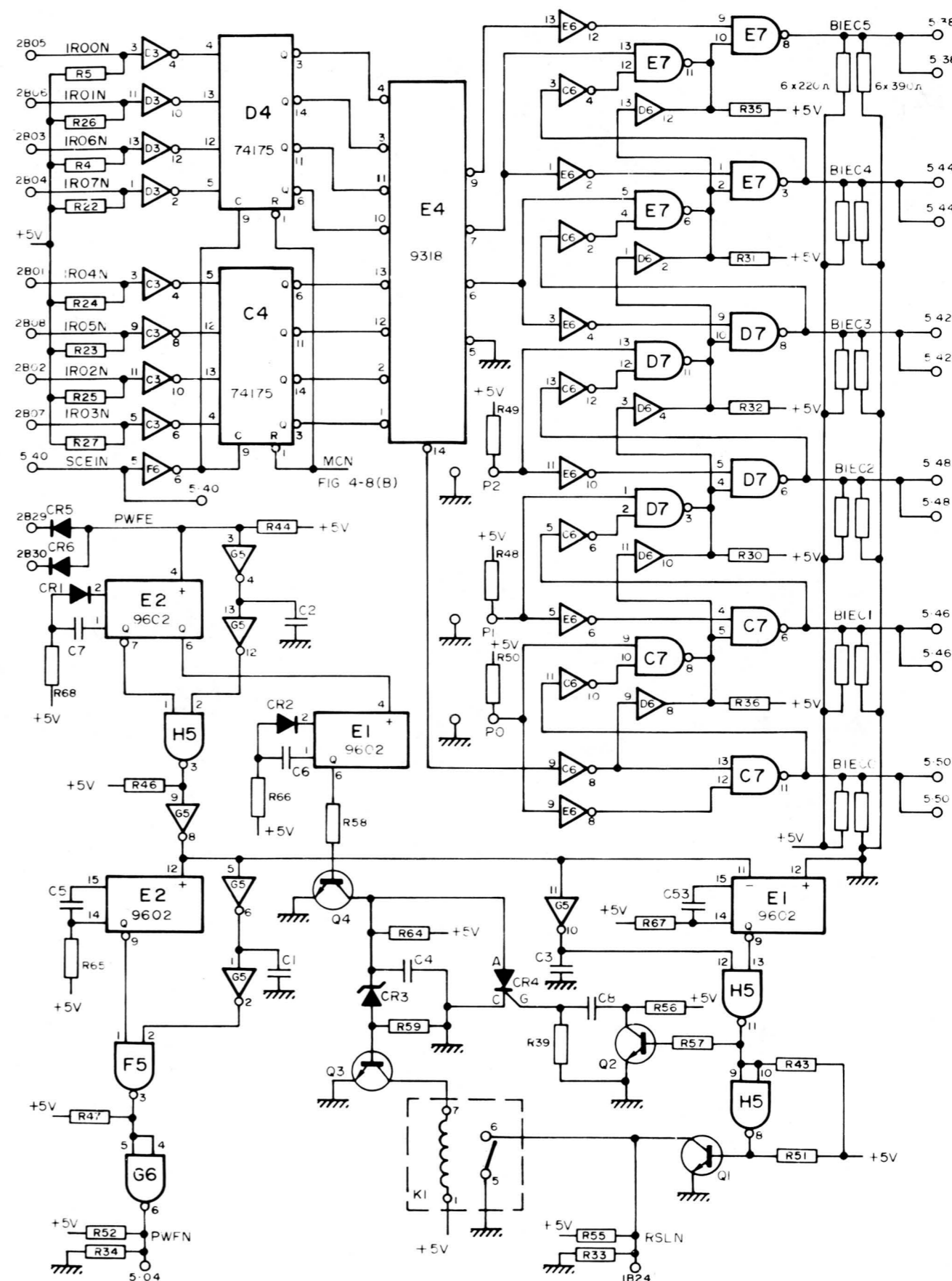


Figure 4-8(A) TAIE Card Schematic

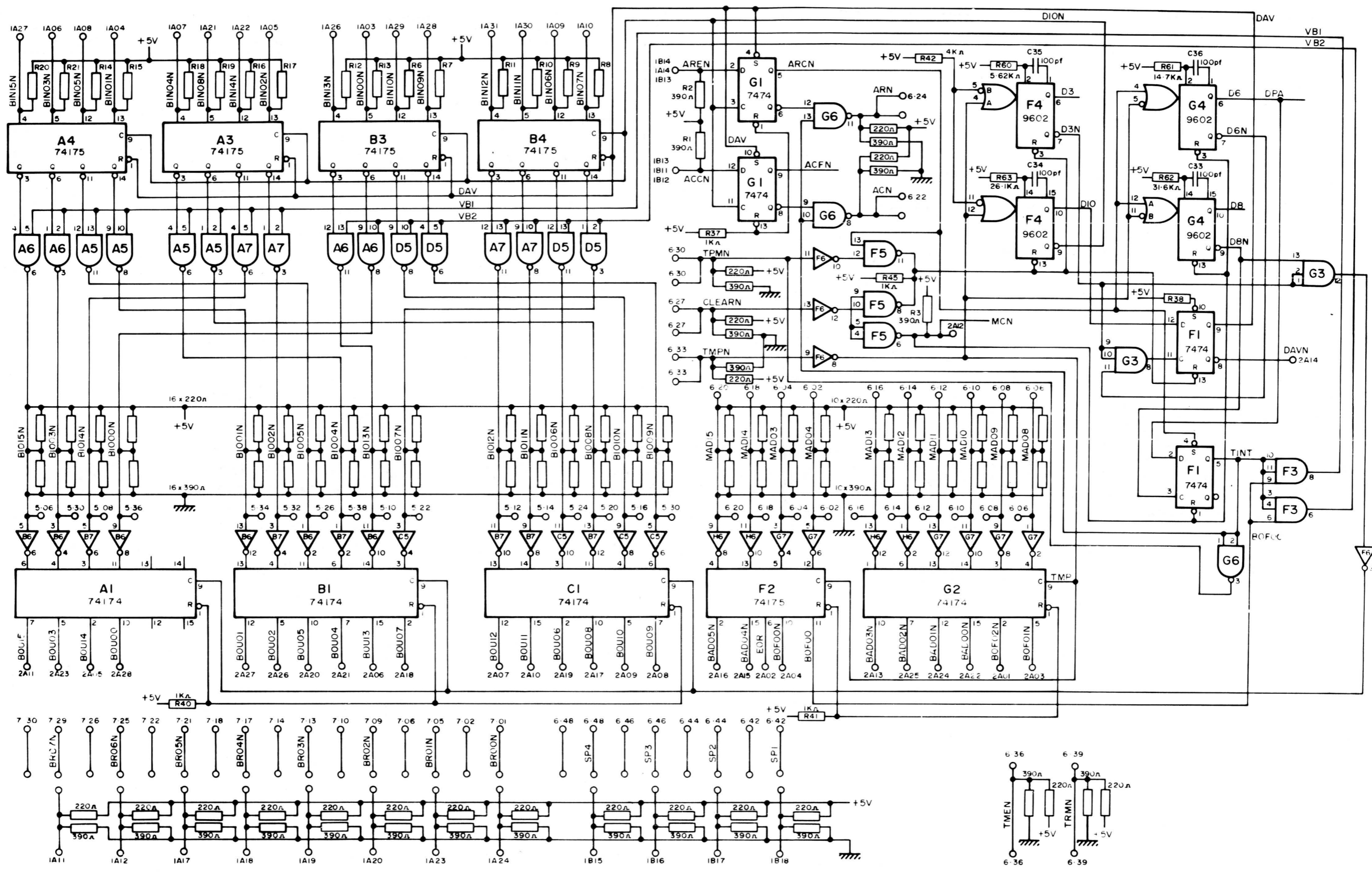


Figure 4-8(B) TAIE Card Schematic

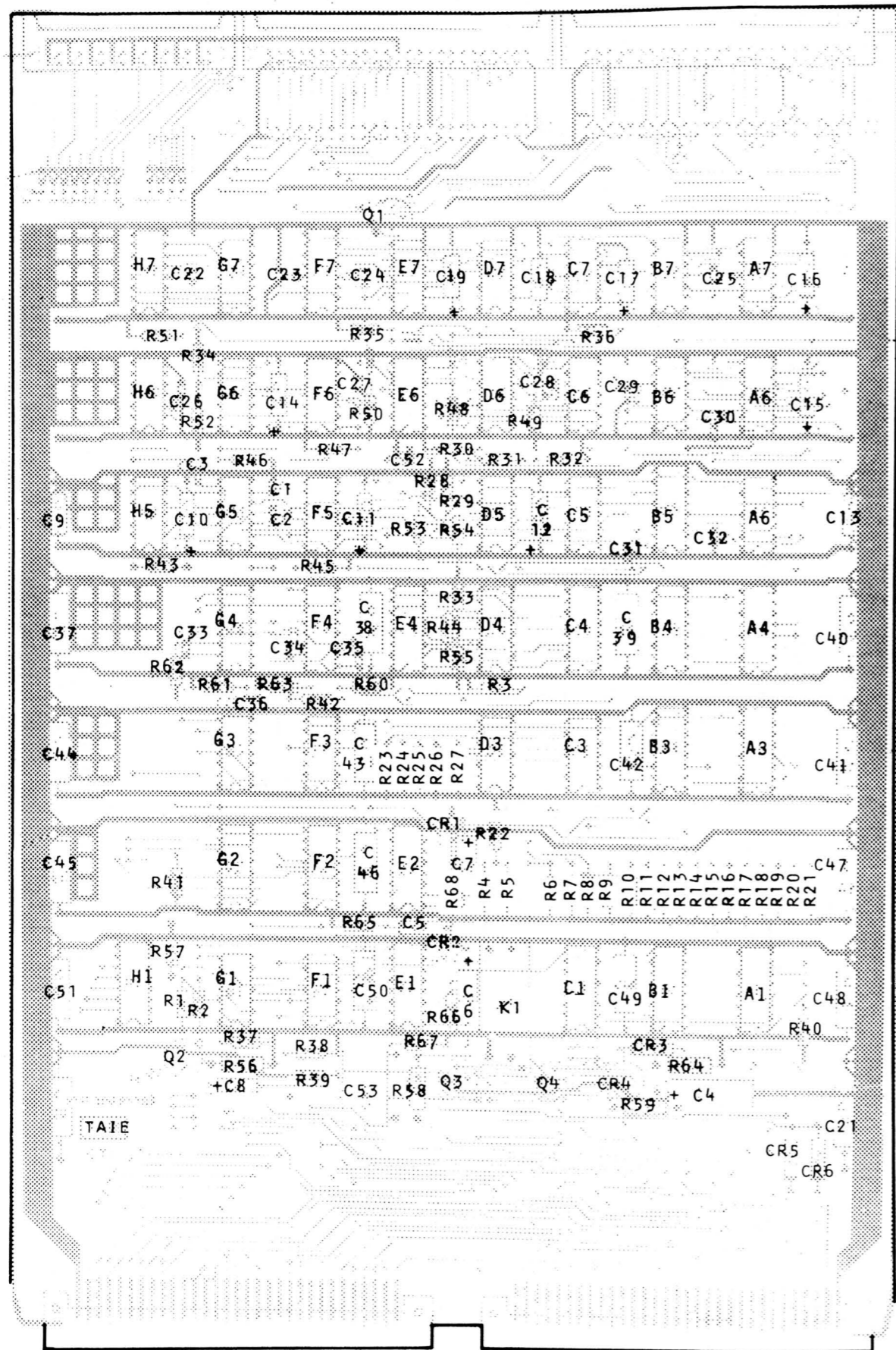


Figure 4-9 TAIE Card Layout

Table 4-11 TAIE Parts List

Reference	Description	12NC-Code
	TAIE CARD	5111 199 81440
	U-link	
A5,A6,A7,C7,D5,D7,E7,F5,G6,H5	IC 1801	
A1,B1,C1,G2	IC 74174	
G1,F1	IC 7474	
A3,A4,B3,B4,C4,D4,F2	IC 74175	
F3,G3	IC 74H11	
E1,E2,F4,G4	IC 9602	
E4	IC 9318	
B5,F7,H1,H7	IC Ternet Resistors	
C3,D3,E6,G5	IC 7404	
D6	IC 7417	
B6,B7,C5,C6,F6,G7,H6	IC REC 0612	
R37 thru R51	Resistor, 1K Ω , +5%, 0,25W	
R1 thru R36	Resistor, 390 Ω , +5%, 0,25W	
R60	Resistor, 5,62K Ω , +1%, 0,125W	
R61	Resistor, 14,7K Ω , +1%, 0,125W	
R62	Resistor, 31,6K Ω , +1%, 0,125W	
R63,R68	Resistor, 26,1K Ω , +1%, 0,125W	
R57	Resistor, 1,6K Ω , +5%, 0,25W	
R65	Resistor, 12K Ω , +5%, 0,25W	
R66	Resistor, 24K Ω , +5%, 0,25W	
R56,R58,R59	Resistor, 300 Ω , +5%, 0,5W	
R64	Resistor, 100 Ω , +5%, 0,5W	
R67	Resistor, 10K Ω , +1%, 0,125W	
R52 thru R55	Resistor, 220 Ω , +5%, 0,25W	
Q1	Transistor BSX60	
Q2,Q3,Q4	Transistor 2N2219	
K1	Relay MRMD 15005	
CR1,CR2	Diode BAX 13	
CR3	Diode 1N746A	
CR5,CR6	Diode AAZ 18	
CR4	Thyristor 2N1595	
C53	Capacitor, 0,33 μ f, 10%, 100V, MPR	
C1,C2,C3	Capacitor, 470pf, +10%, 100V, cer.plat	
C4	Capacitor, 47 μ f, 25V, FITCO	
C5	Capacitor, 0,1 μ f, 10%, 100V, MPR	
C6	Capacitor, 22 μ f, 16V, CTS13	
C7	Capacitor, 33 μ f, 10V, CTS13	
C8	Capacitor, 1 μ f, 35V, CTS13	
C9 thru C19	Capacitor, 3,3 μ f, 16V, CTS13	
C20, C21	Capacitor, 10 μ f, 15V, FITCO	
C22 thru C32, C52	Capacitor, 10000pf, 40V, cer.plat	
C33 thru C36	Capacitor, 100pf, 2%, 63V, cer.plat	
C37 thru C51	Capacitor, 0,01 μ f, -10%, 100V, MAC	

Table 4-12 AIE Parts List

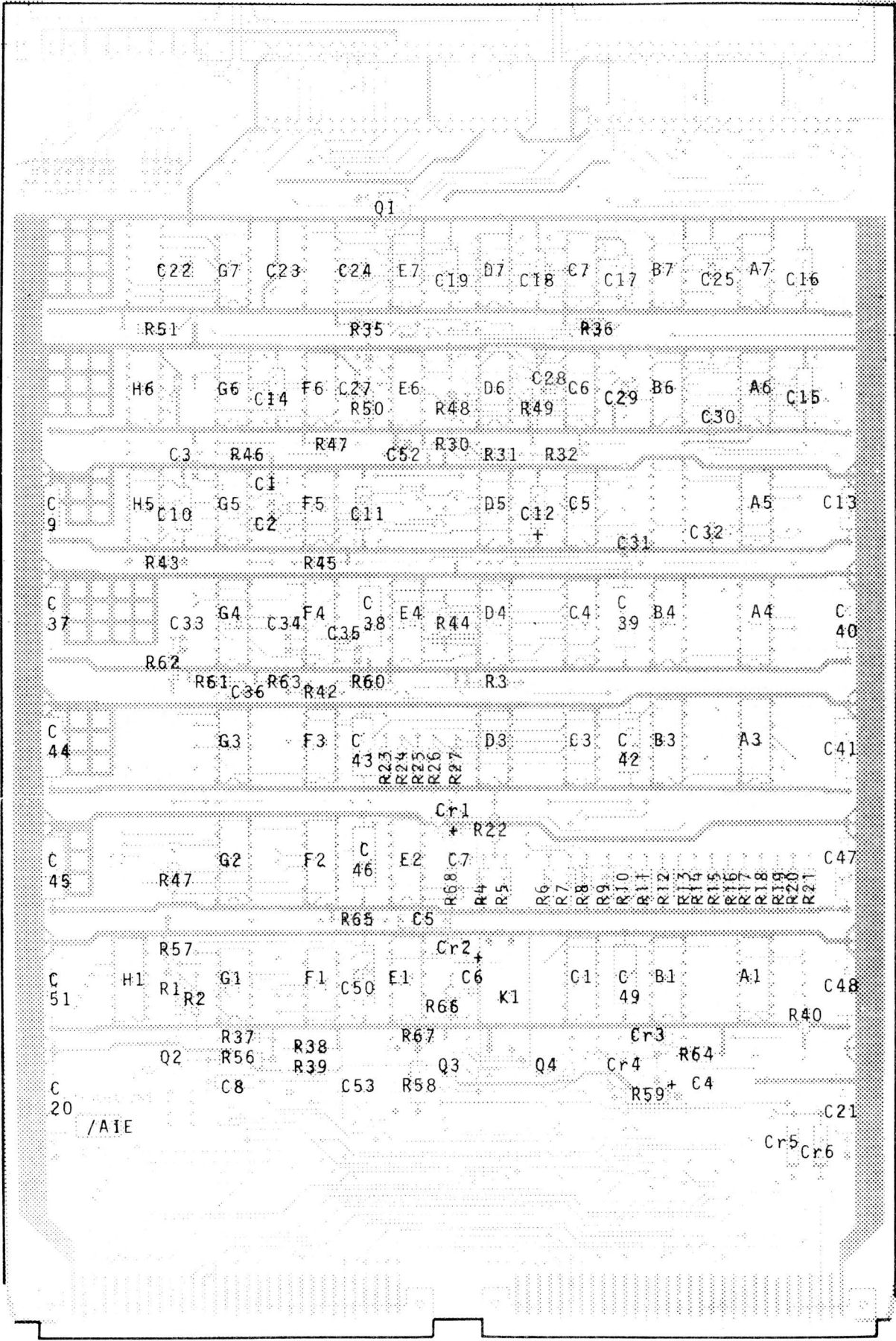


Figure 4-10 AIE Card Layout

Reference	Description	12NC-Code
	AIE CARD	5111 199 78450
A5,A6,A7,C7,D5,D7,E7,F5,G6,H5	IC 1801	
A1,B1,C1,G2	IC 74174	
G1,F1	IC 7474	
A3,A4,B3,B4,C4,D4,F2	IC 74175	
F3,G3	IC 74H11	
E1,E2,F4,G4	IC 9602	
E4	IC 9318	
H1	IC Ternet Resistors	
C3,D3,E6,G5	IC 7404	
D6	IC 7417	
B6,B7,C5,C6,F6,G7,H6	IC REC 0612	
R37 thru R51	Resistor, 1K Ω , \pm 5%, 0, 25W	
R1 thru R27,R30,R31,R32,R35,R36	Resistor, 390 Ω , \pm 5%, 0, 25W	
R60	Resistor, 5,62K Ω , \pm 1%, 0, 125W	
R61	Resistor, 14,7K Ω , \pm 1%, 0, 125W	
R62	Resistor, 31,6K Ω , \pm 1%, 0, 125W	
R63,R68	Resistor, 26,1K Ω , \pm 1%, 0, 125W	
R57	Resistor, 1,6K Ω , \pm 5%, 0, 25W	
R65	Resistor, 12K Ω , \pm 5%, 0, 25W	
R66	Resistor, 24K Ω , \pm 5%, 0, 25W	
R56,R58,R59	Resistor, 300 Ω , \pm 5%, 0, 5W	
R64	Resistor, 100 Ω , \pm 5%, 0, 5W	
R67	Resistor, 10K Ω , \pm 1%, 0, 125W	
Q1	Trnsistor BSX 60	
Q2,Q3,Q4	Trnsistor 2N2219	
K1	Relay MRMD 15005	
CR1,CR2	Diode BAX 13	
CR3	Diode 1N746A	
CR5,CR6	Diode AAZ 18	
CR4	Thyristor 2N1595	
C53	Capacitor, 0,33 μ F, 10%, 100V, MPR	
C1,C2,C3	Capacitor, 470pf, \pm 10%, 100V, cer.plat	
C4	Capacitor, 47 μ F, 25V, FITCO	
C5	Capacitor, 0,1 μ f, 10%, 100V, MPR	
C6	Capacitor, 22 μ f, 16V, CTS13	
C7	Capacitor, 33 μ f, 10V, CTS13	
C8	Capacitor, 1 μ f, 35V, CTS13	
C9 thru C19	Capacitor, 3,3 μ f, 16V, CTS13	
C20,C21	Capacitor, 10 μ f, 25V, FITCO	
C22 thru C32-C52	Capacitor, 10000pf, 40V, cer. plat	
C33 thru C36	Capacitor, 100pf, 2%, 63V, cer. plat	
C37 thru C51	Capacitor, 0,01 μ f, \pm 10%, 100V, MAC	

