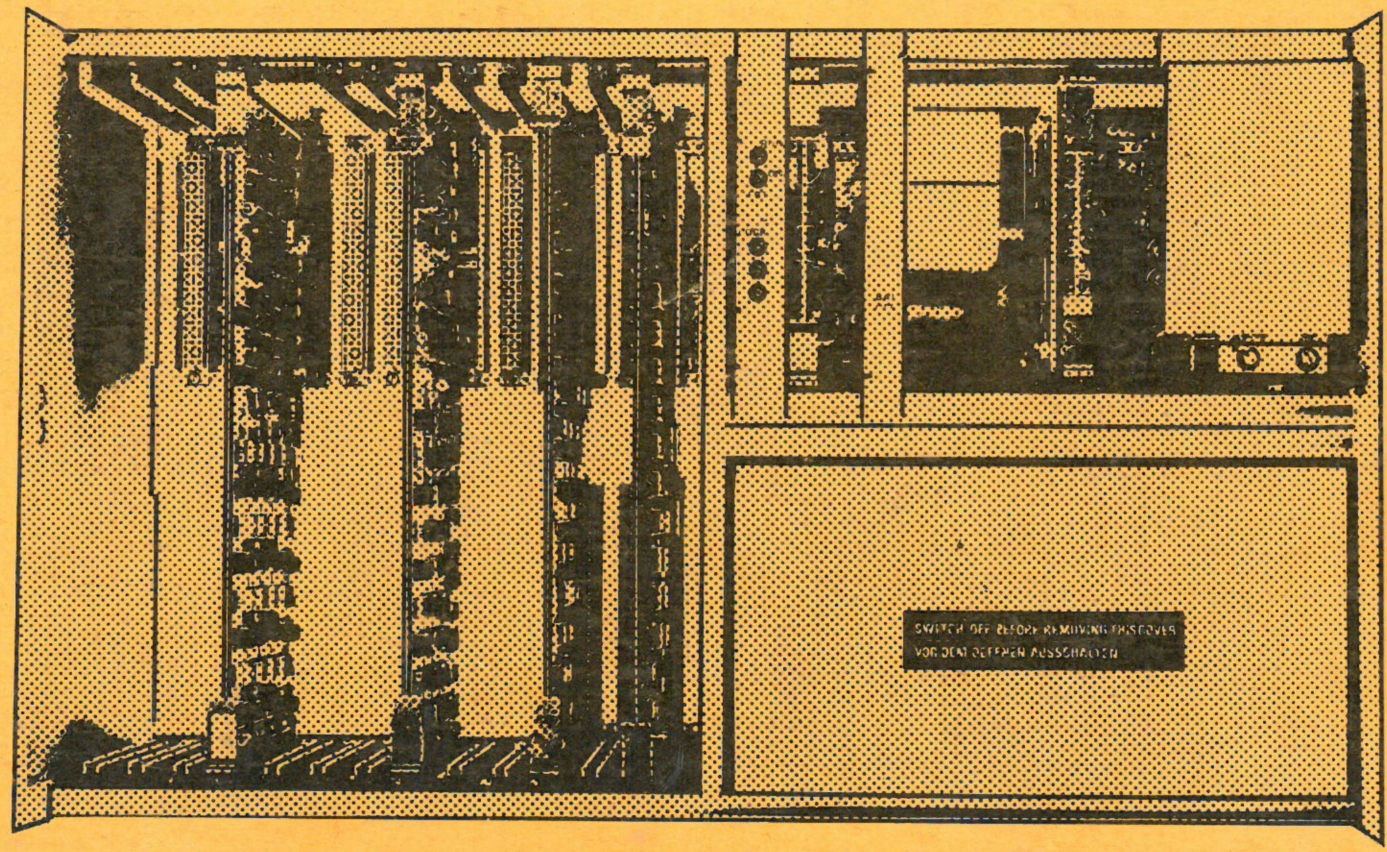
Data Systems

PHILIPS

P851M Volume 1
Central Processor & Memories
Technical Manual



P851M

TECHNICAL MANUAL

Volume 1 — CENTRAL PROCESSOR AND MEMORIES

The information contained in this manual is based on the documentation available at the time of printing, October 1978. 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 28073

October 1978
Copyright © by Philips Data Systems B.V. 1978
All rights strictly reserved. Reproduction or issue to third parties in any form whatever is not permitted without written authority from the publisher.

TABLE OF CONTENTS

Paragraph		Page
	I SYSTEM DESCRIPTION	1-1
1.1	General	1-1
1.2	Eurocards	1-2
1.3	Cabinet	1-2
1.4	Control Panels	1-2
1.5	Data Transfers	1-3
1.6	Programmed Channel	1-3
1.7	Wait Mode	1-3
1.8	Interrupt Mode	1-3
1.9	Input/Output Processor Channel	1-3
1.10	IOP Priority System	1-3
1.11	IOP Uses Three Data Paths	1-8
1.12	P851 Control Instructions	1-8
1.17	Data Formats	1-9
1.18	Memory Addressing	1-9
	II INSTALLATION AND INTERFACE	2-1
2.2	Site Preparation	2-1
2.3	Environmental Conditions	2-1
2.4	Electrical Requirements	2-2
2.5	Installation	2-2
2.6	Eurorack	2-2
2.7	Air Ventilation Units	2-5
2.8	Power Distribution Panels	2-5
2.9	Configuration Guide	2-10
2.10	Rack Earthing and Cabling Rules	2-12
2.11	Installing	2-12
2.12	Interface	2-13
2.13	Physical Aspects of the GPBS Connections	2-13
2.14	GPBS Signals	2-13
2.15	Data Transfers	2-15
2.16	I/O Instructions	2-15
2.17	Programmed Channel	2-15
2.18	IOP Channel	2-15
2.19	Exchange Timing	2-16
2.20	Interrupt Signals	2-17
2.21	Break Signals	2-18
2.22	Back Panel Connector Pinning	2-18
2.23	Interfacing for Customer Designed Control Units	2-18
2.24	Card Size	2-18

Paragraph		Page
2.25	General Purpose Card	2-19
2.26	Signals	2-19
2.29	Major Logic Units	2-21
2.35	Design Rules	2-22
2.42	Rules for Connecting Grounds in a System	2-26
2.43	Grounding for Cabinets and Racks	2-26
2.44	Logic Ground and Mechanical Ground	2-27
2.45	Flat Cables	2-27
2.46	Mains Cables	2-28
2.47	General Rules for Connecting Shielded Cables	2-28
2.48	Special Rules for Connecting Shielded Cables	2-29
2.49	Connecting the Ground Lead in a Cabinet or Rack	2-29
2.50	Connecting a Ground Lead at the Devices	2-30
	III POWER SUPPLY	3-1
3.1	Description	3-1
3.2	Input Voltages	3-1
3.3	Output Voltages	3-1
3.4	Logic Signals	3-1
3.5	Mechanical	3-2
3.6	Block Diagram	3-2
3.7	Mains Transformer Chassis	3-2
3.8	Mains Voltage Connections	3-2
3.9	Rectifier and Transformer