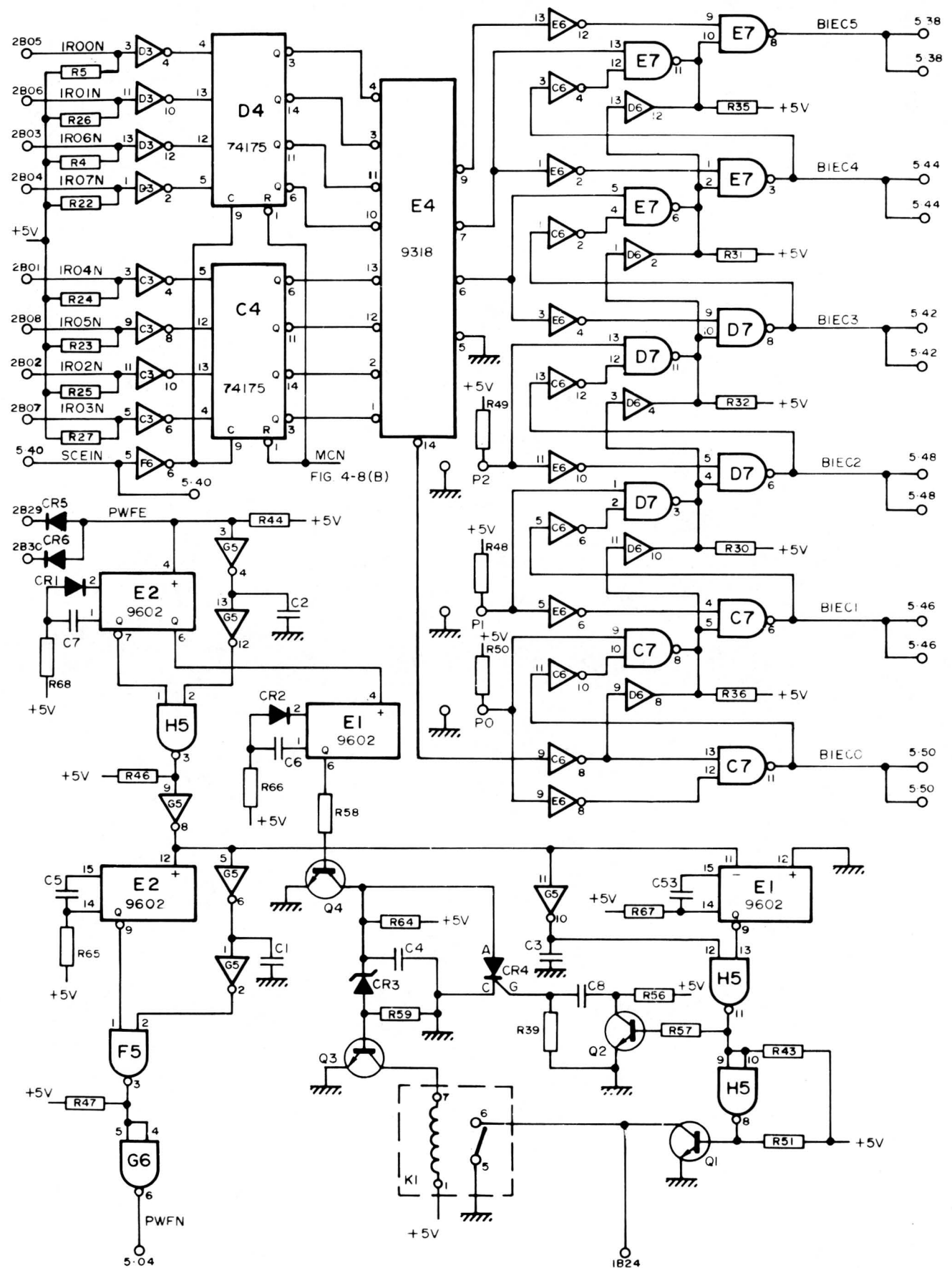


Figure 4-11(A) AIE Card Schematic

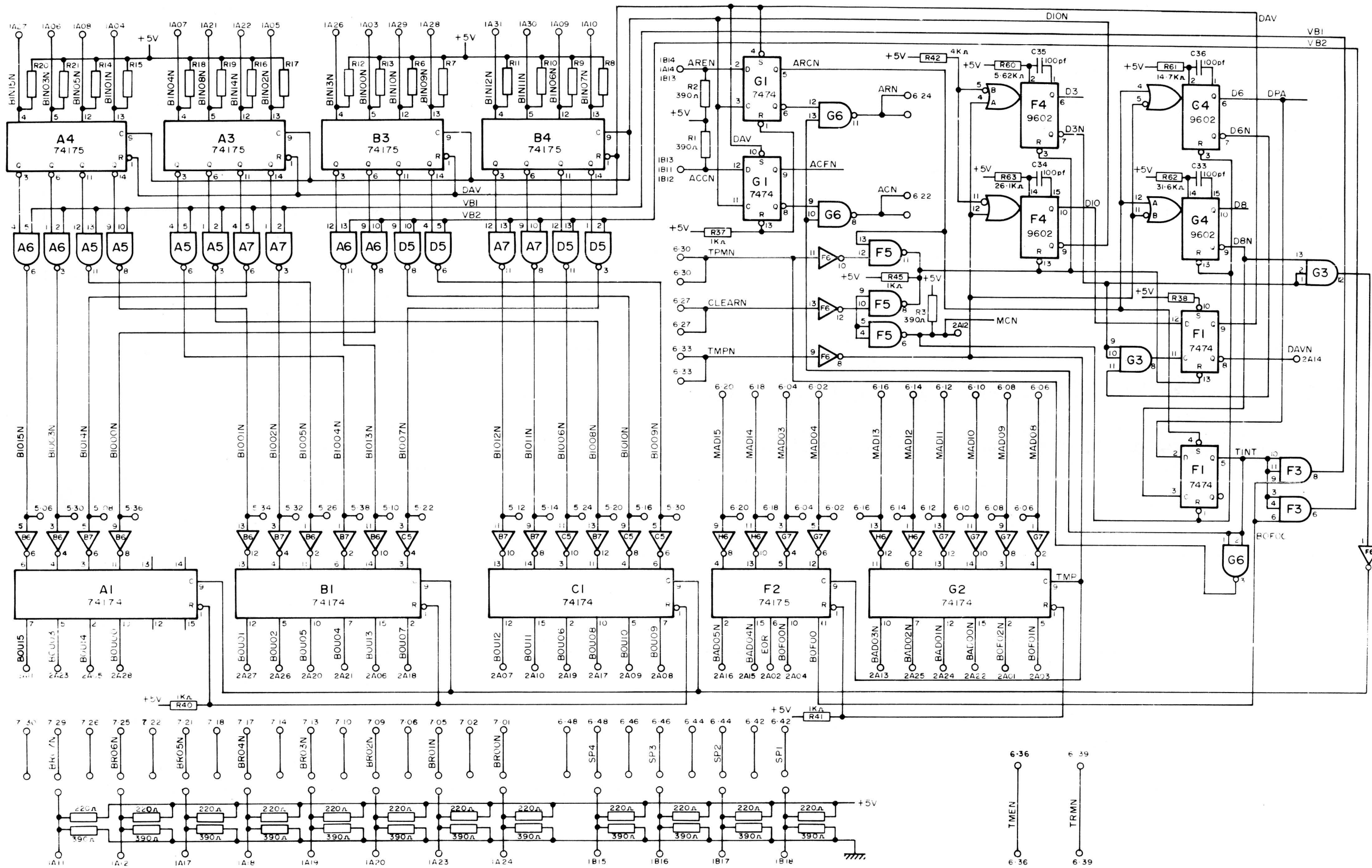


Figure 4-11(B) AIE Card Schematic

Control Panel Parts List

Reference	Description	12NC Code
	Control Panel	5111 199 81450
	Front Cover	79700
	Printed Circuit, equipped	82070
	Circuit Board	100 05213
IC18, IC19	Integrated Circuit 74729	
IC17	7410	
IC13	7404	
IC12	9602	
IC11, IC15	7432	
IC1, IC7, IC8	7416	
IC3, IC4, IC5, IC10	1801	
IC14	7420	
IC16	7400	
IC2, IC6, IC9	REC 0612	
C19, C20	Capacitor, 47 μ F, 20V,	CTS13
C1	Capacitor, 4700pF, 100V, \pm 10	cer plat
C2 thru C18, C21	Capacitor, 3900pF, 100V, \pm 10	cer plat
L1, L2, L3	Inductor	
R1	Resistor, 10K Ω , 0.250W, \pm 5	
R2	Resistor, 46.4K Ω , 0.125W, \pm 1	
R23 thru R38	Resistor, 330 Ω , 0.250W, \pm 5	
R3 thru R22, R39, R40	Resistor, 1K Ω , 0.250W, \pm 5	
	Toggle Switch 7101 LYCG	
	Push Button Switch 2RT - TFB	
	Lamp 6V 30mA ref. 2306	
	Jumper Block F088	

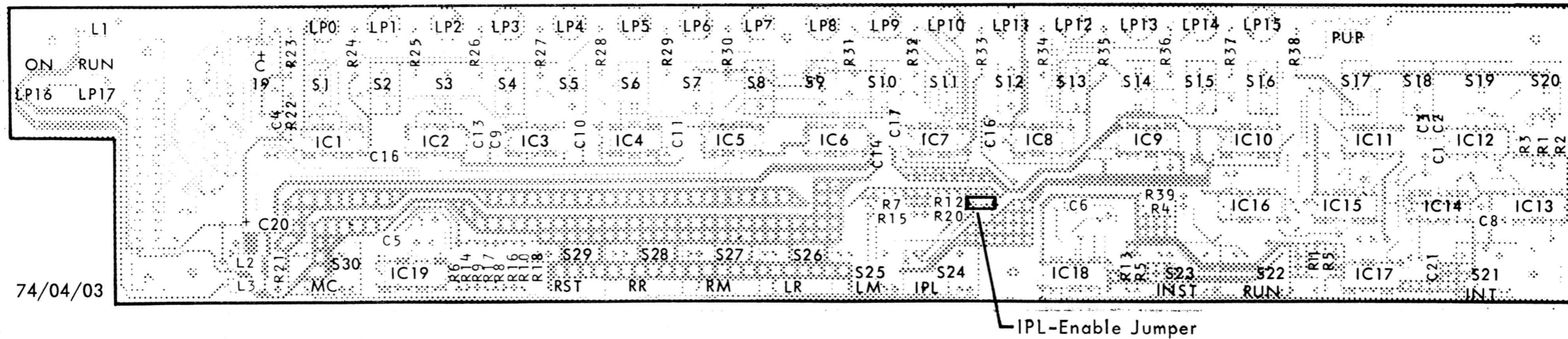
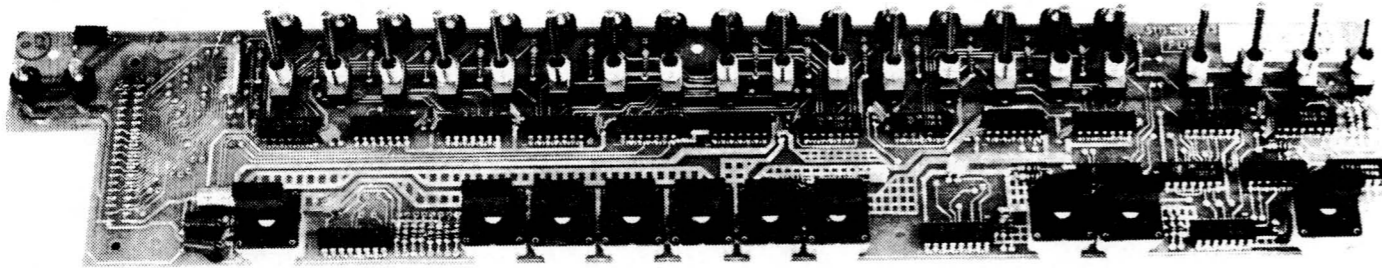
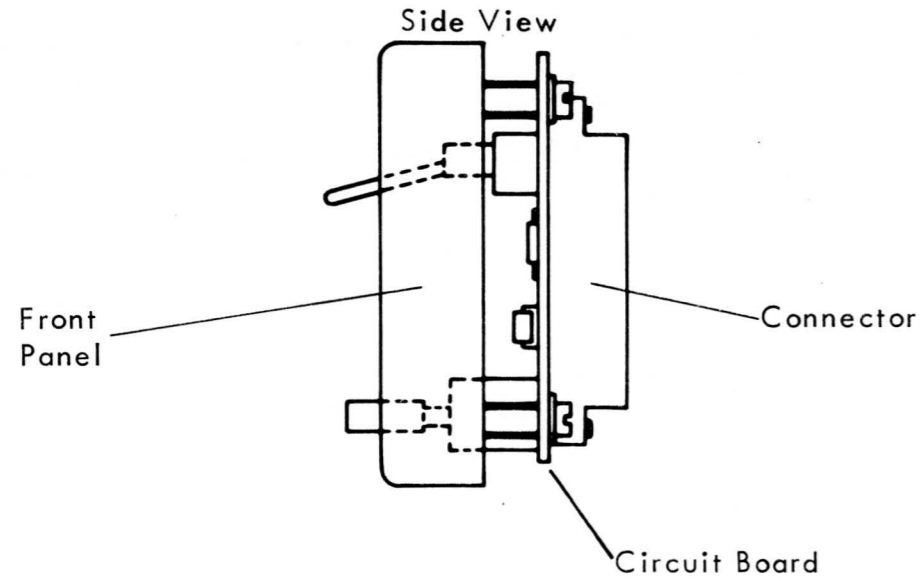


Figure 4-12A Standard Control Panel Layout

Extended Control Panel Parts List

Reference	Description	12NC Code
U1,3,5,6,7,29.	Printed Circuit	5111 100 05764
U2,4,8,26.	Integrated circuit 1801	
U28.	Integrated circuit REC 0613	
U27.	Integrated circuit 74S11	
U9,18,25.	Integrated circuit 74500	
U10,13,16,19,22.	Integrated circuit 7417	
U11,14,17,20,23.	Integrated circuit 74157	
U12,15,21,24.	Integrated circuit 74S169	
C1-17,19-39.	Capacitor 10nF, ceramic.	
C18.	Capacitor 560pF, 10%, ceramic.	
	Capacitor 22μF, 10V, FITC0.	
R18,19,20,22,23,24.	Resistor 1K Ω , 0.25W, 5%.	
R1-17.	Resistor 330 Ω , 0.25W, 5%.	
R21.	Resistor 100 Ω , 0.25W, 5%.	
LPI-17.	Lamp 6V 30 mA ref. 2306.	

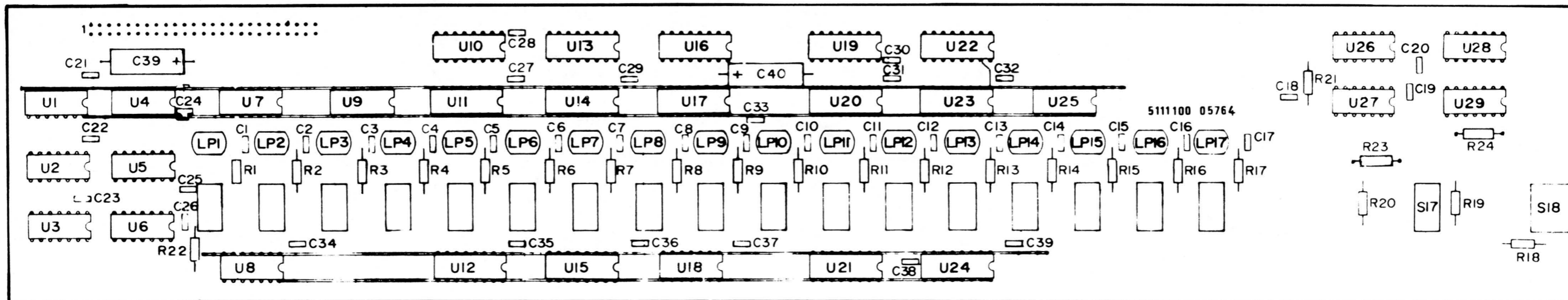


Figure 4-9B Extended Control Panel Layout

SECTION V

POWER SUPPLIES

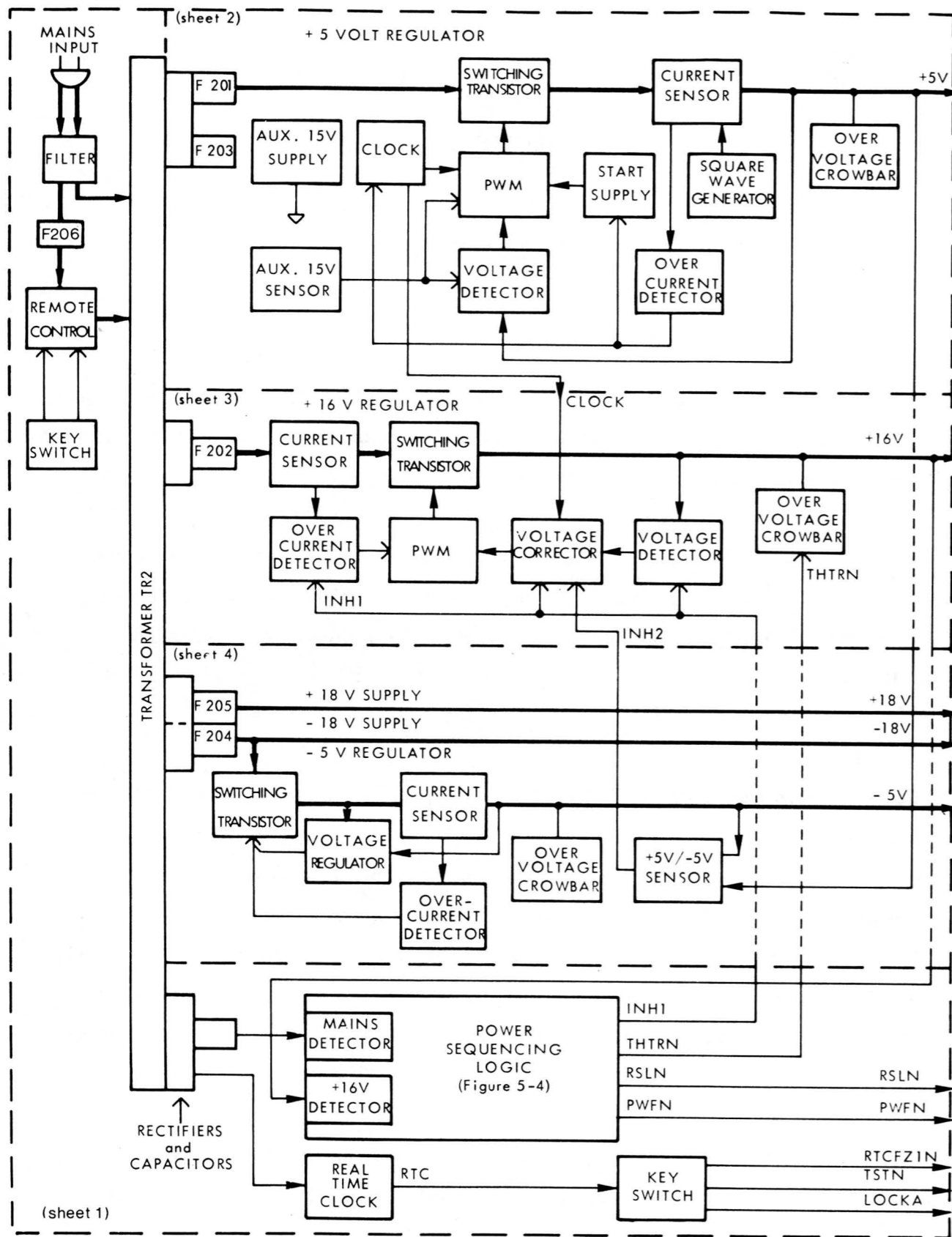
5.1 GENERAL

Each basic chassis contains its own power supply (Table 4-1). The power supply with the M4 chassis provides regulated +5V at 43 amps, +16V at 9 amps, and -5V at 2 amps, as well as unregulated +18 and -18 volts at 2 amps each. The M5 chassis contains two power supplies : one identical to the M4 supply, and a second supply that differs only in some component positions. The main power and logic signals of the basic supply are shown in the block diagram, Figure 5-1. The mechanical configuration and parts locations are shown in Figures 5-6 through 5-9.

5.2 INPUTS

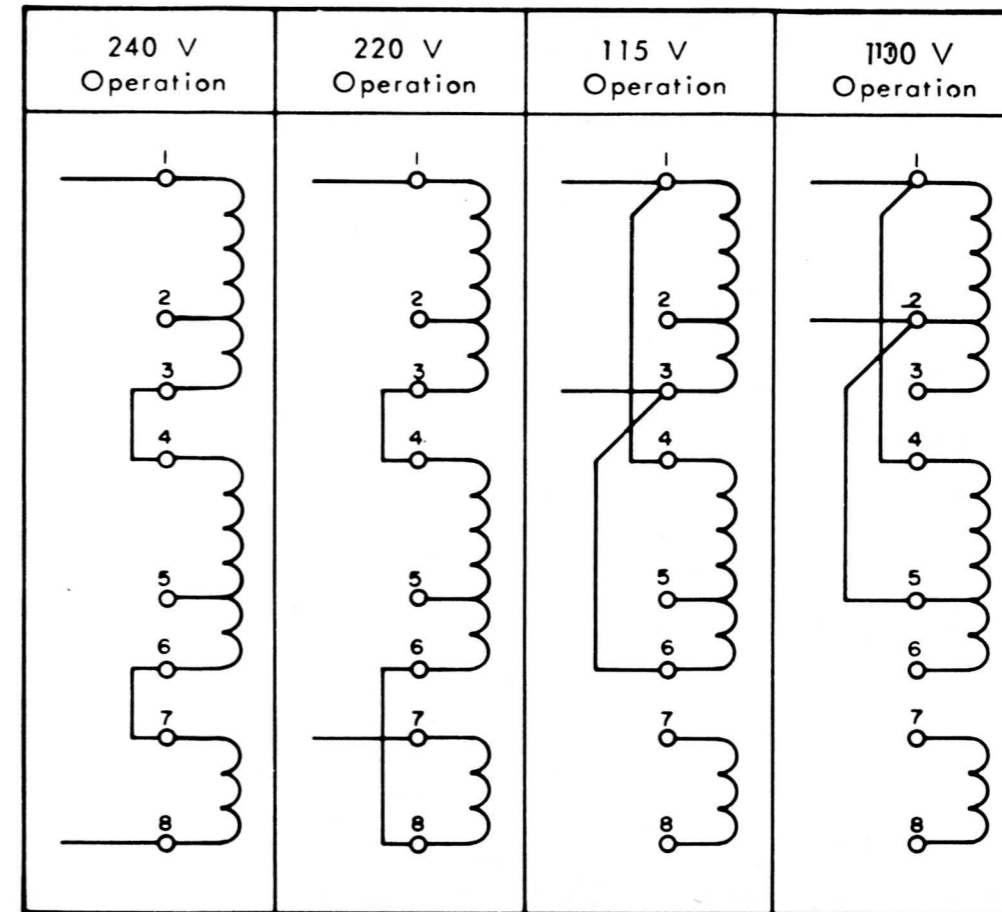
The power supply operates on voltages of 100V, 115V, 220V, or 240V, $\pm 10\%$, single phase, at either 50 Hz (± 2 Hz) or 60 Hz (± 3 Hz). The power supply is adapted to the selected input voltage by wiring the input-side of the mains transformer, T201, as shown in Figure 5-2. The transformer input connections and jumpers are made by individual plugs at the rear of the chassis (Figure 5-7). The AC input is filtered, fused, and then switched by means of a remote-control circuit. The remote-control circuit (Figure 5-3, sheet 1) is located on printed-circuit card P1.

5.3 A battery connection is provided on the power supply chassis. The purpose of an external battery supply is to maintain the +16V and part of the +5V supplies during a mains power failure. The battery connector wiring is shown on Figure 5-3, sheet 1.



Notes - The dashed-line sections show the locations on schematic diagram Figure 5-3.
 The arrows \blacktriangleright show normal flow.
 The arrows \rightarrow show the inhibit/enable signals.

Figure 5-1 P857 Power Supply Block Diagram



Point 9 of the transformer is a shield between primary and secondary windings. The shield must be connected to the ground. The fan is connected on the first 115V winding (1-3) so that it is always supplied with 115V.

Figure 5-2 Mains Input Wiring

5.4 OUTPUTS

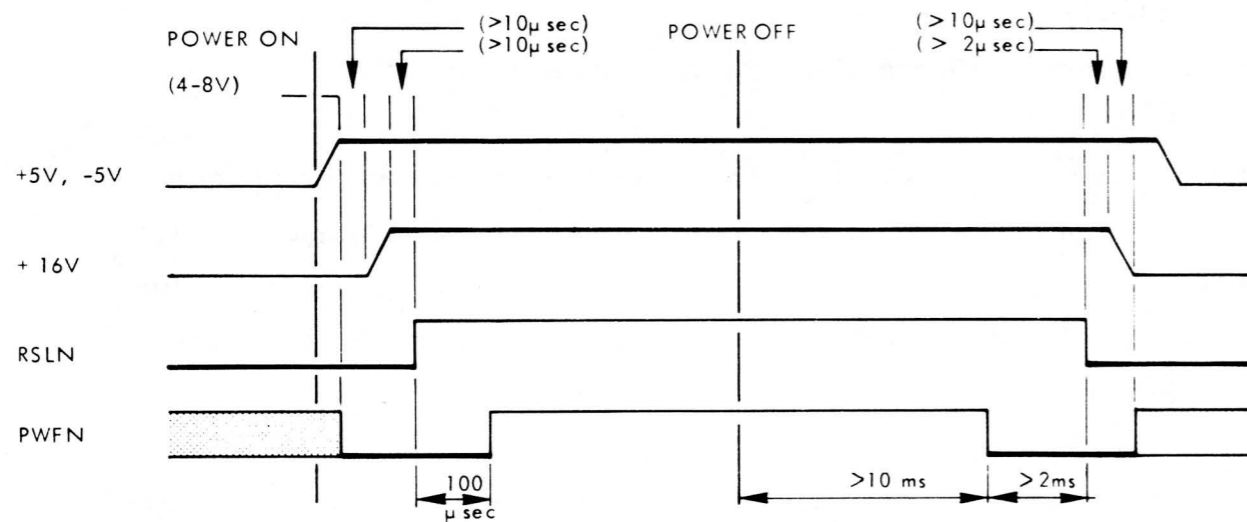
The output voltages, and maximum currents from this supply are :

- +16V, 9A, $\pm 3\%$ (due to $\pm 10\%$ mains and load variation)
- + 5V, 43A, $\pm 2\%$ (due to $\pm 10\%$ mains and load variation) $\pm 2\%$ for ripple and noise, p-p from 0 to 20 MHz.
- - 5V, 2A, $\pm 2\%$ (due to $\pm 10\%$ mains and load variation) $\pm 2\%$ for ripple and noise, p-p from 0 to 20 MHz.
- +18V, 2A, $-10\%, +70\%$ (non regulated)
- -18V, 2A, $-10\%, +70\%$ (non regulated)

The +16V supply is used by the memory (inhibit amplifiers). About 9 amps are required for 32k words of memory; the 16V, 9-amps supplies 64k words of memory with two modules working at the same time, with interleaving. The -5V supply is used by memory (about 1.5 amps) and by the interface drivers for the big disc control unit (about 1 amp). The unregulated +18V and -18V supplies are used by data-communication and teletype control units; the 18 volts are converted to 12 or 6 volts by regulators on the control-unit cards. The +5V supply is used by the P857 logic.

5.5 LOGIC SIGNALS

The RSLN (Reset Line) and PWFN (Power Failure) signals are used for power on/off sequences and automatic restart. The full logic description is given in paragraph 2.45. The power supply controls both signals with the following timing :



The power-off sequence is shown for switching off or for a power failure greater than 10 ms. Shorter line failures will not cause the "off" sequence. Once PWFN goes active (low), the sequence must continue to activate RSLN.

5.6 REAL TIME CLOCK

The Real Time Clock (RTC) is a 1µs pulse every 20ms which is sent to the RTCF

flip-flop (program status bit 12) to be used as an internal interrupt. The RTC pulse is generated by pulse-shaping the mains frequency. The RTC pulse is then sent via the control-panel key switch (positions ONRTC and LOCK) to the CPU logic RTCF flip-flop (Figure 2-8 PP). The operation of the General Flip-Flop (GF) RTCF is described in Section II. The RTCF interrupt is also shown on the Interrupts and Breaks diagram in Section I.

5.7 FUSES

The power supply contains one fuse in the mains input (F206) and additional fuses for the supplied voltages. The fuses for the supplied voltages (at the rectifier/filter outputs) are to protect the rectifiers and transformer in case of regulator failure; the regulators themselves are electronically protected. The power supply fuses are listed in the following table :

Fuse	Purpose	Schematic	Location
F206	Mains input, slow operating : 4 amp for 220V/240V inputs 8 amp for 100V/115V inputs	sheet 1	
F101	0.1 amp, mains detector	sheet 1	
F201	20 amp, +5V regulator	sheet 2	
F202	10 amp, +16V regulator	sheet 3	
F203	2 amp, auxillary 15V supply for the power-supply circuits.	sheet 2	
F204	3.15 amp, -18V and -5V supplies	sheet 4	
F205	3.15 amp, +18V supply	sheet 4	

5.8 RECTIFIER CIRCUITS

The inputs to all of the DC supplies use center-tapped, full-wave rectifiers. The AC voltage is supplied by five center-tapped secondary windings of transformer TR2. The rectifier circuits are shown on the schematic diagram (Figure 5-3) with their appropriate supplies. The +5V and +16V regulators take the positive voltage from the transformer center taps, which allows the anodes of the high-current diodes to be mounted in heat sinks at zero volts

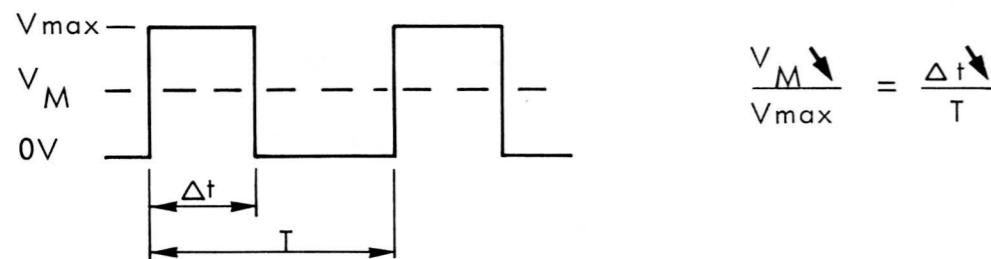
(ground). The +18V and -18V supplies use a full-wave bridge package, but each supply is using only two diodes of the bridge together with the center-tapped transformer winding. The -5V regulator takes its input voltage from the unregulated -18V supply.

5.9 +5V REGULATOR

5.10 Voltage Regulation. The +5V, 43-amp power is supplied by a switching type regulator (Figure 5-3, Sheet 2). High efficiency is obtained because of the regulator transistors (Q201, 202, and 203) being always on or off. The regulator transistors are controlled by a voltage detector and a voltage corrector. The switching frequency is controlled by an independent clock circuit common to the +5 and +16 volt supplies, and is thereby not load-dependent.

The +5V regulator controls the mean voltage V_M (following diagram) by varying the switch-on time in direct relation to the required correction.

The Δt is reduced, while the switching frequency is held constant, in order to reduce the mean output voltage.



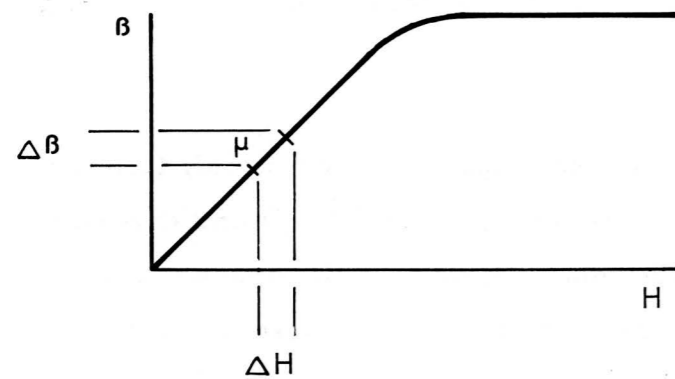
5.11 The switching regulation operates at a frequency (20 KHz, $\pm 5\%$) determined by the type 555 clock circuit, IC1. The clock provides a timing input to voltage corrector IC2. IC2 is a type 555 pulse-width modulator (PWM) which generates a pulse to switch regulator Q201/2/3 on and off. Regulation is provided by the variable pulse width from IC2. When Q201/2/3 switches on, current flows through inductor L203 to the load, increasing linearly, and charges the output capacitors. When Q201/2/3 switches off, L203 current starts to decrease, and diodes CR201/2 are forward biased; the current now flows through CR201/2, decreasing linearly.

5.12 The +5V output sense voltage is compared to a reference voltage by error amplifier IC4. (Since the type uA723 reference voltage is about 7 volts, the +5V supply ($< 7V$) adjusts the reference level.) As the sense voltage increases, the output current at IC4, pin 9 increases and the Q8 collector voltage to PWM IC2 decreases. The reduced threshold voltage at pin 5 of IC2 decreases the pulse width output from the voltage corrector. The narrower pulse to Q201/2/3 (via transformer T1) results in less "on time" for the regulator transistors, and reduces the final output voltage.

5.13 The 1:1 transformer T1 is used to couple the high-current supply with sensitive, low-current control circuits. This allows higher-voltage regulation control with resultant higher efficiency and higher-speed operation for the voltage regulation. The coupling transformer is supplied with 3.5 volts at high current during power-on time by starting-supply IC5, Q12. Once the +5V supply is up, T1 is supplied by the +5 volts via diode CR1; the emitter of Q12 is back-biased and the starting supply is held off. The starting supply is inhibited (along with the clock) by an overcurrent cutoff.

5.14 Overvoltage Crowbar. An overvoltage detection circuit (crowbar) for the +5V supply is made up of a threshold detector (Q101) and thyristor (CR101). If the supply output voltage increases beyond the nominal value, between 6 and 7 volts, Q101 switches on and the thyristor fires to short-circuit the supply.

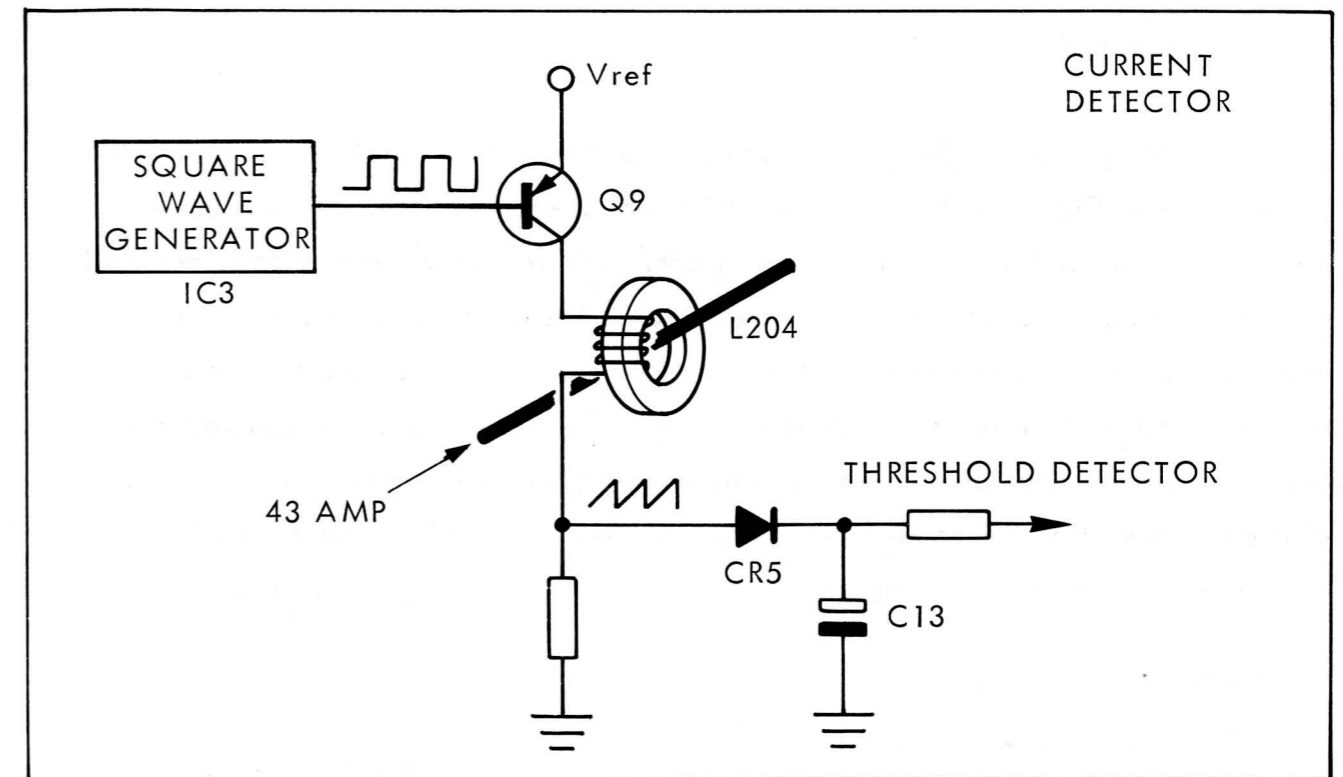
5.15 Overcurrent Detection. Current detection on this 43-amp supply is performed by passing the +5V line through the center of a variable- μ ferrite core (L204) which has a sensing coil wound about it. Direct current through the +5V line produces a magnetic field (H) which causes a magnetic flux in the core. The flux density (B) and the properties of the core material determine the permeability (μ) of the core; thus, any change in the current through the core (I) changes the field strength (H) which changes the permeability (μ).



$$B = \mu H$$

$$\mu = \frac{B}{H}$$

5.16 A type 555 square-wave generator (IC3, Q7, Q9) produces a voltage square wave which passes through the L204 coil (from pin A to B) to the over-current detector (IC6). The inductance (L) of the coil retards the leading edge of the squarewave to produce a current sawtooth output to the detector (following diagram). Since the inductance of the coil (L) is approximately $L = \mu N^2$ (where N is the number of turns, a constant), a change in μ causes a change in the sensing current through the coil (ΔI). The height of the sawtooth is thus directly dependant on the coil's L, as follows : $E = L \Delta I / \Delta T$, with ΔT and E constants from the squarewave generator. Excess current therefore increases the amplitude of the sawtooth sufficiently to trigger the threshold detector at the IC6 input.



5.17 When overcurrent detector IC6 is triggered, it produces an inhibit signal which blocks the T1 starting supply and the clock. With the clock (IC1) stopped, the +5V and +16V supplies are blocked. The starting supply to T1 must be inhibited separately because it derives its power from the unregulated part of the +5V supply which is not stopped when the IC1 clock stops.

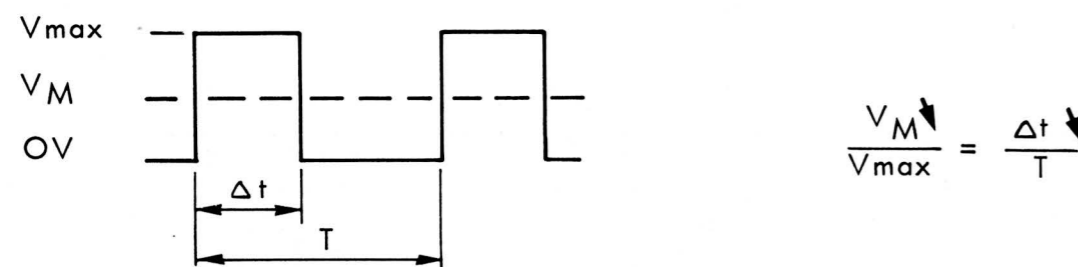
5.18 When the +5 volts is off, the squarewave output is off and transistor Q9 is on. Since the L204 coil is short for DC, the threshold detector detects a constant high and adds to the shut off condition. Negative spikes (caused by L) are removed from the Q9 squarewave output by zener diodes CR14 and CR15. The resistors shown as part of L204 are selected during manufacture to produce the same μ for all units. The relationship from current (I) to squarewave voltage is :

$$I \rightarrow H \rightarrow \mu \rightarrow L \rightarrow \Delta I \rightarrow \Delta E.$$

$\uparrow \quad \uparrow \quad \downarrow \quad \downarrow \quad \uparrow \quad \uparrow$

5.19 +16V REGULATOR

5.20 Voltage Regulation. The +16V, 9-amp power is supplied by a switching type regulator (Figure 5-3, Sheet 3). High efficiency is obtained because regulator transistor Q204 is always on or off. The regulator transistor is controlled by a voltage detector and a voltage corrector. The switching frequency is controlled by an independent clock circuit common to the +5 and +16 volt supplies, and is thereby not load-dependent. The +16V regulator controls the mean voltage V_M (following diagram) by varying the switch-on time in direct relation to the required correction. The Δt is reduced, while the switching frequency is held constant, in order to reduce the mean output voltage.



5.21 The switching regulation operates at a frequency (20 KHz) determined by the type 555 clock circuit, IC1. The clock provides a timing input to voltage corrector IC9. IC9 is a type 555 pulse-width modulator (PWM) which generates a pulse to switch regulator Q204 on and off. Regulation is provided by the variable pulse width from IC2. When Q204 switches on, current flows through inductor L202 to the load, increasing linearly, and charges the output capacitors. When Q204 switches off, L202 current starts to decrease, and diode CR203 is forward biased; the current now flows through CR203, decreasing linearly.

5.22 The +16V output sense voltage is compared to a reference voltage by error amplifier IC8. (Since the type uA723 reference voltage is about 7 volts, the +16V supply (>7V) adjusts the sense level.) As the sense voltage increases, the output current at IC8, pin 9 increases and the Q19 collector voltage to PWM IC9 decreases. The decreased threshold voltage at pin 5 of IC9 decreases

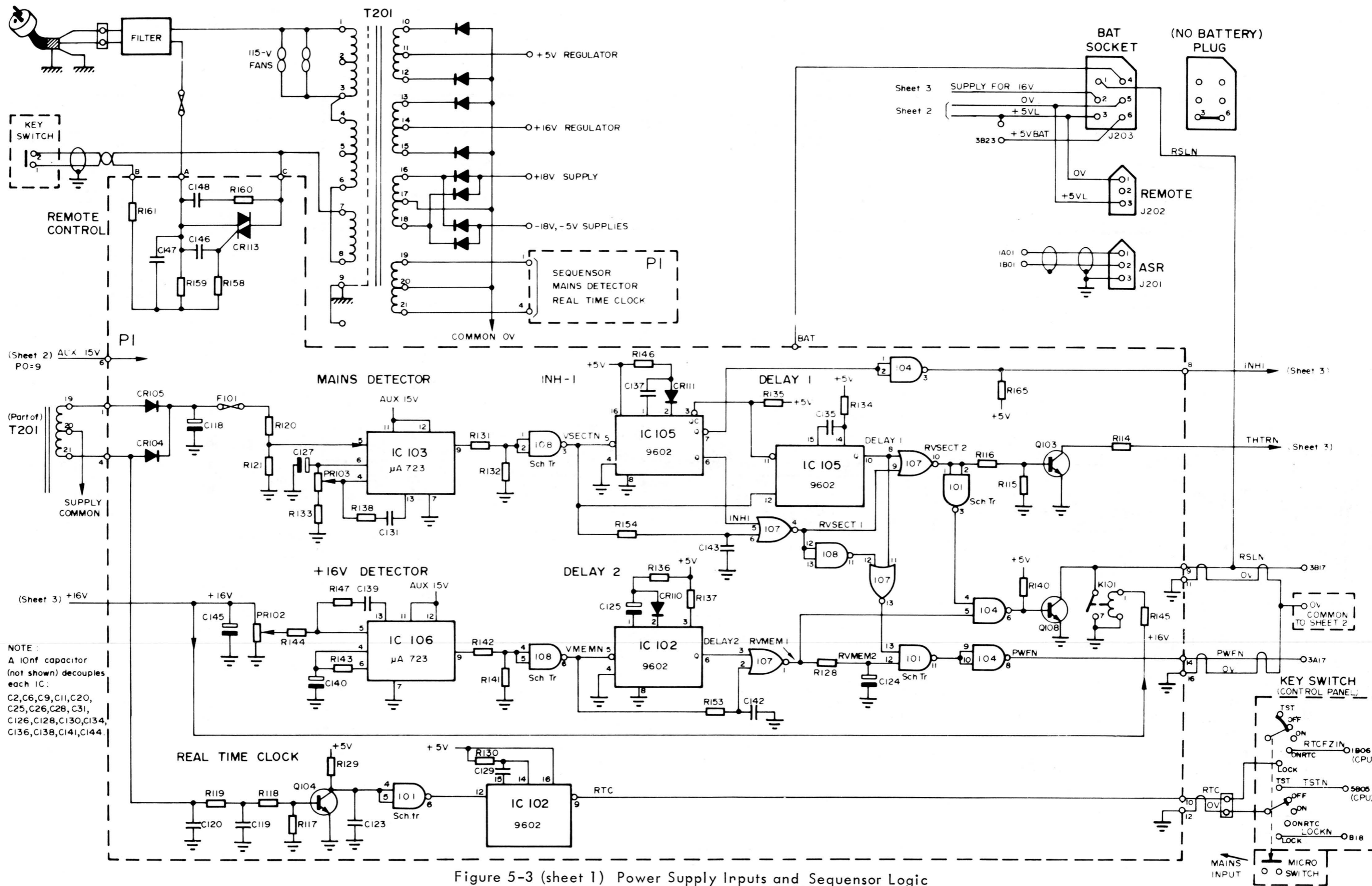
the pulse width output from the voltage corrector. The narrower pulse to Q204 (via Q20 and Q13) results in less "on time" for the regulator transistor, and reduces the final output voltage.

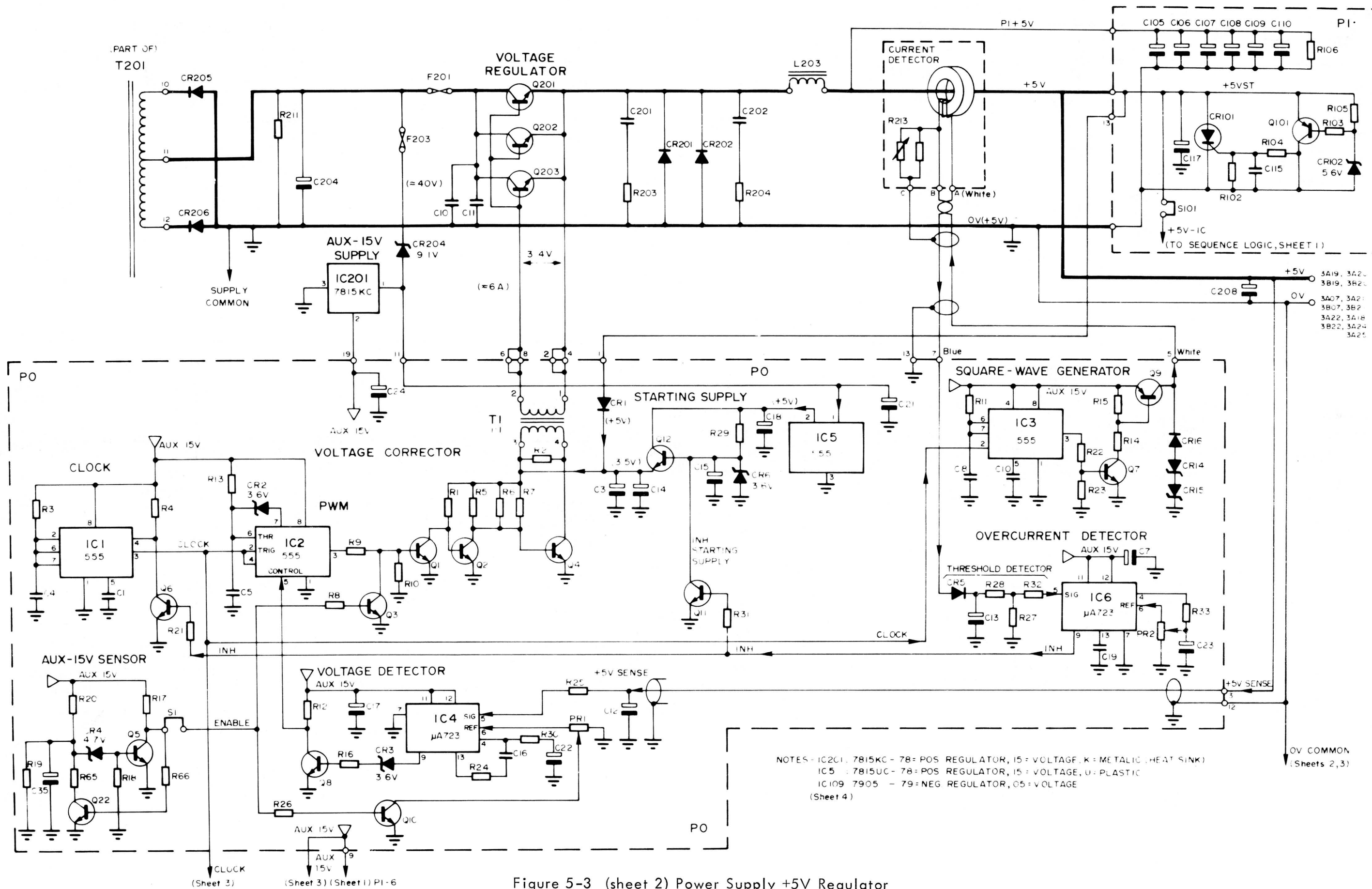
5.23 Overvoltage Crowbar. An overvoltage detection circuit (crowbar) for the +16V supply is made up of a threshold detector (Q102) and thyristor (CR103). If the supply output voltage increases beyond the nominal value, between 16.8 and 20.3 volts, Q102 switches on and the thyristor fires to short-circuit the supply. The THTR signal is used to shut off the 16-volt supply during the power-off sequence (paragraph 5-41).

5.24 Overcurrent Detection. Current sensing in the +16V supply is performed by resistor R207 in series with the supply. The lower-voltage half of R207 is used as a reference at the pin-4 input of error amplifier IC7. This reference voltage is held constant by zener diode CR8, and an adjustment is provided by trimpot PR4. The higher-voltage half of R207 is used as a sense input at IC7, pin 3. Zener diode CR10 is used to reduce the sense-input voltage. An increase in current through R207 will increase the sense voltage (developed across R45) in relation to the reference voltage. The resultant high output from IC7, pin 2, via Q15, will gate on thyristor CR13 and shut off the +16 volt supply.

5.25 Inhibit Signals. The inhibit signal INH1 delays operation of the +16V regulator during initial power-on time to produce a slow rise of the +16 volts. INH1 is active (high) for approximately 300 ms after power on (Figure 5-5). During this initial delay time, the overcurrent detector is blocked via Q17 and the overvoltage circuit is blocked via Q16 and Q18.

5.26 The inhibit signal INH2 blocks the +16V regulator if either the +5 volts or -5 volts are not present. Loss of either voltage drops out relay K102 (Figure 5-3, sheet 4) which then grounds the output of the overvoltage corrector circuit (sheet 3).





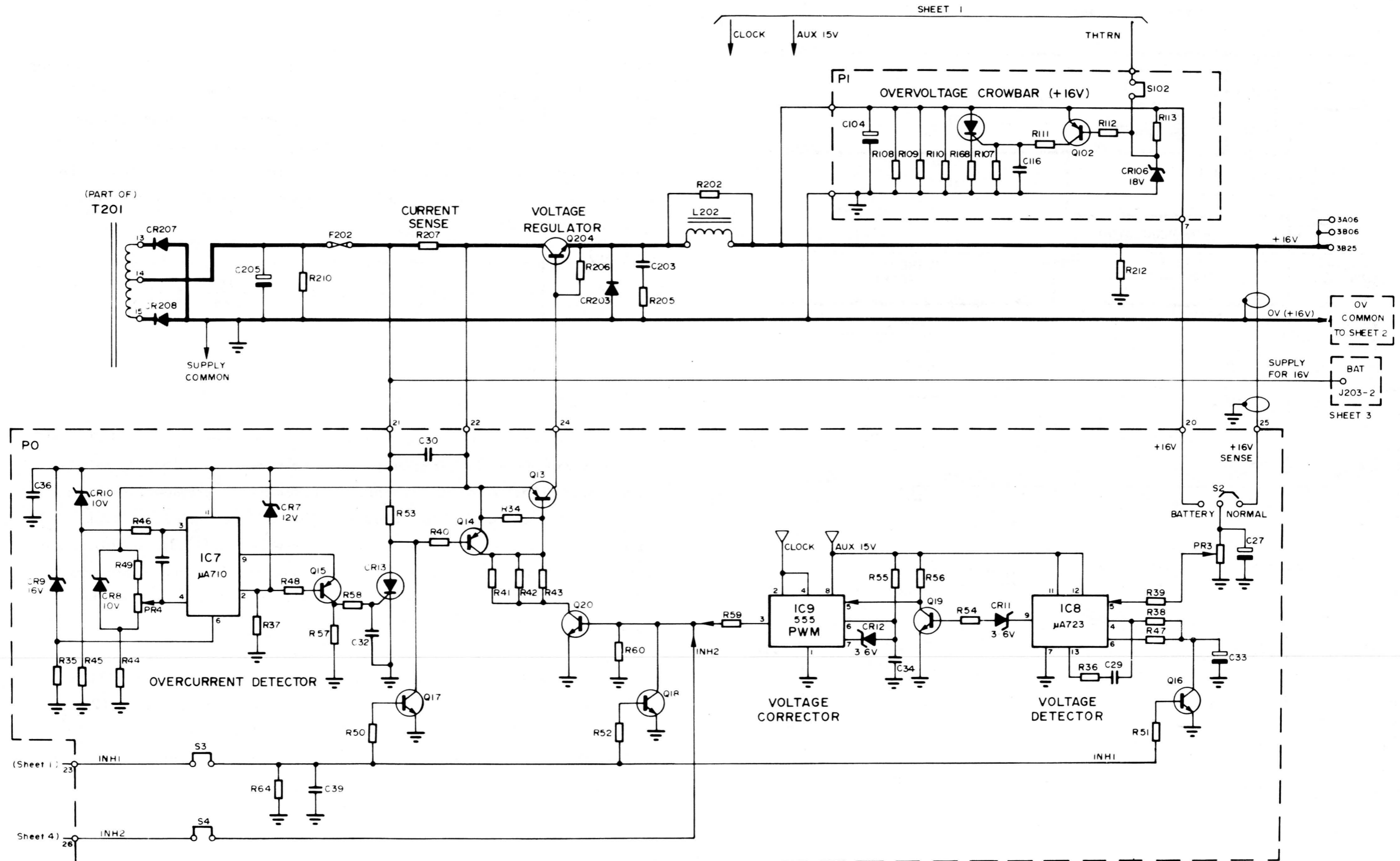


Figure 5-3 (Sheet 3) Power Supply +16V Regulator

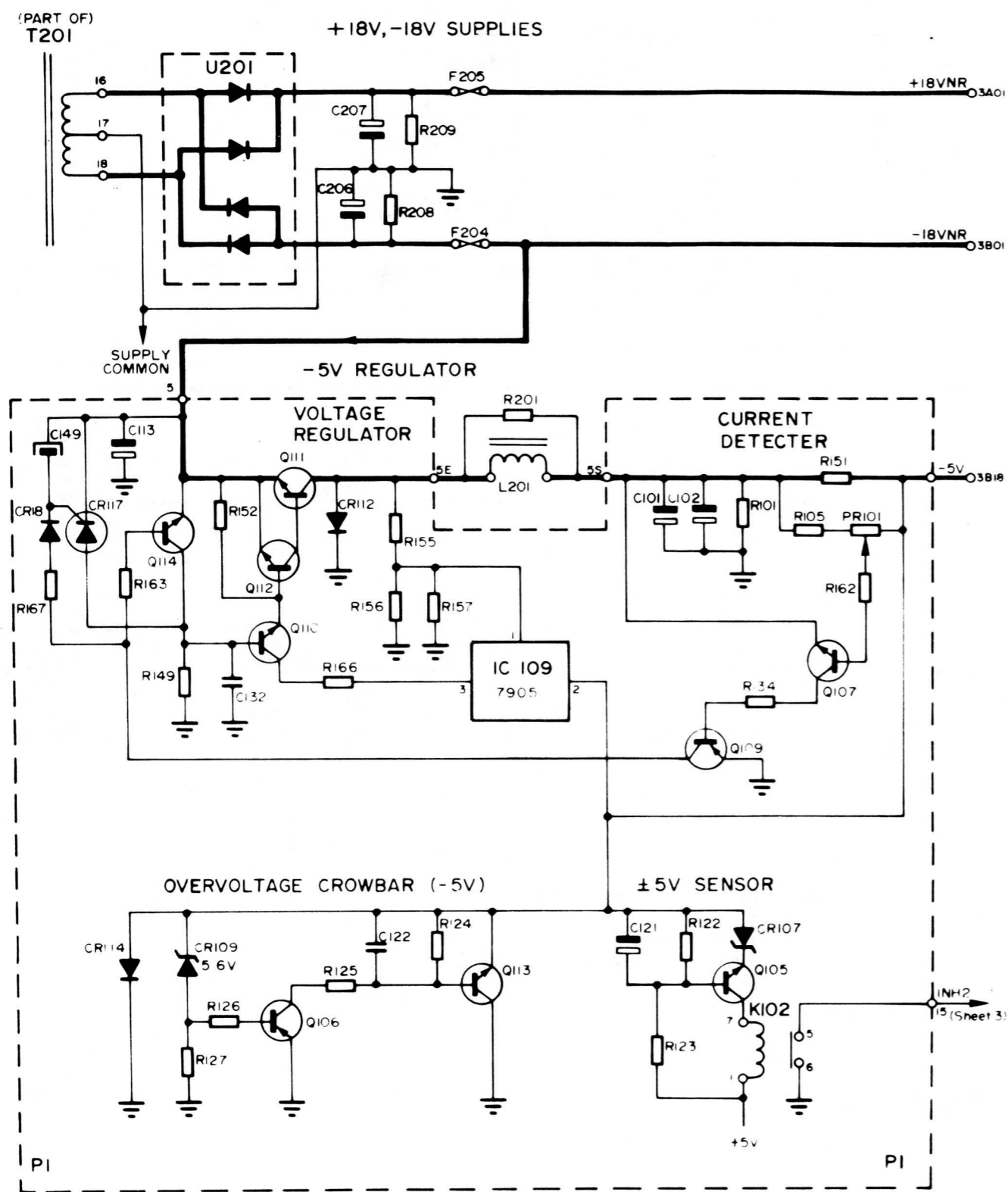
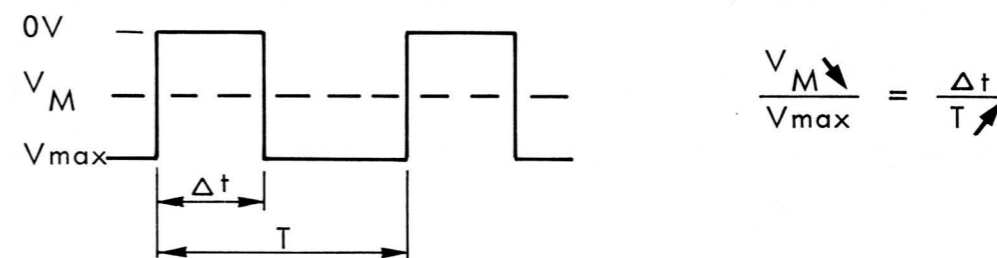


Figure 5-3 (Sheet 4) Power Supply +18V, -18V, -5V, Supplies

5.27 -5V REGULATOR

5.28 Voltage Regulation. The -5V, 2-amp power is supplied by a switching type regulator (Figure 5-3, sheet 4). Transistor Q111 is switched on and off at about 20 kHz by a type 7905 regulator (IC109) used as the voltage detector. The -5V regulator controls the mean voltage V_M (following diagram) by varying the switching frequency T in inverse relation to the required correction. The T is increased, while the on-time Δt is held constant, in order to reduce the mean output voltage.



5.29 The output voltage is compared to a reference voltage by IC109. The reference voltage at pin 1 is established by the small-value R156/157 relative to the large R155. When IC109 detects less than 5 volts between pins 2 and 1 it switches off, and Q111 is switched on. With Q111 on, diode CR112 is reverse biased and not conducting; current flows through Q111 and L201 to the load, increasing (more negative) linearly by charging the output capacitors C101/102. The capacitors continue charging until the -5V is detected by IC109.

5.30 When IC109 detects more than 5 volts, it switches on and Q111 is switched off. With Q111 off, L201 current starts to decrease and CR112 is forward biased; the current now flows through CR112, and discharging from C101/102, decreases linearly. When the sense voltage falls below the reference, IC109 switches Q111 back on and the cycle repeats.

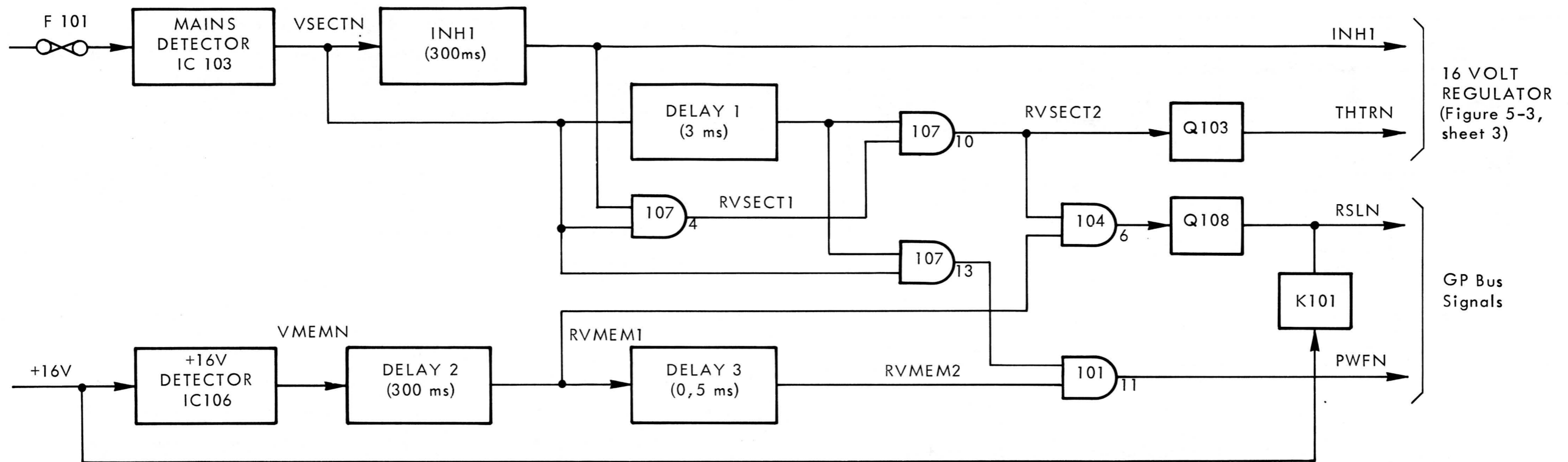


Figure 5-4 Power Sequencing Block Diagram

VMEMN (memory voltage). The leading edge of VMEMN triggers Delay-2. RVMEM1 is generated approximately 300 ms after VMEMN, when Delay-2 drops. RVSECT2 (inverted at gate 101-3) and RVMEM1 together switch off Q108 to terminate the reset-line signal RSLN.

5.38 Delay-3 is an R-C circuit (R128 and C124) which produces the RVMEM2 signal 0.5 ms after the rise of RVMEM1. RVMEM2 terminates the power failure signal PWFN, as long as VSECTN and DELAY1 are both low.

5.39 Power-Off Sequence. The Mains Detector circuit (amplifier IC103 and schmitt trigger 108-3) detects any mains failure longer than 10 ms, and indicates the failure by de-activating the VSECTN signal. The circuit uses the discharge time of C127 through PR103 and R133. The time constant is adjusted by PR103 so that the dropping power reaches the critical level at 10 ms.

5.40 The rising edge of VSECTN drops RVSECT1 without a delay, via the bypass of the INH-1 circuit. VSECTN immediately generates power-failure signal PWFN via gates 107-12/13, 101-13/11, and 104-9,10/8. The Delay-1 circuit drops 3 ms after RVSECT1 falls, and RVSECT2 goes high. RVSECT2 thus generates reset-line signal RSLN (via gates 101-1,2/3 and 104-4/6) 3 ms after PWFN.

5.41 The RVSECT2 signal switches on transistor Q103 to generate signal THTRN. This signal activates the +16V overvoltage circuit which quickly shuts off the +16V supply. The loss of the +16 volts also drops out relay K101; the closed contacts then ground RSLN, thus holding the signal in its active state while power is off. Since transistor Q108 operates after K101 for power on and before K101 for power off, contact bounce is masked from the reset line.

5.42 The +5V supply begins to drop after the +16V supply is off. The fall time of the +5V supply is about 20 ms with minimum load.

5.43 MECHANICAL

The power-supply chassis is mounted on a base plate in the basic mounting box (Figure 5-6). The power supply output voltages are connected to the system circuit cards via back-panel connector 3. The main power-supply assemblies are shown in Figure 5-7. Most of the electronic circuits are located on two printed-circuit cards, P0 and P1 (Figure 5-8). Power transistors and rectifiers and the fuses are mounted on the heat-sink assembly (Figure 5-9). A few very large components (coils and capacitors) are mounted directly on the chassis base plate. The mains power transformer (T201) and the two fans are mounted on the basic box at the rear. The key switch is mounted on the front of the basic box (Figures 5-6, 5-9). All components not mounted on the two circuit cards have the reference designations 2__.

5.44 Top Cover Removal

Figures 5-6 and 5-7 show the power supply with the top cover removed. The cover is attached by five screws : two on each side and one in the back, at the corner near the fan.

5.45 Power-Supply Chassis Removal

The power -supply chassis (except mains transformer and fans) can be removed on its base plate from the basic mounting box. (Refer to Figure 5-7.)

Disconnect :

- connections at connector 3 (five small plugs, not the two bigger ones; main ground, at connector-3 end).
- RTC connector.
- TR201 outputs 10 through 21.
- Circuit card P0, for access to remote-key connections :
 - a. connector plug off.
 - b. Remove four screws and spacers onto P1 (one at each corner of P0). A ground wire from the remote key is attached by one of the corner screws.

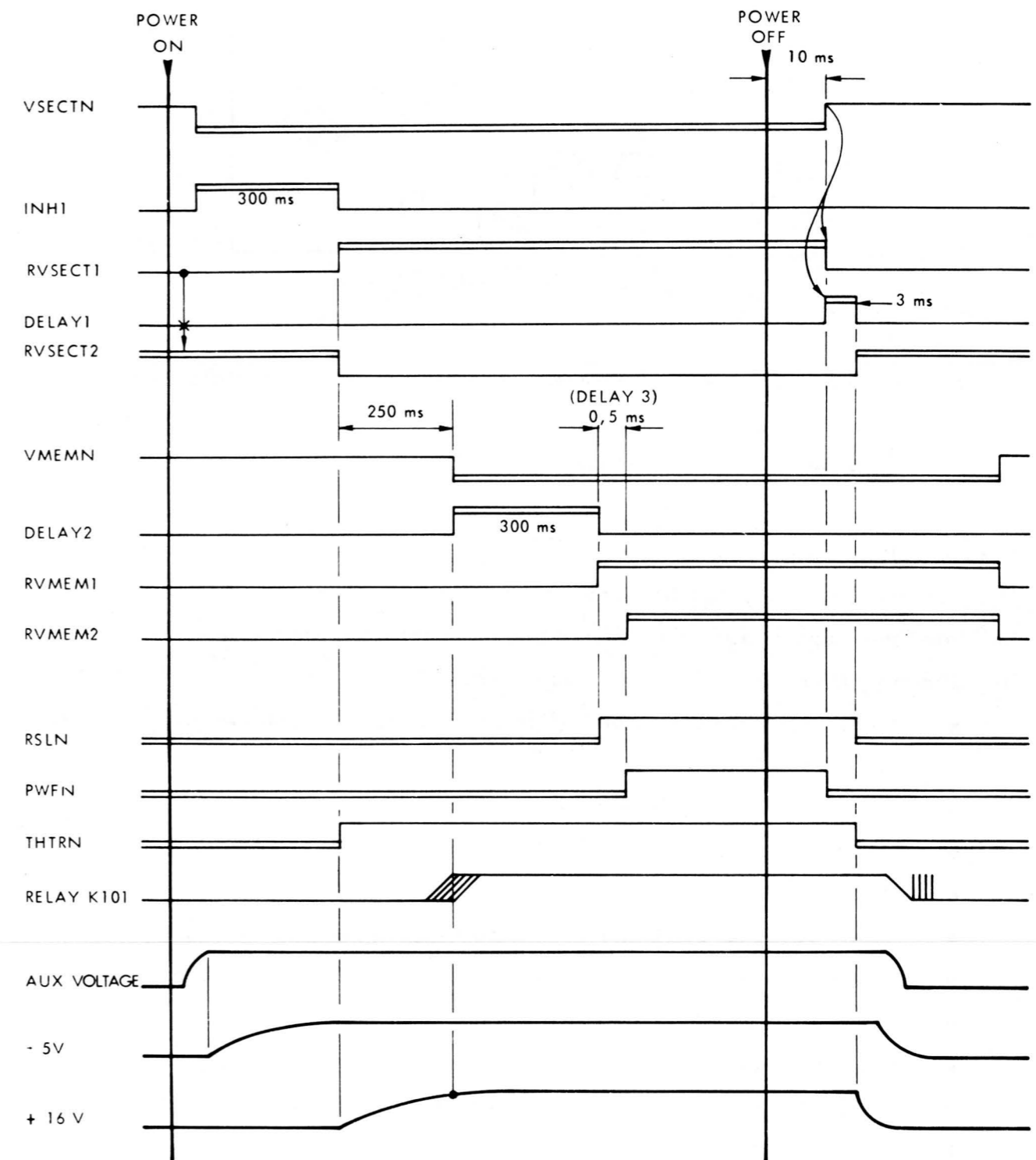
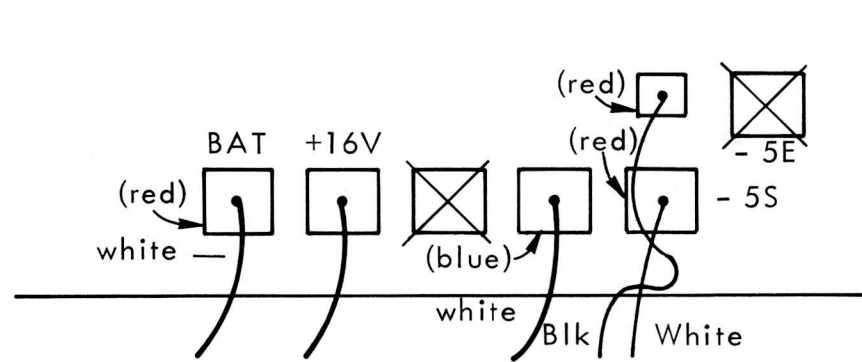


Figure 5-5 Power Sequence Timing

- remote-key connections A, B, C.
- P1 power connections (five plugs) as follows :



- Six screws, three on each side, directly through base plate -- not those through any other brackets mounted on the base plate (however, one does hold a plastic cable clamp).

5.46 Heat-Sink Assembly Removal

The heat-sink assembly can be removed from the basic mounting box. The * indicates the steps already done if the power-supply chassis has been removed from the mounting box. (Refer to Figure 5-7.) Disconnect :

- * • connections at connector 3 (five small plugs, not the two bigger ones; main ground, at heat-sink end).
- connector plugs at P0 and P1.
- * • RTC connector.
- * • TR201 outputs 10 through 21.
- three cables to C204 and C205 (the two biggest capacitors).
- Unsolder three wires to C206 and C207 (the two smaller capacitors).
- +5V cable to L203.
- +5V power cable (through L204) at both ends.
- ground connection between heat sink and base plate.
- four mounting screws, each through a plastic foot bracket, from the base plate.

5.47 Circuit-Card P0 Removal

The P0 circuit card must be removed before the power-supply chassis is removed or the P1 circuit card is removed. P0 can also be removed alone from the basic mounting box. (Refer to Figure 5-7.) Disconnect :

- connector plug.
- four screws and spacers onto P1 (one at each corner of P0). A ground wire from the remote key is attached by one of the corner screws.

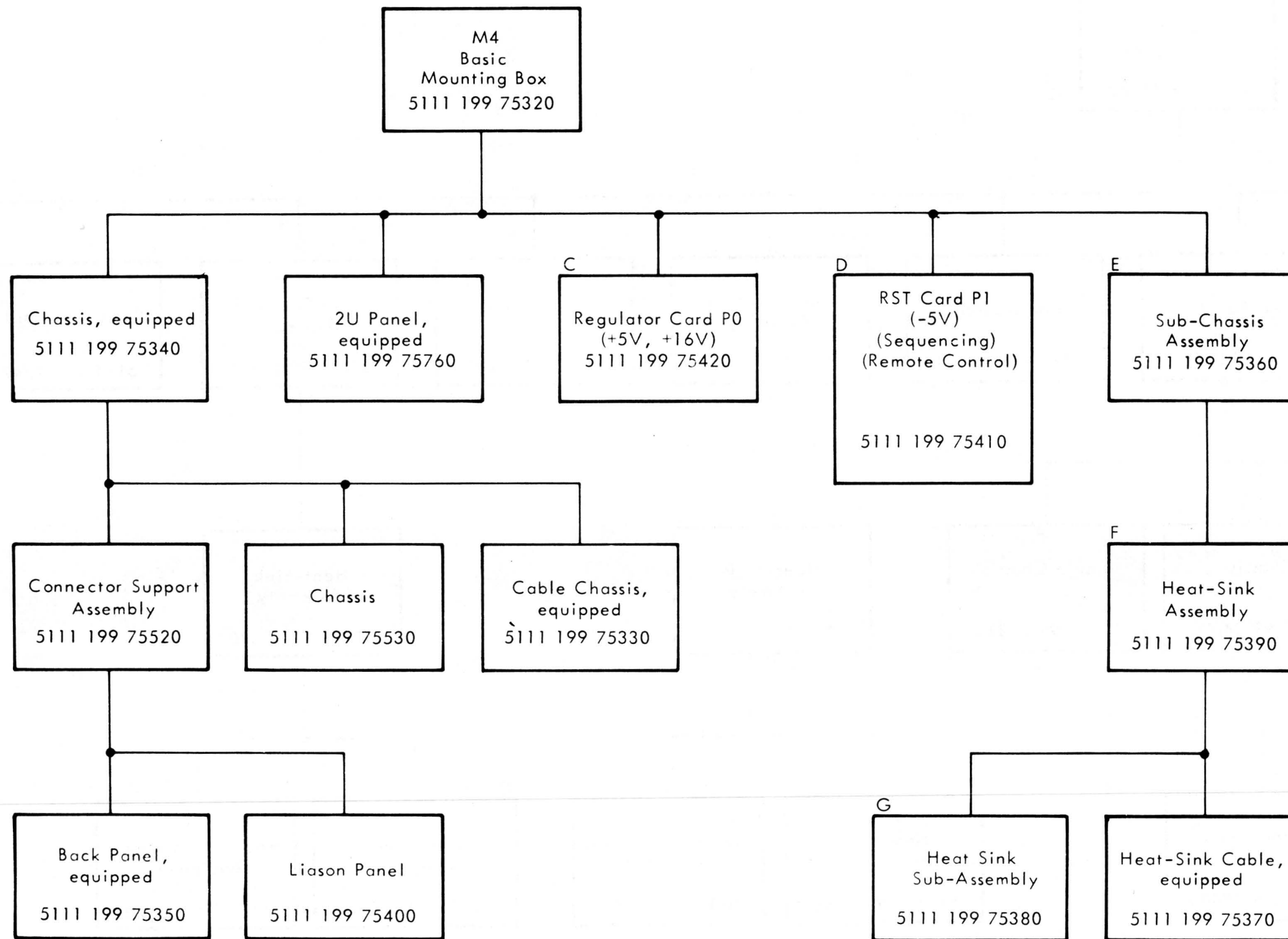
5.48 Circuit-Card P1 Removal

The P1 circuit card can be removed from the power-supply chassis. The * indicates the steps already done if the chassis has been removed from the mounting box. (Refer to Figure 5-7.) Disconnect :

- * • and remove card P0 (previous paragraph).
- * • remote-key connections A, B, C.
- all eight power connections at connector 3.
- the +5V cables to L203 and L204 (from the same point on the circuit card).
- ground cable to the heat sink.

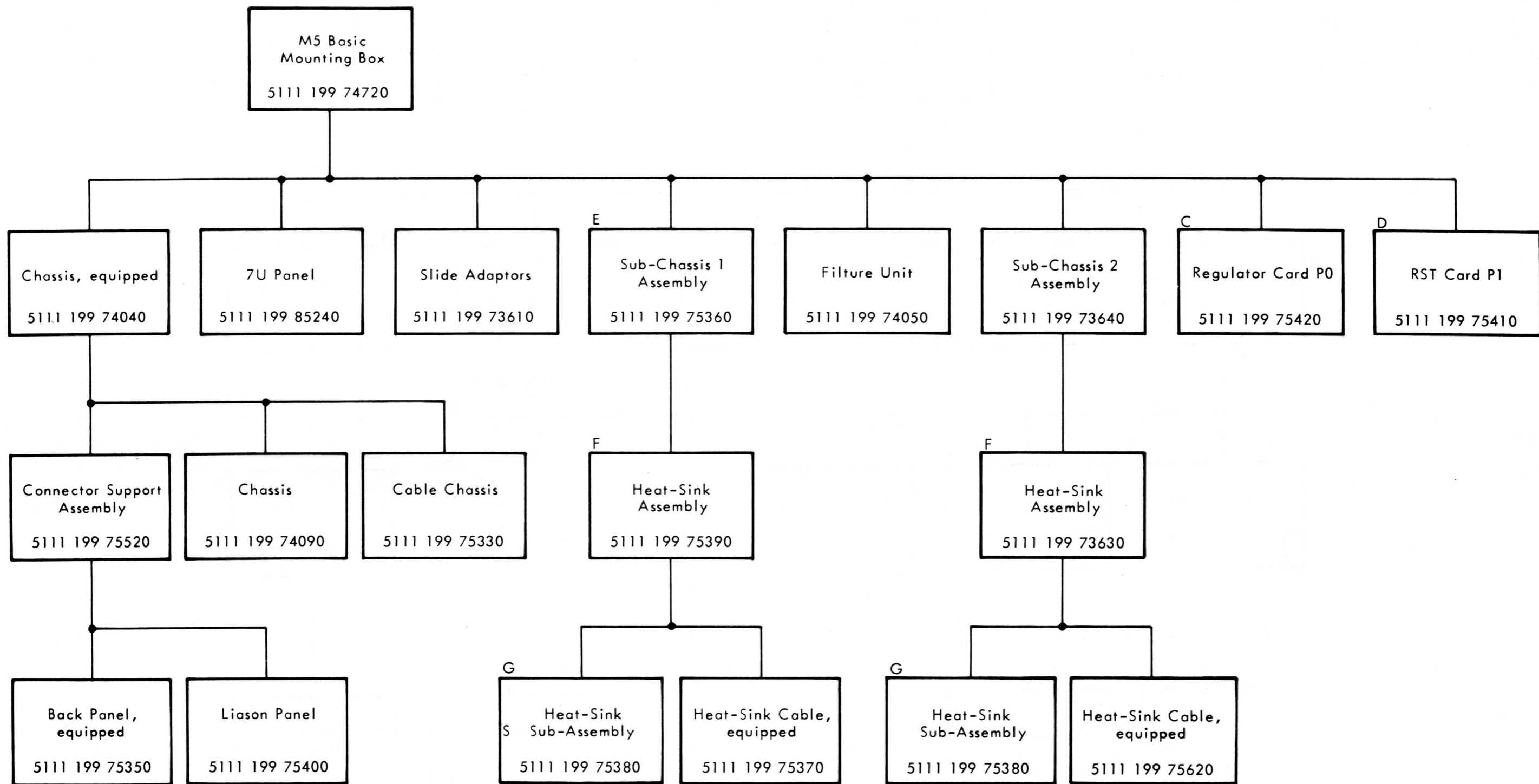
5.49 List of Components

All power-supply components are listed in the following parts-list, Table 5-1.



C to G Parts Lists Included

Table 5-1A M4 Parts List Guide



C to G = Parts Lists Included

Table 5-1B M5 Parts List Guide

Table 5-1C Regulator Card P0 Parts List

Reference	Description	12NC Code
IC1,2,3,9.	Printed Circuit	5111 100 05843
IC7.	Integrated Circuit NE 555 V.	
IC4,6,8.	Integrated Circuit μ A 710 DC.	
IC5.	Integrated Circuit UGA 7723 393.	
	Voltage Regulator 7815 (TO22O).	
Q 12 .	Transistor 2N 3055 (TO3)	
Q2,4.	Transistor BDX 77 (TO22O)	
Q 1,3,7,8,11,17,20.	Transistor 2N2219	
Q 9,14,15.	Transistor 2N2905	
Q 13.	Transistor BDX78 (TO22O)	
Q 5,6,10,16,18,19,21,22.	Transistor BSX20	
CR4.	Diode BZX79 C4 V7.	
CR8,10,14,15.	Diode BZX79 C10.	
CR9.	Diode BZX79 C16.	
CR7.	Diode BZX79 C12.	
CR2,3,11,12.	Diode BZX75 C3 V6.	
CR5,16.	Diode BAX 12.	
CR1.	Diode BYX49-300.	
CR6.	Diode BZX79 C5 V6.	
CR13.	Thyristor 2N2323.	
P2,3.	Potentiometer 2600 P102.	
P4.	Potentiometer 2600 P203.	
P1.	Potentiometer 2600 P502.	
R11,38,39,47.	Resistor 1K Ω , 0.25W, 1%.	
R3.	Resistor 5.62K Ω , 0.25W, 1%.	
R13.	Resistor 215 Ω , 0.25W, 1%.	
R49.	Resistor 68K Ω , 0.25W, 5%.	
R46.	Resistor 10K Ω , 0.25W, 1%.	
R4,8,12,19,21,24,25,26,29,30,32,33,36,50,51,52,56.	Resistor 1K Ω , 0.25W, 5%.	
R15.	Resistor 200 Ω , 0.25W, 5%.	
R14.	Resistor 1.2K Ω , 0.25W, 5%.	
R16,31,48,54,58.	Resistor 2K Ω , 0.25W, 5%.	
R9.	Resistor 5.1K Ω , 0.25W, 5%.	
R10.	Resistor 510 Ω , 0.25W, 5%.	
R60,62.	Resistor 100 Ω , 0.25W, 5%.	
R57.	Resistor 4.7K Ω , 0.25W, 5%.	
R40.	Resistor 8.2K Ω , 0.25W, 5%.	
R59,61.	Resistor 390 Ω , 0.25W, 5%.	
R20.	Resistor 470 Ω , 0.25W, 5%.	

Table 5-1C Contd.

Reference	Description	12NC Code
R22,27,28.	Resistor 10K Ω , 0.25W, 5%.	
R18.	Resistor 150 Ω , 0.25W, 5%.	
R17,23.	Resistor 1.5K Ω , 0.25W, 5%.	
R35,37,44,45.	Resistor 3.83K Ω , 0.5W, 1%.	
R1.	Resistor 200 Ω , 0.5W, 5%.	
R34.	Resistor 20 Ω , 0.5W, 5%.	
R53.	Resistor 3K Ω , 0.5W, 5%.	
R2.	Resistor 47 Ω , 10%, WRO617E.	
R41,42,43.	Resistor 1.5K Ω , 5%, WRO617E.	
R5,6,7.	Resistor 33 Ω , 5%, WRO825E.	
R63.	Resistor 620 Ω , 0.25W, 5%.	
R64.	Resistor 3K Ω , 0.25W, \pm 5%.	
R55.	Resistor 1.47K, 0.25W, 1%.	
R65.	Resistor 680 Ω , 0.25W, \pm 1%.	
R66.	Resistor 10K Ω , 0.25W, \pm 1%.	
C4,8.	Capacitor 0.01 μ F, 125V, 1%.	
C3.	Capacitor 1500 μ F, 10V.	
C2,6,9,11,20,25,26,28,31,39.	Capacitor 10nF, ceramic	
C19.	Capacitor 100pF, ceramic	
C5,13,16,30.	Capacitor 0.1 μ F, 100V, 10%, MPR	
C1,10.	Capacitor 0.01 μ F, 250V, 10%, MPR	
C7,17,18,24,27.	Capacitor 15 μ F, 20V, CTS13.	
C12,15,C22.	Capacitor 33 μ F, 10V, CTS13.	
C23.	Capacitor 3.3 μ F, 16V, CTS13.	
C14.	Capacitor 220 μ F, 25V, FITCO.	
C33.	Capacitor 100 μ F, 10V, FITCO.	
C32,34.	Capacitor 0.022 μ F, 250V, MPR	
C37.	Capacitor 470pF, 10%, ceramic	
C29.	Capacitor 0.47 μ F, 100V, 10%, MPR.	
C38.	Capacitor 3900pF, 100V, \pm 10%, ceramic	
C35.	Capacitor 10 μ F, 25V, FITCO.	
	Heat Sink PBI-2U.	
	Mica insulator 56325.	
T 1.	Transformer TRI M4.	

Table 5-1D RST Card P1 Parts List

Reference	Description	12NC Code
	Printed Circuit	5111 100 05854
IC103,106.	Integrated Circuit UGA 7723 393.	
IC101,108.	Integrated Circuit 74132.	
IC102,105.	Integrated Circuit 9602.	
IC107.	Integrated Circuit 7402.	
IC104.	Integrated Circuit 1801.	
IC109.	Regulator 7905 (TO22O).	
Q 111,113.	Transistor BDX77 (TO22O).	
Q 103,112.	Transistor 2N2219.	
Q 109.	Transistor 2N2905.	
Q 104,107.	Transistor BSX20.	
Q 105,108,110,114.	Transistor BSX60.	
Q 101,102,106.	Transistor 2N2906.	
CR 107,116.	Diode BZX79 C4 V7.	
CR 102,109.	Diode BZX79 C5 V6.	
CR 106.	Diode BZX79 C18.	
CR 114.	Diode 1N4005.	
CR 104,105,110,111,118.	Diode BAX12.	
CR 112.	Diode MR820.	
CR 101,103.	Thyristor BTW92/600RM.	
CR 117.	Thyristor 2N2323.	
CR 113.	Triac TXAL 615 M.	
P 101,103.	Potentiometer 2600 P 102.	
P 102.	Potentiometer 2600 P 502.	
R 105,127.	Resistor 46.4n, 0.25W, 1%.	
R 113.	Resistor 121n, 0.25W, 1%.	
R 134.	Resistor 19.6Kn, 0.25W, 1%.	
R 117.	Resistor 5.11Kn, 0.25W, 1%.	
R 118,119.	Resistor 4.22Kn, 0.25W, 1%.	
R 130.	Resistor 31.6Kn, 0.25W, 1%.	
R 115,116,135,137,143,144,167.	Resistor 1 Kn, 0.25W, 5%.	
R 123.	Resistor 270n, 0.25W, 5%.	
R 129,133,150.	Resistor 510n, 0.25W, 5%.	
R 103,111,112,126,128,131,142,153,154,160,161,166.	Resistor 100n, 0.25W, 5%.	
R 124.	Resistor 300n, 0.25W, 5%.	
R 158.	Resistor 51n, 0.25W, 5%.	
R 132,141.	Resistor 330n, 0.25W, 5%.	
R 136,146.	Resistor 30Kn, 0.25W, 5%.	

Table 5-1D Contd.

Reference	Description	12NC Code
R 159.	Resistor 470n, 0.25W, 5%.	
R 156,157.	Resistor 1n, 0.25W, 5%.	
R 122.	Resistor 680n, 0.25W, 5%.	
R 125.	Resistor 33n, 0.25W, 5%.	
R 145.	Resistor 120n, 0.25W, 5%.	
R 138,147.	Resistor 2.4Kn, 0.25W, 5%.	
R 101,106.	Resistor 100n, 0.5W, 5%.	
R 104,152,192.	Resistor 10n, 0.5W, 5%.	
R 149.	Resistor 1Kn, 0.5W, 5%.	
R 114.	Resistor 200n, 0.5W, 5%.	
R 102,107.	Resistor 300n, 0.5W, 5%.	
R 120.	Resistor 510n, 0.5W, 5%.	
R 121.	Resistor 240n, 0.5W, 5%.	
R 108,109,110.	Resistor WRO617E 560	
R 151,168.	Resistor 0.3n, 5%, 224E.	
R 139.	Resistor 10Kn, 0.25W, 5%.	
R 164.	Resistor 2Kn, 0.25W, ± 5%.	
R 165.	Resistor 390n, 0.125W, 5%.	
R 162.	Resistor 6.8Kn, 0.25W, 5%.	
R 140.	Resistor 200n, 0.25W, 5%.	
R 163.	Resistor 20Kn, 0.25W, 5%.	
R 155.	Adjustable Resistor 510n to 2 Kn, 0.5W, 5%, E 24.	
C 123,126,128,130,134,136,138,141,144,190.	Capacitor 10nF, ceramic	
C 131,139.	Capacitor 560pF, ceramic	
C 142,143.	Capacitor 1000pF, ceramic	
C 129.	Capacitor 56pF, ceramic	
C 115,116,120,122.	Capacitor 0.1μF, 100V, 10%, MPR	
C 119,132.	Capacitor 0.01μF, 250V, 10%, MPR	
C 135.	Capacitor 0.47μF, 100V, 10%, MPR	
C 117,125,137.	Capacitor 33μF, 10V, CTS13	
C 124,127,140.	Capacitor 3.3μF, 16V, CTS13	
C 118,121,133,145,149.	Capacitor 22μF, 25V, FITCO	
C 146.	Capacitor 47000pF, 400V, 20%, PMA	
C 147.	Capacitor 22000pF, 400V, 20%, PMA	
C 148.	Capacitor 0.1μF, 630V, 20%, PMA	
C 104.	Capacitor 680μF, 25V.	
C 105-114.	Capacitor 1500μF, 10V.	
C 101,102.	Capacitor 100μF, 10V, FITCO	

Table 5-1D Contd.

Reference	Description	12NC Code
K 101. K 102. F 101.	Relay MRMD 15006. Relay MRMD 15005. Fuse D1/0.1. Mica insulator 56325 Triac Heat-Sink	5111 100 22541.

Table 5-1F Heat Sink Assembly Parts List

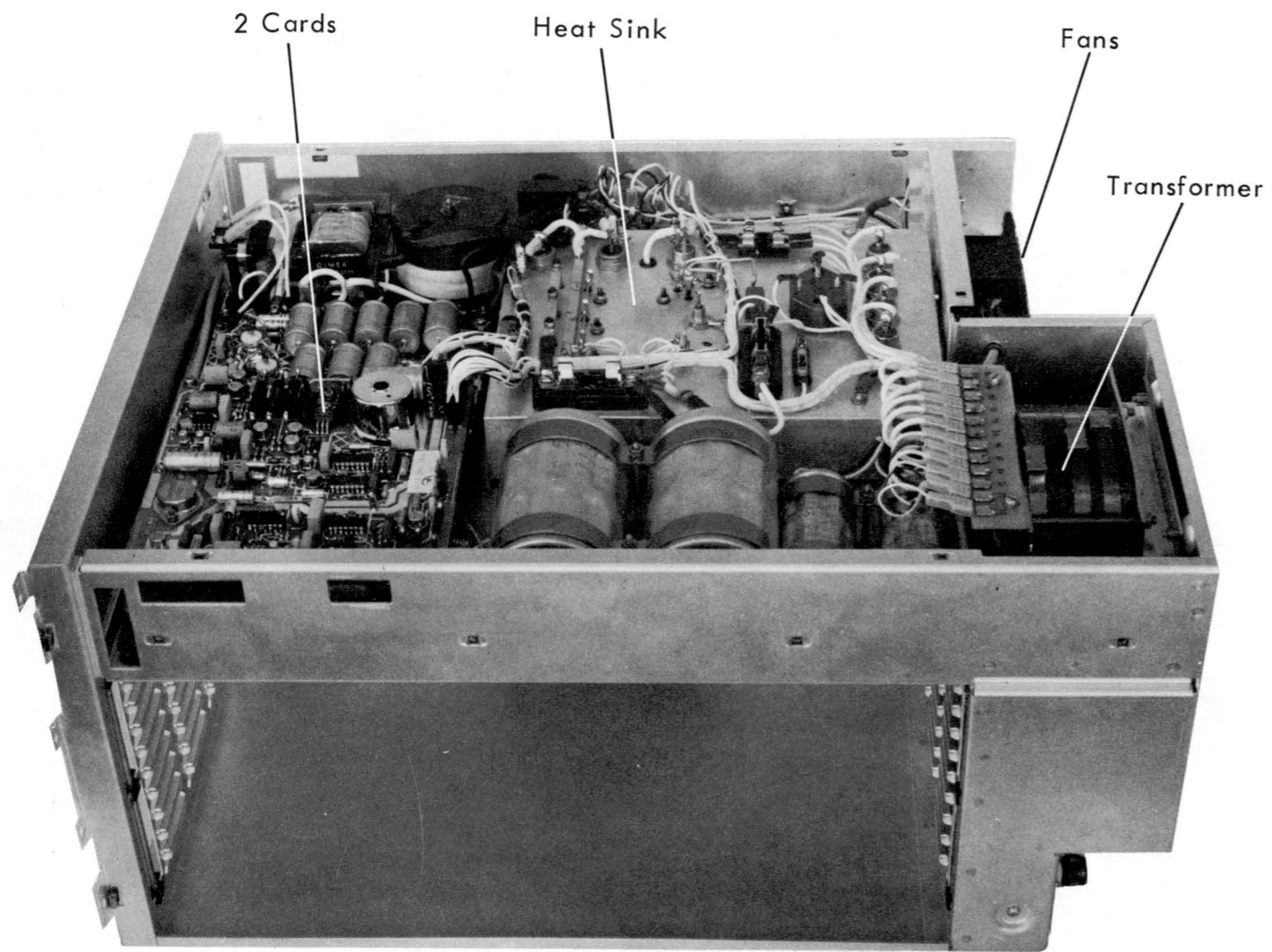
Reference	Description	12NC Code
	Heat-Sink Sub-Assembly	5111 199 75380
	Harness (M4, M5).	5111 199 75370
	Harness (M5).	5111 199 73630
R203, 204, 205 .	Resistor 10 Ω , 0.5W, 5 $^{\circ}$.	
R206.	Resistor 27 Ω , 0.5W, 5 $^{\circ}$.	
C201, 202, 203.	Capacitor 0.022 μ F, 250V, 10 $^{\circ}$, PMA.	
C210, 211.	Capacitor 10 μ F, 100V, 20 $^{\circ}$, PMA.	
F201.	Fuse A13/20.	
F202.	Fuse D8/10.	
F203.	Fuse D1/2.	
F204, 205.	Fuse D1/3.15.	

Table 5-1E Sub-Chassis Assembly Parts List

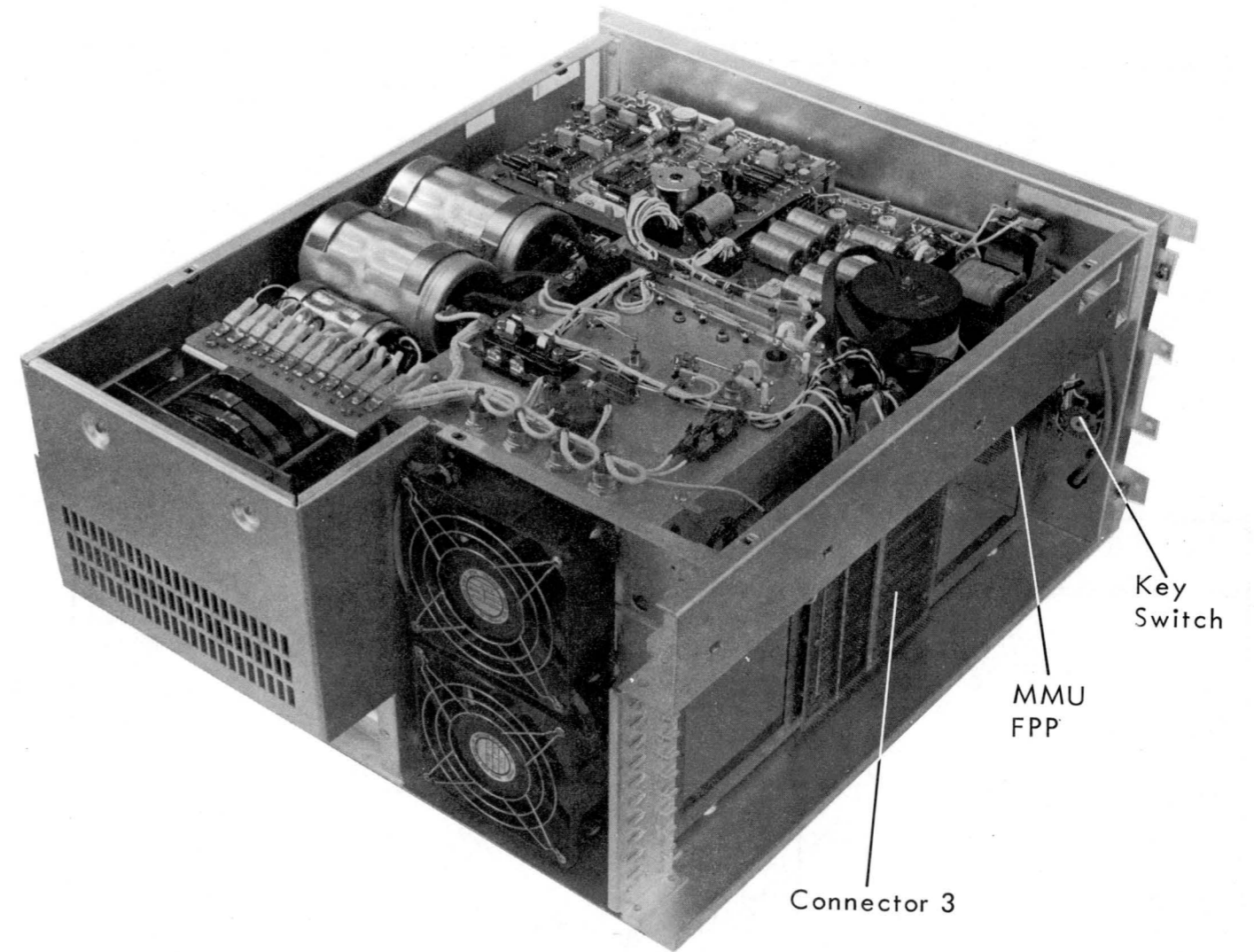
Reference	Description	12NC Code
	Sub-Chassis	5111 100 22312
	Heat-Sink Assembly	5111 199 75390
L201.	Inductance SLF 2091	
L202.	Inductance SLF 1712	
L203.	Inductance SLF 2101	
C204-C205.	Capacitor 33000 μ F, 40V.	
C206-C207.	Capacitor 5000+5000 μ F, 40V.	
R 208-209.	Resistor 1K Ω , WR0617E.	
R210-211.	Resistor 3.9K Ω , 0.5W, 5 $^{\circ}$.	
R202.	Resistor 560 Ω , 0.5W, 5 $^{\circ}$.	
R212.	Resistor 30 Ω , 25 W, \pm 3 $^{\circ}$.	
R213.	Adjustable Resistor 22 Ω to 3K Ω , 0.5W, \pm 5 $^{\circ}$.	

Table 5-1G Heat-Sink Sub-Assembly Parts List

Reference	Description	12NC Code
	Heat Sink	5111 100 22523
L204.	Inductance SLF 2541.	
IC 201.	Regulator 7815 (T03).	
Q 201, 202, 203, 204.	Transistor 2N5685.	
CR201, 202.	Diode RPR 1040 R.	
CR203.	Diode 1N3910 R.	
CR204.	Diode BZY93 C9V1.	
CR205, 206, 207, 208.	Diode BYX52 300R.	
U 201.	Bridge MDA 952-2.	
R 207.	Resistor 0.1 Ω , 3 $^{\circ}$, (RH10).	
	Relay REVIC ID.	



TOP/CARD SIDE



REAR/BACK-PANEL SIDE

Figure 5-6 Basic Mounting Box

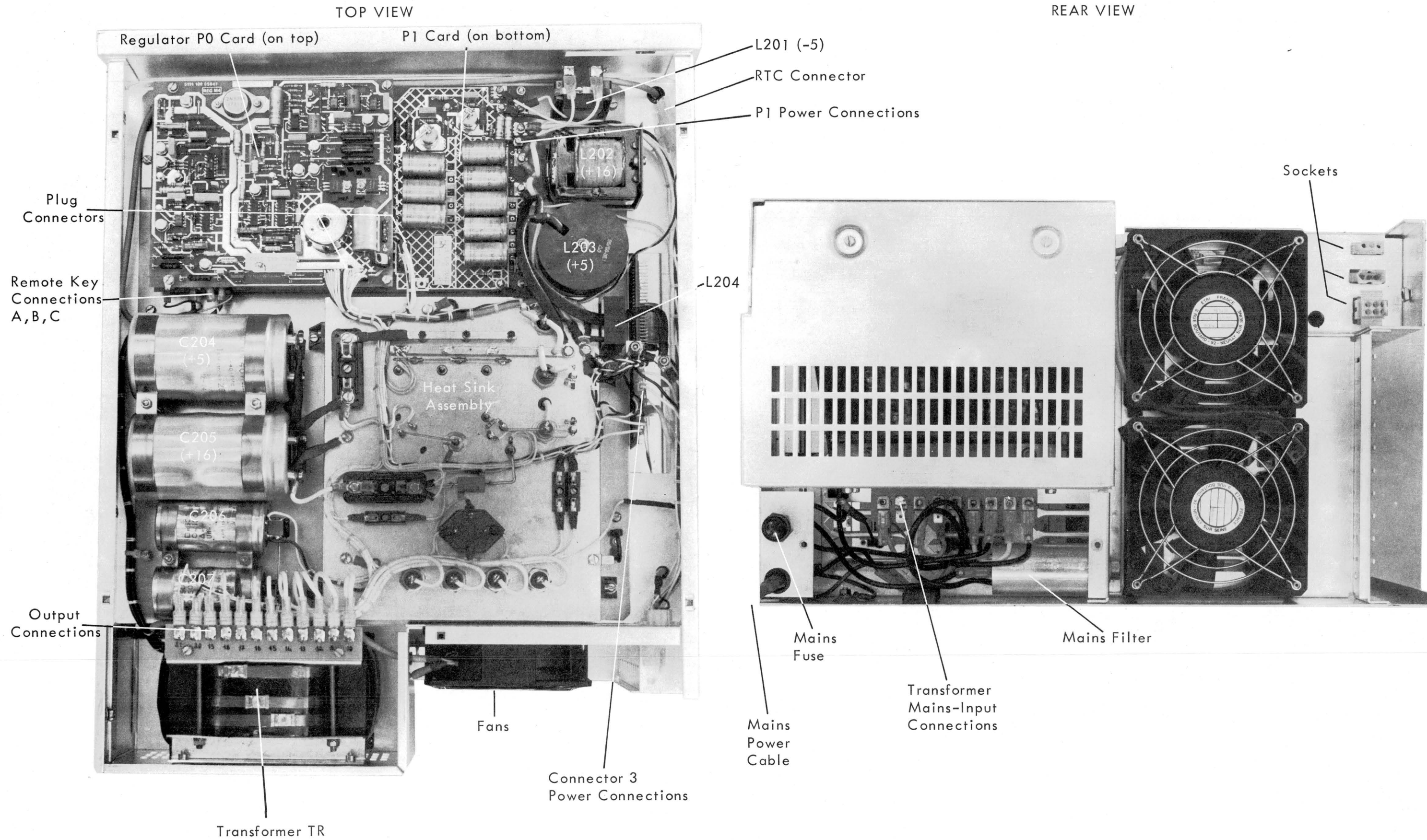
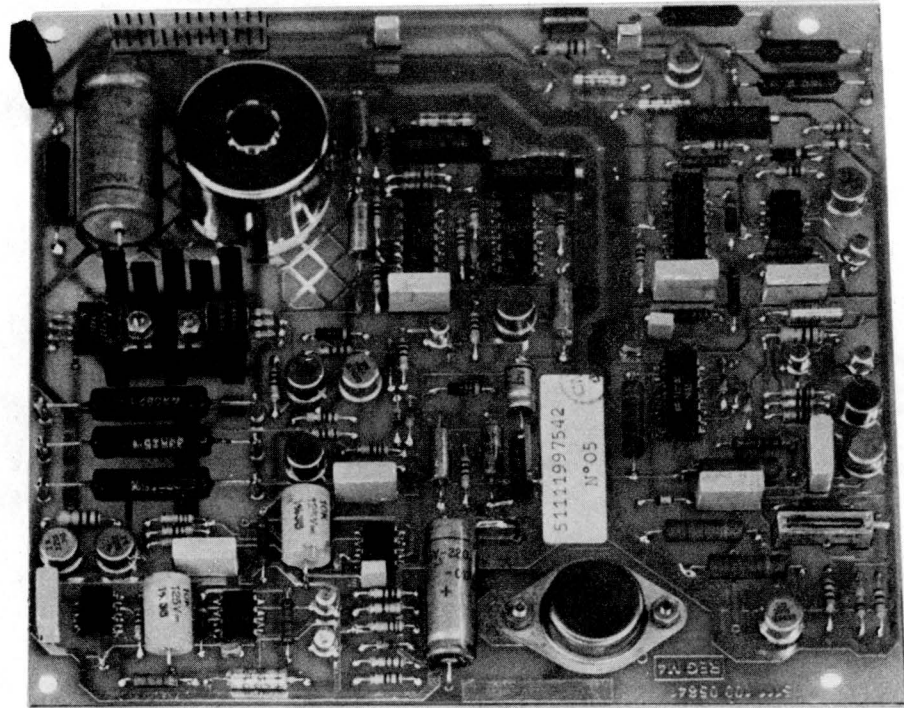
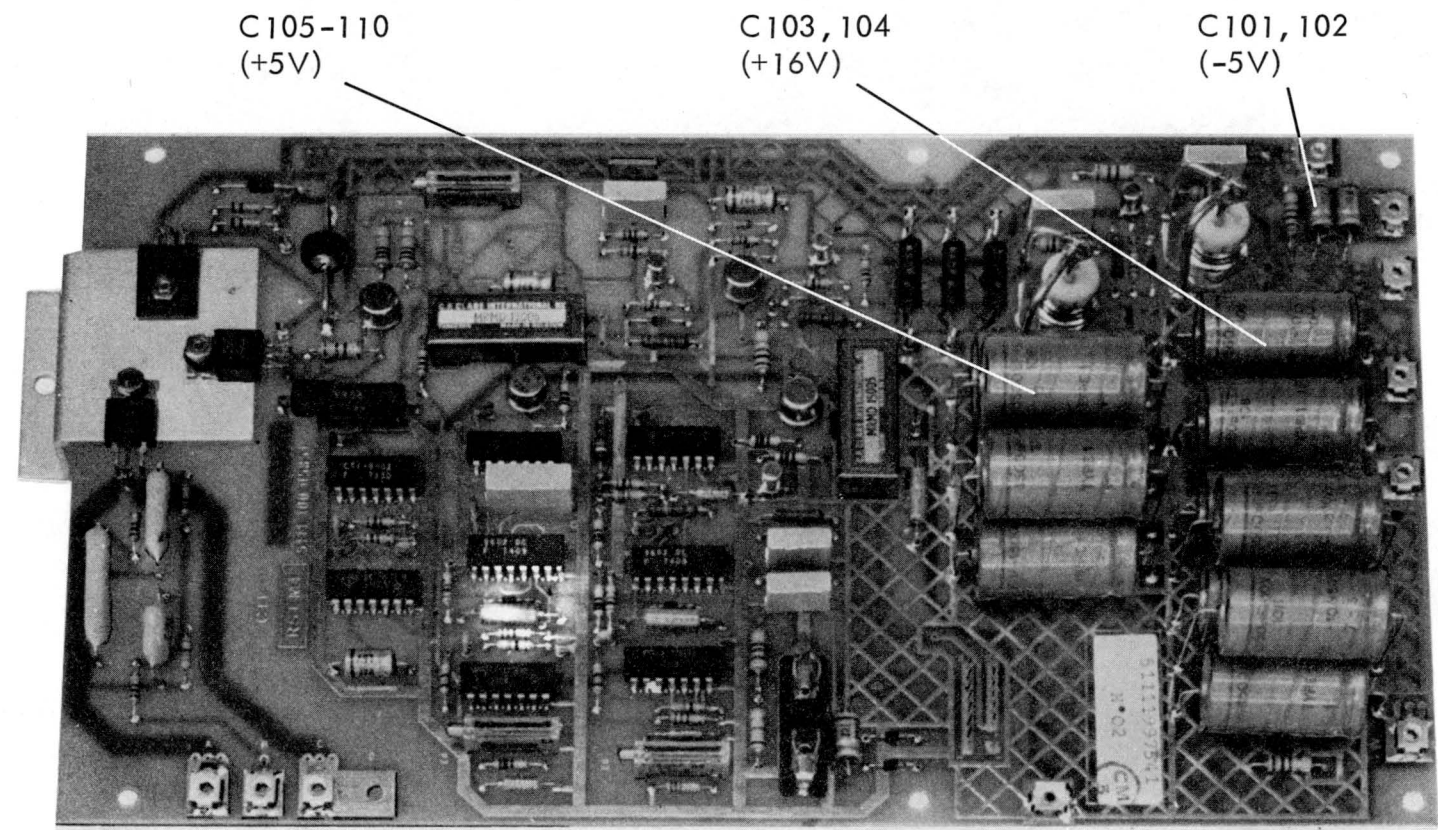


Figure 5-7 Power Supply Assembly Locations



REG CARD



RST CARD

Figure 5-8 Power Supply Circuit Cards

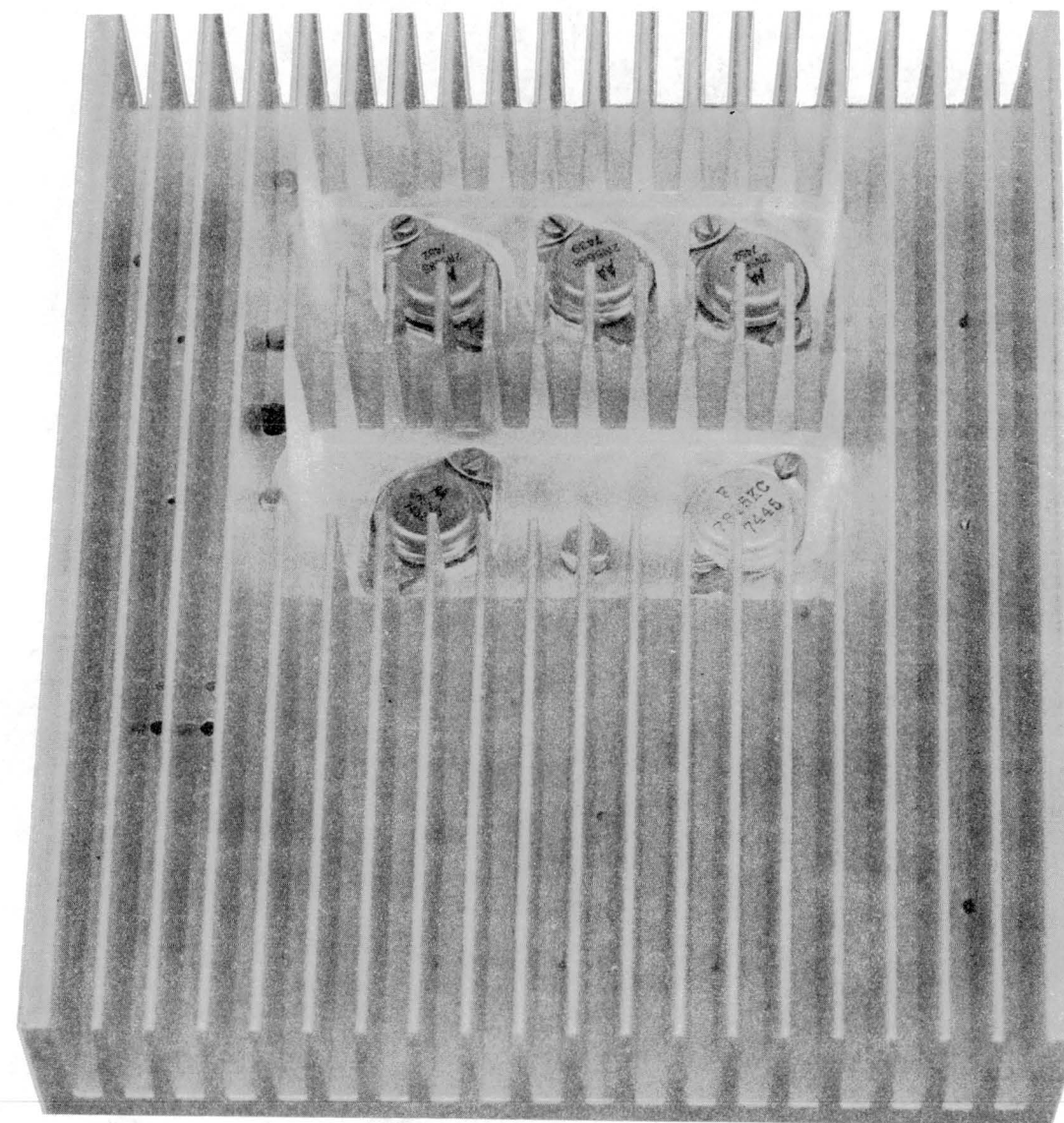
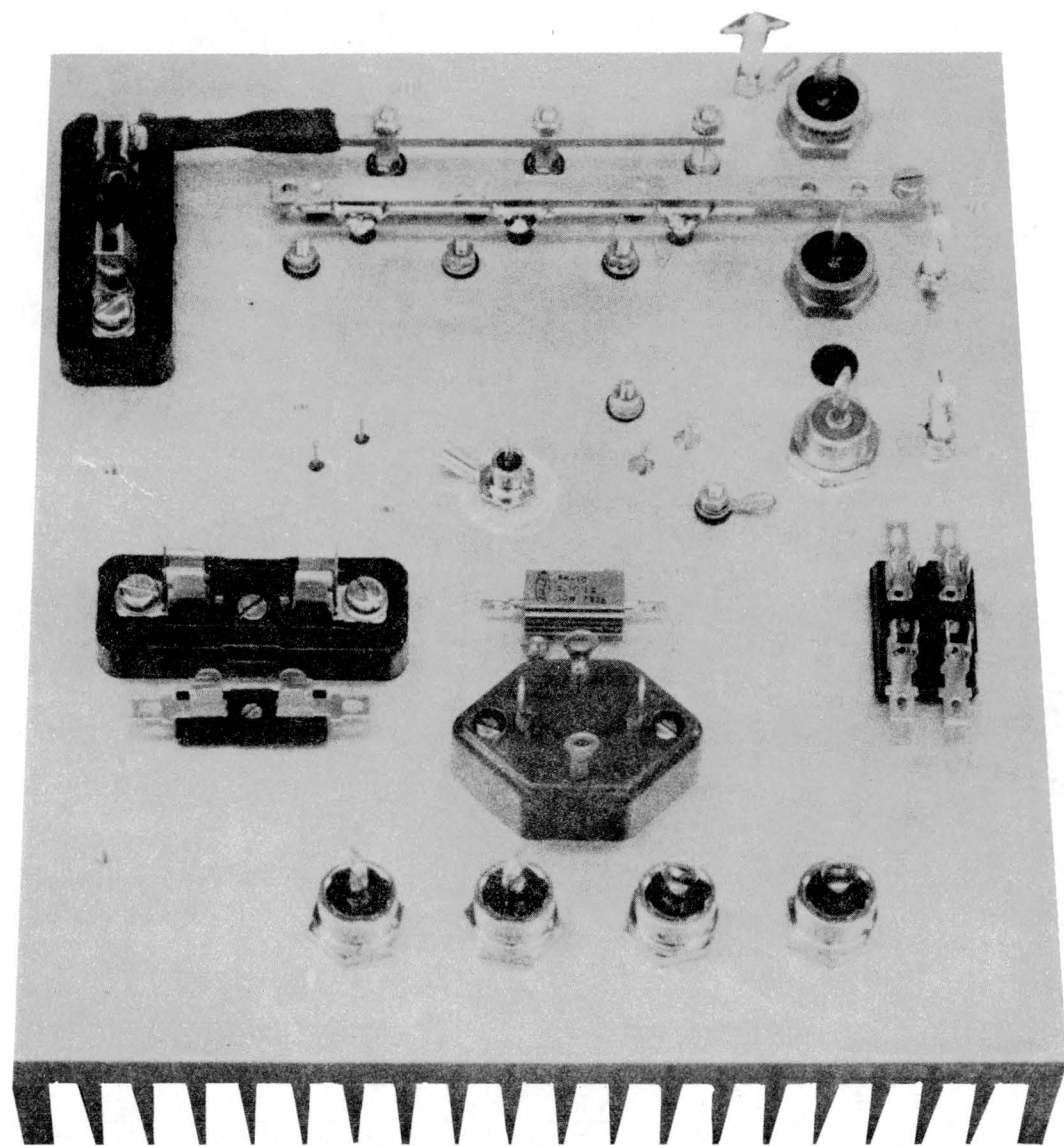
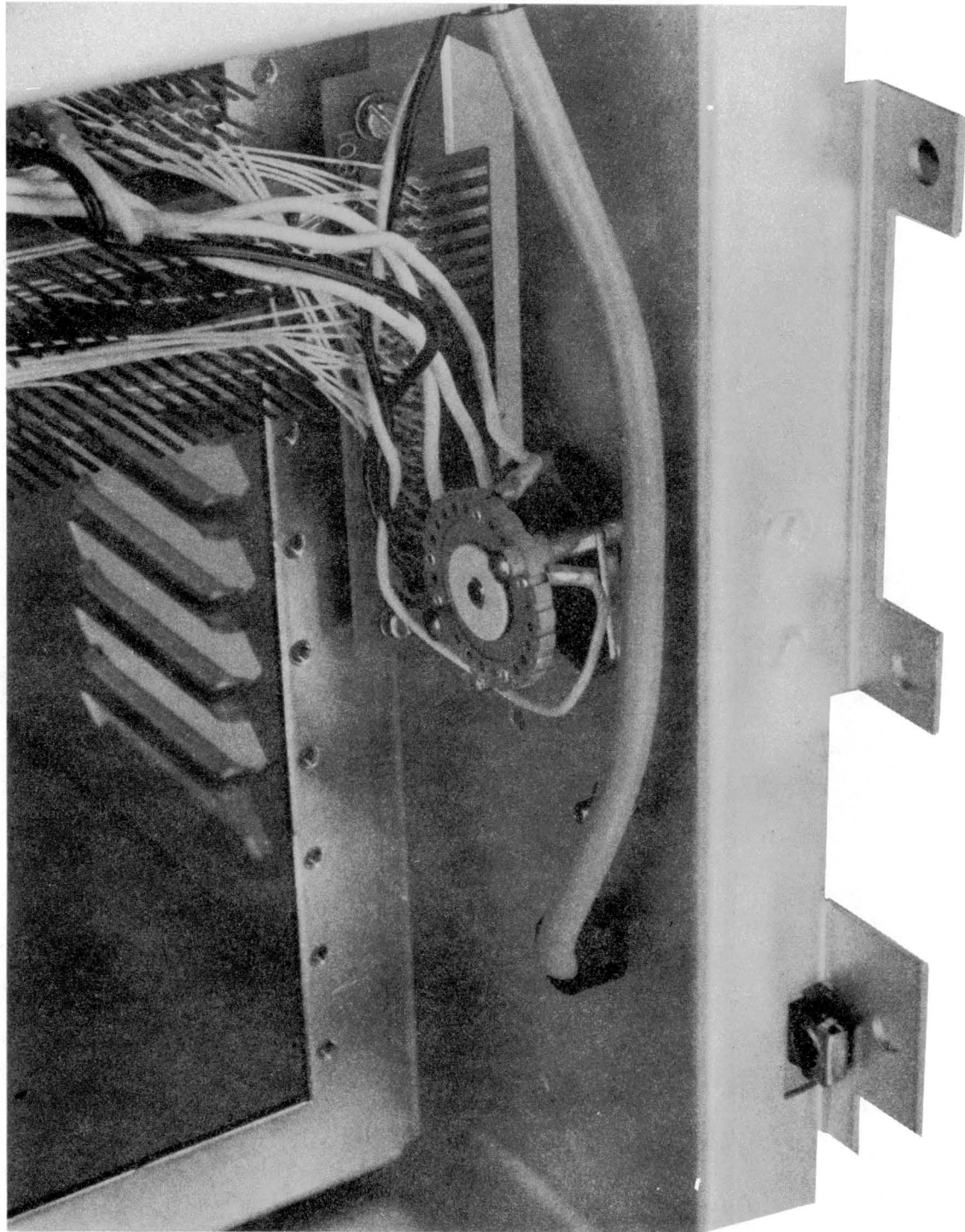


Figure 5-9 Heat-Sink Assembly

FRONT VIEW
(Control Panel Removed)



REAR/SIDE VIEW

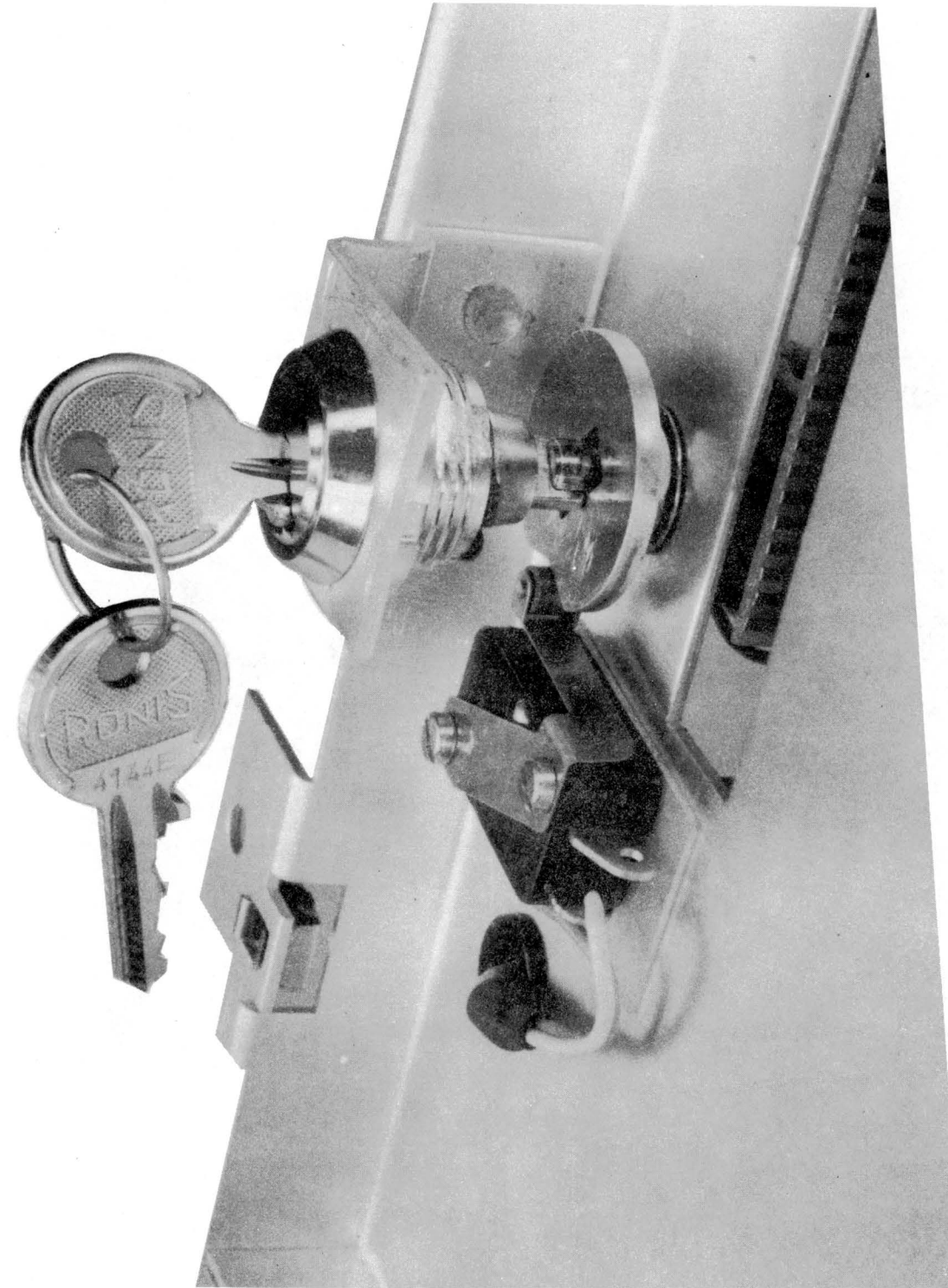


Figure 5-10 Detail of Key and Control-Panel Socket

SECTION VI

V24 SERIAL CONTROL UNIT

6.1 GENERAL

The V24 Serial Control Unit (V24-CU), located on the CPU card, interfaces an Operator's device with the CPU. The V24-CU converts between eight-bit parallel CPU characters (at +5/0V) for the CPU and bit-serial data (at +12/-12V) for the device. The V24-CU can provide parity bit insertion and checking; this feature is plug-programmable by means of U-links on the CPU/CU card.

Operating speed is also plug-programmable, with the possibilities :

110 baud, for ASRV24

600 baud, for PER 3100

1200 baud
2400 baud
4800 baud
9600 baud

} for display

The V24-CU uses direct interface connections with the CPU for address, command, and data to minimize GP-Bus utilization. The address and commands are received directly from the CPU K-register. CU/CPU data is received directly from the least-significant half of the L-register (L8-15) and sent directly to the CPU via the C-selector.

6.2 Channel and Interrupts

The CU operates via the system programmed channel, under direct control of the CPU. The CU sends an interrupt (INTSERN) to the CPU to request each character transfer and to indicate when it is in Wait Status mode.

6.3 Serial Data Format

The bit-serial data format for a single character is shown in Figure 6-1. The eighth "Data" bit may be either an actual CPU data bit or it may be a parity bit supplied and checked by the V24-CU. The parity bit, if used, can be either even or odd. Either one or two stop bits may be selected (by a plug on the card) to conform to the device requirements.

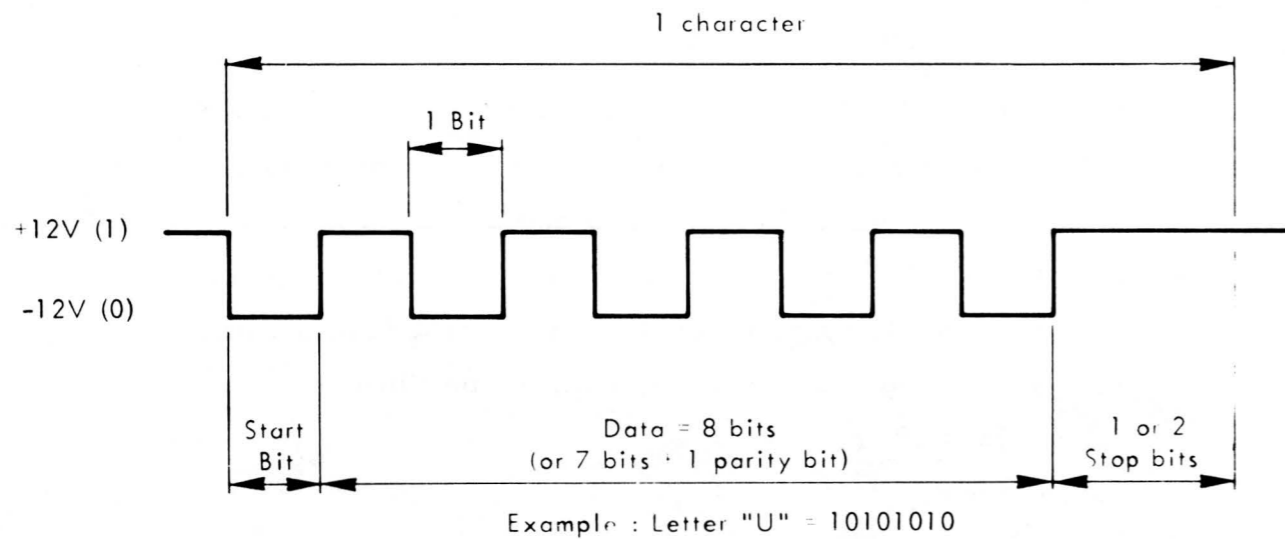


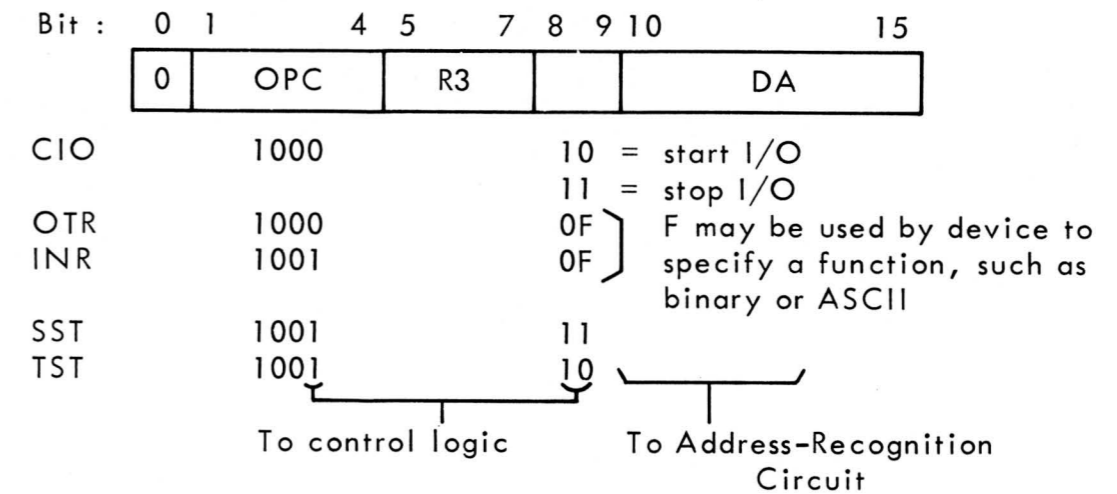
Figure 6-1 Serial Data Format

A logic One is represented by a high level (+12V) and a logic Zero is represented by a low level (-12V). When no character is being exchanged, the line is in the One state. Characters may be transferred end-to-end or may be separated by any amount of gap.

6.4 CPU INTERFACE AND CONTROL

The V24-CU operations are controlled by CPU I/O instructions (CIO Start, CIO Halt, OTR, INR, SST). These instructions are format-0, type T8. The instruction word contains address and control information which is sent directly to the CU on the K lines 4, 8-15, as follows :

K-Register



Additional control is provided directly from the CPU D-register. Bit D15 indicates input (D15 high) or output (D15 low) operations. This bit controls the setting of the CU flip-flop IN/OUT during a CIO Start operation. Bit D10 is set during a CIO Start operation to indicate Echo mode (routing the device input data directly back to the device during input data transfers).

6.5 The CU detects its address from K10-15 when it receives CPU timing signal TMR from the CPU. The CU recognizes its address, generates internal Address-Recognized signal TYARE/N, and sends AREDELA back to the CPU. The address code is determined by the addressing U-links. If the command is accepted, the CU sends signal TYAC back to the CPU; if the command is not accepted, TYAC is not sent and no other action is performed by the CU.

6.6 OPERATION

The V24-CU operates in six different states, as follows :

Operational State	F0/F1	Function
Inactive		The CU is inactive.
Execute-In		The CU is waiting for or receiving a character bit-serial from the device.
Exchange-In		The CU is waiting for or performing an INR command (Character to CPU).
Execute-Out		The CU is transmitting one character bit-serial to the device, or is waiting for a free line.
Exchange-Out		The CU is waiting for or performing an OTR command (Character from CPU).
Wait Status		The CU is waiting for or executing an SST command.

The operating flow is shown in Figure 6-2. The CU is set to the Inactive state by master clear (MCL/RSL), including at system power-on time. When the CU receives a CIO Start command (input or output), it goes to the Execute State if the device is operable; if the device is inoperable, the CU switches to Wait Status state.

6.7 In input mode, the CU goes to Execute-Input state and receives or waits for a character from the device. When the character has been received, the CU switches to Exchange-Input mode and transfers the character to the CPU, with an INR command.

6.8 In output mode, the CU switches from Execute to Exchange state as soon as the device is ready. The CU waits for, or receives, an OTR command with one character from the CPU. The CU switches to Execute-Output state to transfer the character bit-serial to the device, and switches back to Exchange state.

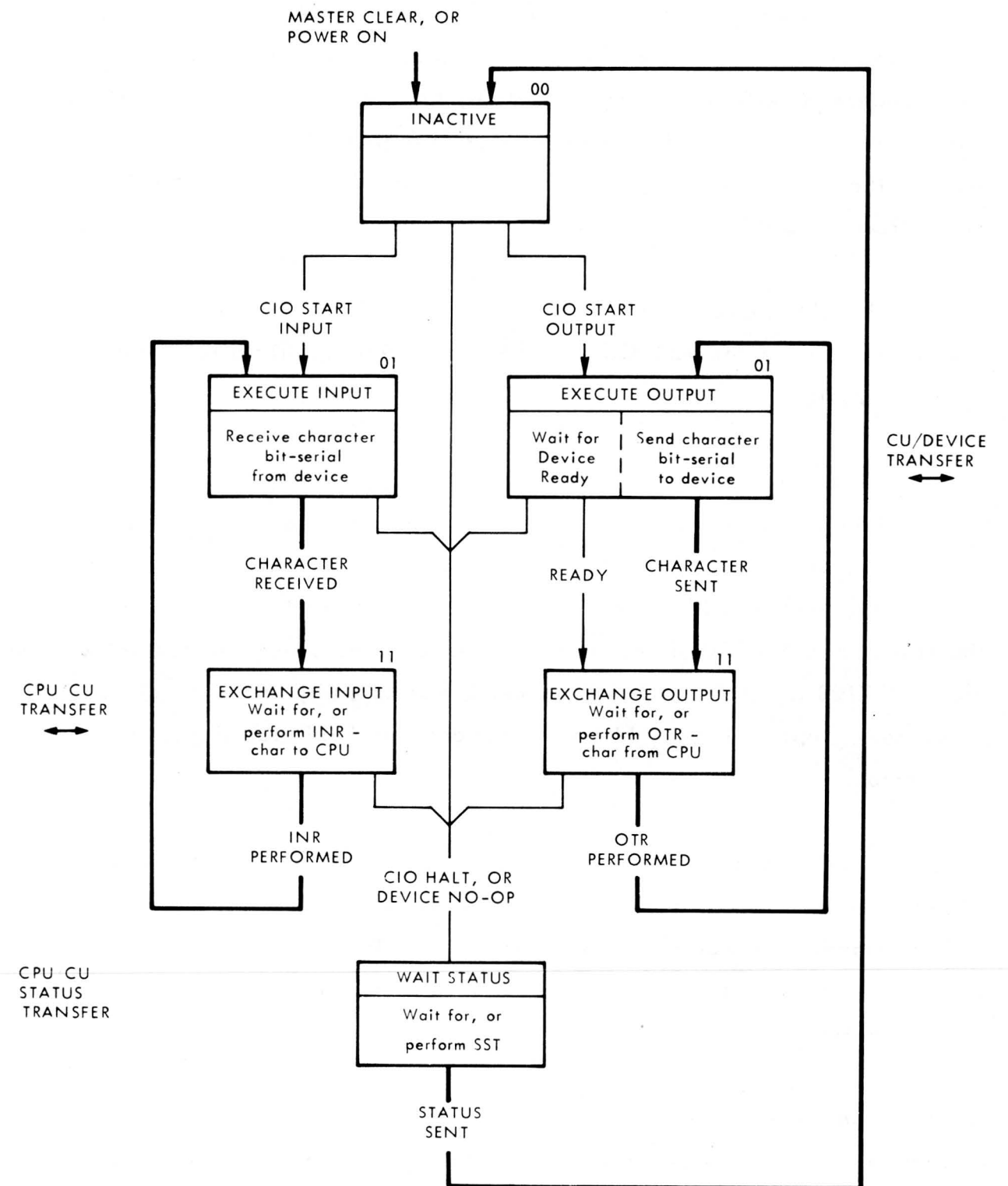


Figure 6-2 V24-CU Operational States

6.9 When the CU receives a CIO Halt command from the CPU, it goes to Wait Status state. The switch to Wait Status is not delayed if CIO Halt is received while the CU is exchanging a character with the device, because the exchange is performed independently by the UART chip once the CU has initiated the operation. If a new exchange is requested, however, it must wait for the end of the current transmission, when FTEOC is reset by the TEOC signal from the UART chip.

6.10 LOGIC DESCRIPTION

A block diagram of the V24-CU is shown in Figure 6-3. Detailed logic is provided in Figure 6-4.

6.11 Data Path

Output data is 8-bit parallel from the CPU L-register direct to the integrated receiver/transmitter circuit UART in the CU. The UART circuit outputs the character bit-serial (with the appropriate start, stop, and parity bits added) to the device on the CT104 line. Input data is received bit-serial from the device via the CT103 line. The input character is assembled by the UART circuit (with start, stop, and parity bits removed) and sent 8-bit parallel to the CPU C-selector.

6.12 The input data path to the CPU is via a type 74157 multiplexer circuit. This circuit is used to place status bits on the data lines during an SST command. Voltage conversion, from +5/0V logic to the +12/-12V used by the device, is performed by the type 1488 output inverters and the 1489A input inverters.

6.13 Character Conversion

Character conversion between bit-serial and bit-parallel is performed directly by receiver/transmitter circuit UART. During output operations, the UART circuit adds the start bit (logical zero) and stop bits (logical 1) to each character. Either one or two stop bits are added, depending on the U-link selection for the NSB input of the UART. (Two stop bits are normal, but a device may require just

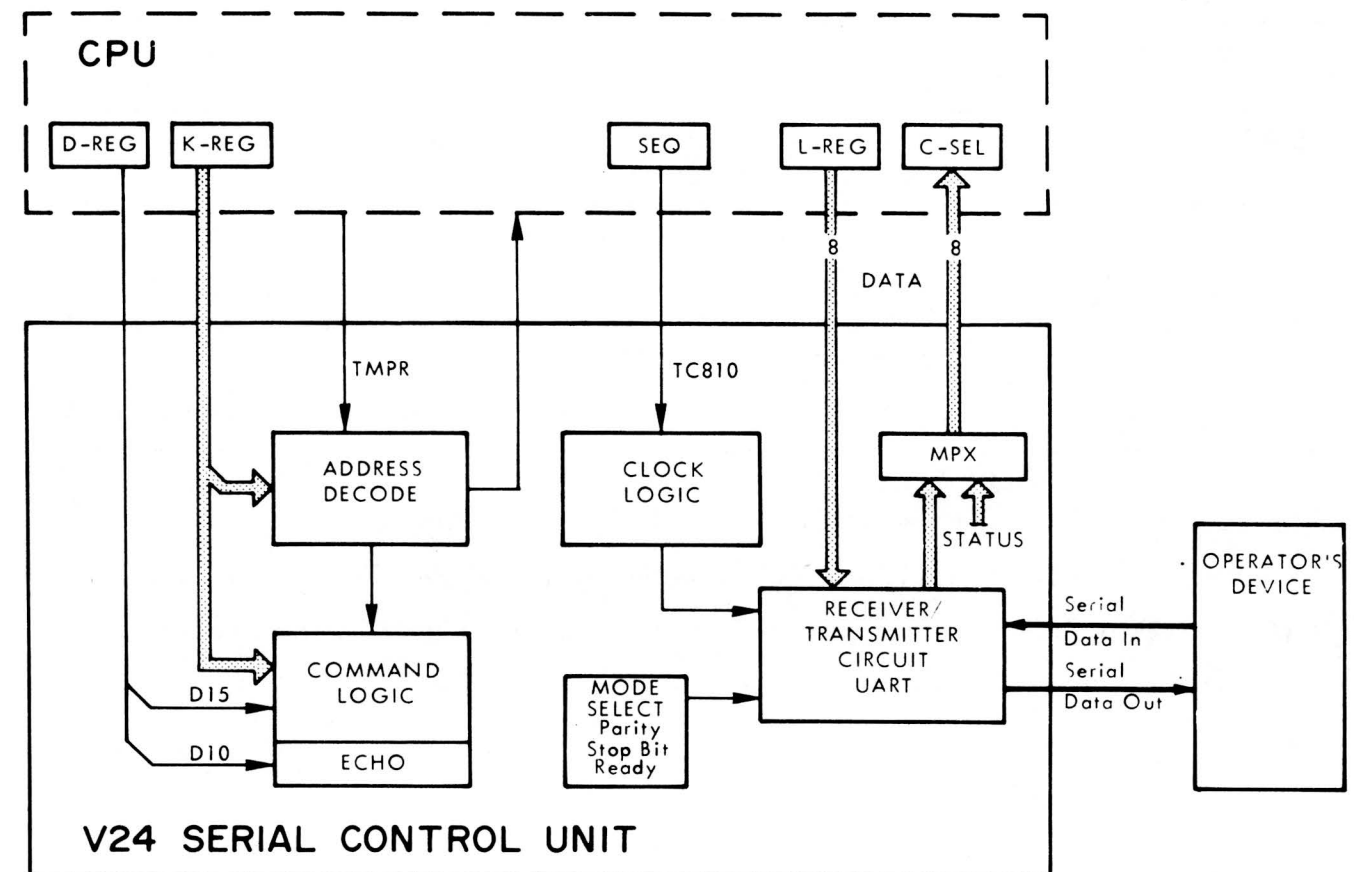


Figure 6-3 V24 Serial CU Block Diagram

one.) During input operations, the UART circuit strips off the same start and stop bits that it is set to add during output operations.

6.14 Transmitting speed is controlled by CU clock logic which is synchronized with the CPU clock. The basic clock signal is TC810 from the CPU. This clock rate is divided by type 74161 counters and flip-flops to provide clock rates for 110, 600, 1200, 2400, 4800, and 9600 baud. The desired rate is selected by positioning a U-link jumper. The selected clock signal drives the UART circuit, via flip-flop CLV24, through the TCP and RCP inputs.

6.15 Parity

The UART circuit can be manually programmed (by means of circuit U-links) to operate without parity, or to insert and check either even or odd parity. With no-parity selected (high level to the UART inputs NPB and ND1), all eight data bits to and from the CPU are sent and received as data bits on the device data lines. If the U-link is set to select parity operation (NPB/ND1 low), a second U-link at the POE input determines whether even or odd parity is to be used.

6.16 Whenever parity is selected, the UART circuit ignores the least-significant bit to and from the CPU (bit 15). For output operations, the UART circuit determines the parity of the seven data bits used and places the resulting parity bit in place of the eighth data bit with the serial character to the device. For input operations, the UART circuit checks the parity of the first seven data bits received, and compares the result with the eighth (parity) bit. The UART circuit indicates an error by activating the PE signal. A parity error sets the CHIPER and FHALT flip-flops and switches the CU to Wait Status state.

6.17 Device Interface

All CU-Device signals are via connector 1, as follows :

Conn. 1	Line	Signal
1A30	CT101	Mechanical Ground, to/from GP Bus connector 3
1B31 to 1B37	CT102	Ground
1A27 1B27	--	Signal Ground
1A28 1B28	--	+5V
1A31	CT103	Data In, serial data from device to CU.
1A32	CT104	Data Out, serial data from CU to device.
1A33	CT106	Not Used
1A34	CT107	Not Used
1A36	CT109	Not Used
1A35	CT1082	Not Operable, low when device is connected and switched on.
1A37	CT133	DREADY, Device ready to transfer next output character. If the device does not provide this signal, a U-link on the CU must be positioned to hold this signal active within the CU.

6.18 Echo Mode

Echo Mode is programmable by bit 10 in the R3 register associated with the CIO Start; this is indicated to the CU during CIO Start by D10 to the FECHO flip-flop. Each accepted INR command is memorized by the FACINR flip-flop. At T7 of the CPU ROM address 159, the active MUQ1 sets the CU flip-flop ECHO, and TDSN is validated up to T5 of the next cycle. The low level to the TDN input of the UART circuit causes the input data to be channelled back to the Data Out line to the device. The FACINR flip-flop remains set until the Wait Status state when it is reset by FNU from the CPU.

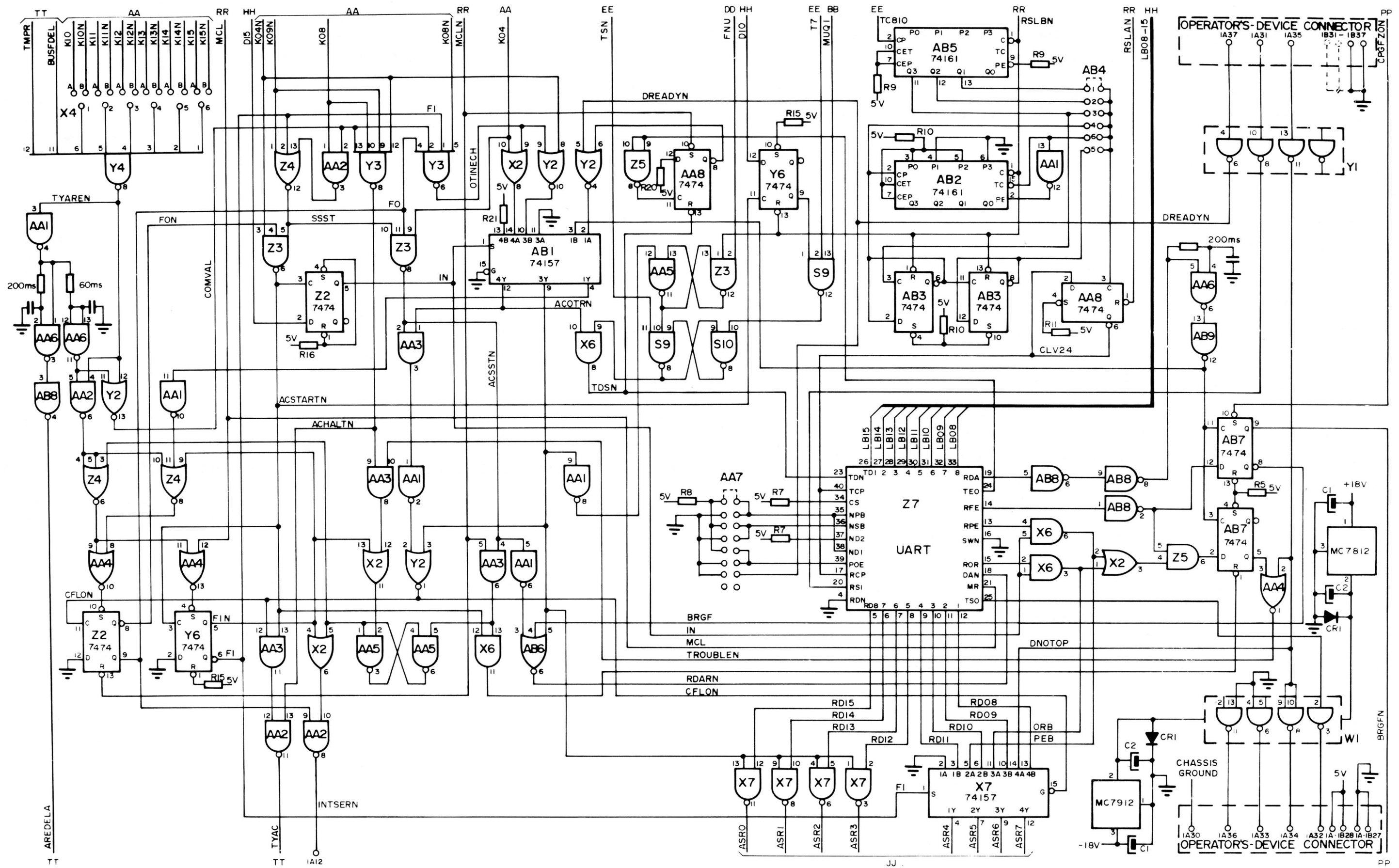


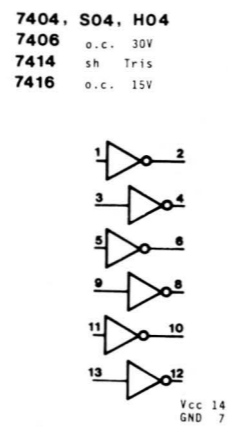
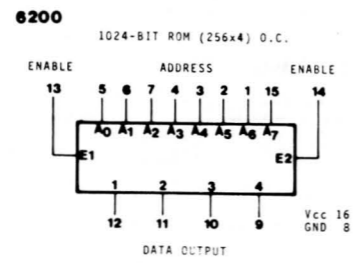
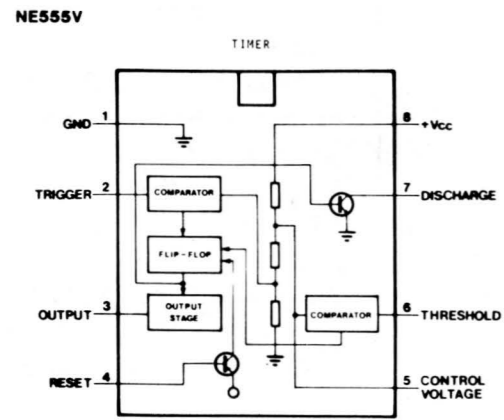
Figure 6-4 V24 Serial Control Unit Detailed Logic Diagram

APPENDIX A INTEGRATED CIRCUITS

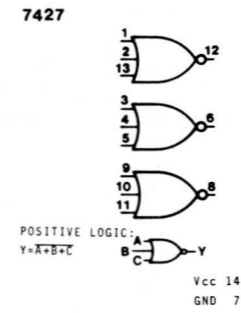
IC List for CPU Card, C.Panel, P.Supply

Integrated Circuit	Used on Logic Sheet (except common)
555 1488 1488X 1489A 1801 (7438) 6200 7400 74S00 7402 7403 7404 74S04 7408 7410 74S10 74S11 7416 7417 7420 74S20 7425 7427 7428 7430 74S30 7432 7437 7450 7453 74S64 7474 74S74 7485 7486 7488A 7489 (93403) 74S112 74121 (9603) 74132 74S138 74148 (9318) 74151A 74157 74158	<p>Timer</p> <p>Line Receiver, quad NAND, quad 2-input 1024-Bit ROM (256x4) oc. NAND, quad 2-input " NOR, quad 2-input NAND, quad 2-input INVerter, hex " AND, quad 2-input NAND, trip 3-input " AND, trip 3-input " AMPLifier, hex buff/driver ov. NAND, dual 4-input " NOR, dual 4-input strobed NOR, trip 3-input " NAND, sing 8-input " OR quad 2-input NAND, quad 2-input AND/OR Inv Exp, dual 2x2-in. AND/OR Inv Exp, sing 4x2-in. AND/OR Invert, 4-3-2-2 input D Flip-flop, dual " 4-bit Comparator Exclusive-OR, quad 256-bit ROM (32x8) 64-bit RAM (16x4) JK Edge-Trig Flip-Flop, dual SS Multivibrator NAND, quad 2-in. Schmitt Trig. Line Decoder Priority Encoder Multiplexer, sing 8-input Multiplexer, quad 2-input Multiplexer, quad 2-input, inv.</p> <p>P.Supply</p> <p>LL</p> <p>C.Panel</p> <p>AA FF</p> <p>SS AA FF,NN MM RR,TT SS</p> <p>BB SS DD,EE, JJ,MM,NN NN,V24,C.Panel SS</p>

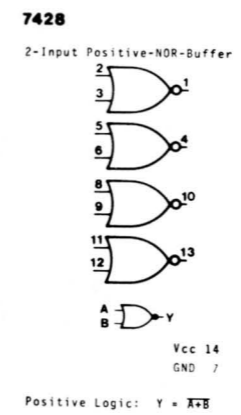
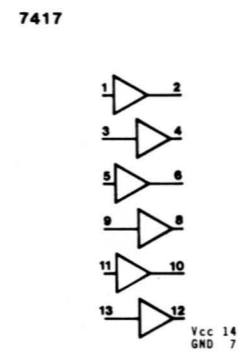
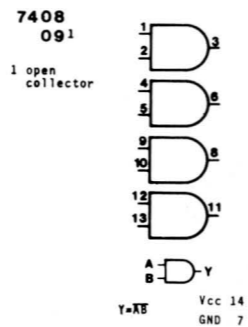
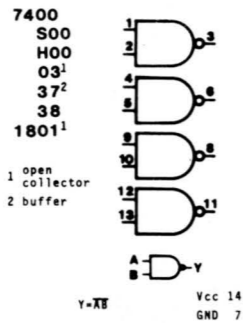
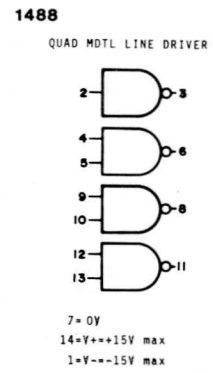
Integrated Circuit	Used on Logic Sheet (except common)
74161 (9316) 74S169 74174 74S174 74175 74S175 74181 (9341) 74182 74194 74298 74297 7812 (TO22O) 7815 (TO22O) 7905 (TO22O) 7912 (TO22O) 8205 8234 8576 9301 9308 9309 9324 9602 ADL2011 CR2021 RCF REC SN74LS169N SWITCH TER UART UGA7723393 μA710 μA723	<p>4-bit Synchronous Counter 4-bit up/down Counter 6-bit Register (hex D-F/Fs) " 4-bit Register (quad D-F/Fs) " 4-bit ALU, 24-pin Look-Ahead Carry Generator 4-bit Shift Register Multiplex/Register, quad 2-in. " " " 4096-bit ROM (512x8) Multiplexer, dual 4-in.oc Programmable Logic Array 1-of-10 decoder " Multiplexer, Dual 4-input 5-bit Comparator SS Multivibrator, dual retriq. " " INVerter, hex " U-Link Jumper Connection Ternet double resistor Receiver/Transmitter, 40-pin see μA723 Linear Amplifier Linear Amplifier</p> <p>EE,MM,V24 KK,LL,C.Panel TT</p> <p>AA,MM,NN,SS,TT</p> <p>GG GG JJ GG,HH</p> <p>BB JJ,LL AA LL</p> <p>FF,HH, JJ C.Panel P.Supply</p> <p>V24</p> <p>P.Supply P.Supply</p>



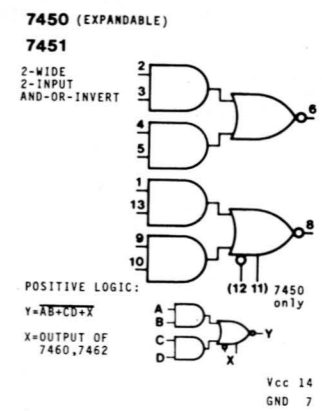
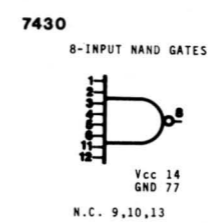
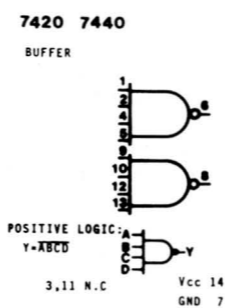
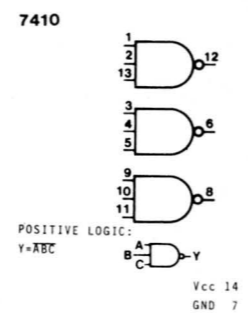
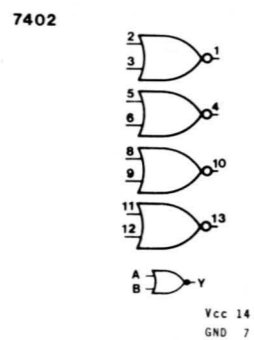
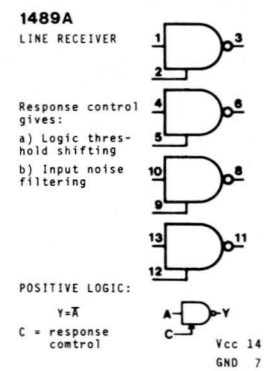
7416
(see 7404)



7437
(see 7400)

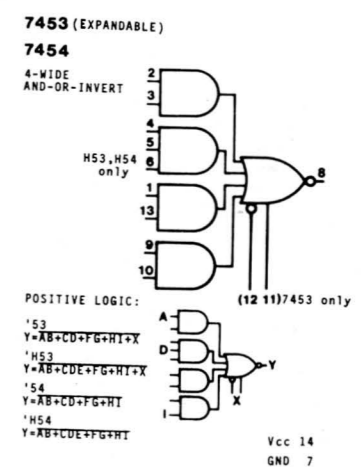
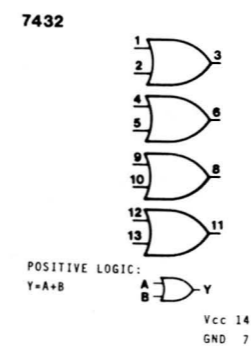
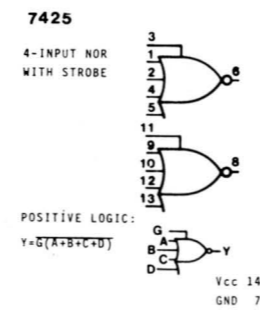
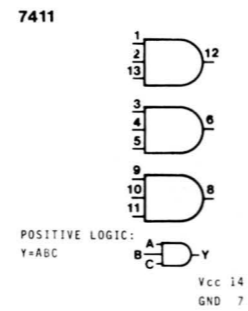


7438
(see 7400)

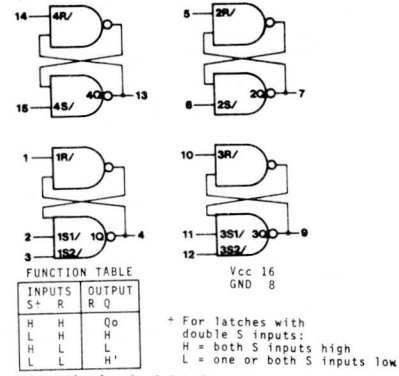


1801
(see 7400)

7403
(see 7400)



74279 QUADRUPLE S-R LATCHES

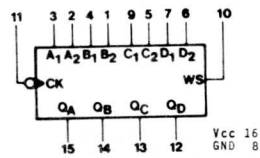


FUNCTION TABLE

INPUTS	OUTPUT
S+ R	Q
H H	Q ₀
L H	H
H L	L
L L	H*

* For latches with double S inputs: H = both S inputs high L = one or both S inputs low
 Q₀ = the level of Q before the indicated input conditions were established. This output level is pseudo stable; that is, it may not persist when the S and R inputs return to their inactive (high) level.

74298 QUADRUPLE 2-INPUT MULTIPLEXER WITH STORAGE

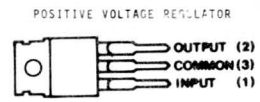


FUNCTION TABLE

WORD SELECT	CLOCK	QA	QB	QC	QD
L	+	a1 b1 c1 d1			
H	+	a2 b2 c2 d2			
X	+	QA0 QB0 QC0 QD0			

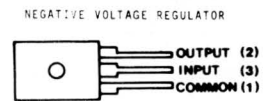
a1, a2, etc. = the level of steady-state input at A1, A2, etc.
 QA0, QB0, etc. = the level of QA, QB, etc. entered on the most-recent transition of the clock input.

7812C (+12V)
7815C (+15V)



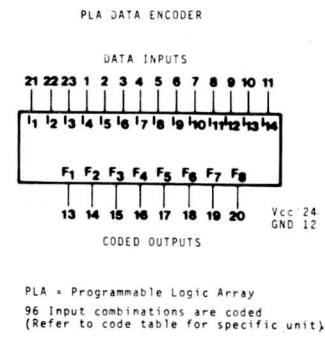
7815
 (see 7815)

7905 (-5V)



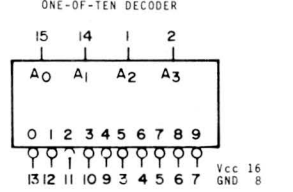
7912
 (see 7812)

8576



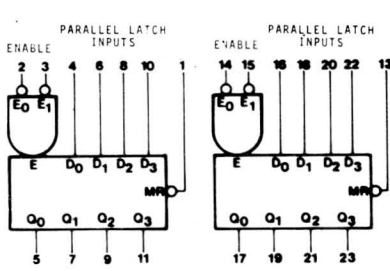
PLA = Programmable Logic Array
 96 Input combinations are coded (Refer to code table for specific unit)

9301



Multipurpose Decoder will accept four inputs and provide 10 mutually exclusive outputs.

9308

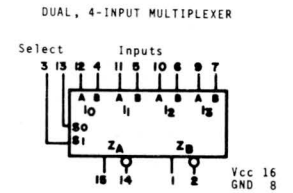


FUNCTION TABLE

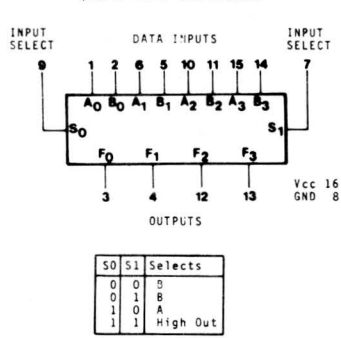
MR	E0	E1	D	Q _n	OPERATION
H	L	L	L	L	Data Entry
H	L	L	H	H	Data Entry
H	L	X	X	Q _{n-1}	Hold
H	H	L	X	Q _{n-1}	Hold
H	H	H	X	Q _{n-1}	Hold
L	X	X	X	L	Reset

Q_{n-1} = Previous Output State
 Q_n = Present Output State

9309



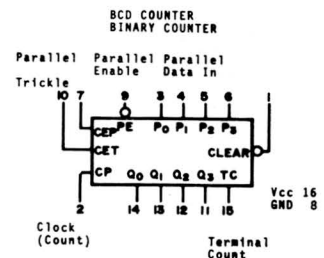
8234



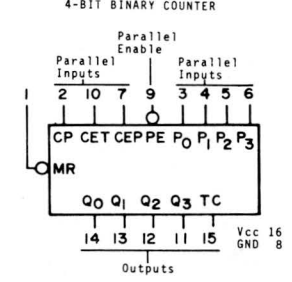
FUNCTION TABLE

S0	S1	Selects
0	0	A
0	1	B
1	0	A
1	1	High Out

9310, 74181

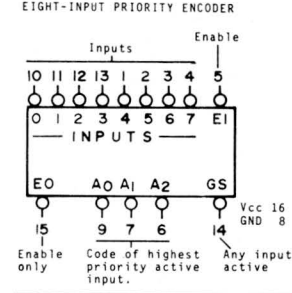


9316



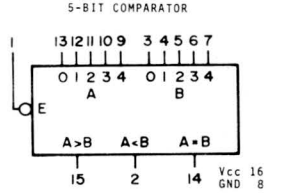
TC = Q₀Q₁Q₂Q₃CEP

9318



9318 is a Multipurpose Encoder designed to accept eight inputs and produce a binary weighted code of the highest order input.

9324



A High Speed Expandable Comparator that provides comparison between two 5-bit words and gives three outputs: "less than", "greater than" and "equal to". A HIGH level on the active LOW enable input forces all three outputs LOW.

9341

(see 74181)

ADL 2011

(see 74188)

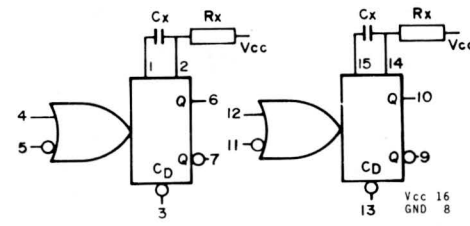
93403

(see 7489)

CR202A

(see 74188)

9602

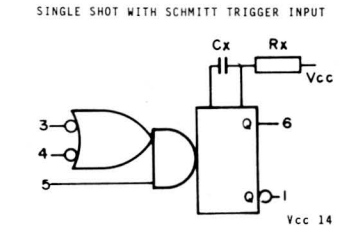


TRIGGERING FUNCTION TABLE

PIN NO'S		Operation
5(11)	4(12) 3(13)	
H-L	L	Trigger
H	L-H	Trigger
X	X	Reset

Dual Retriggerable, Resettable Monostable Multivibrator provides an output pulse whose duration and accuracy is a function of external timing components Ex and Rx.

9603



FUNCTION TABLE

t _n	INPUT	t _{n+1}	INPUT	OUTPUT
L	X	L	X	H
X	L	X	L	H
H	H	X	L	H
H	H	L	X	H

* = Triggering Transition

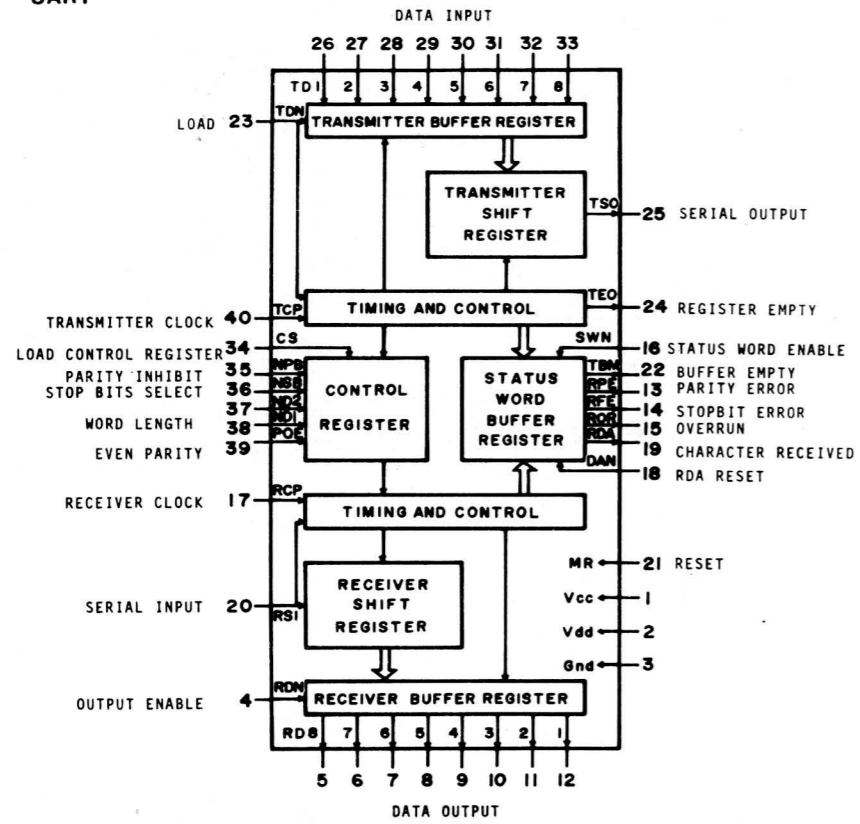
REC

(see 7404)

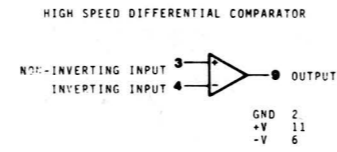
UART

(see over)

UART



μA710DC



μA723

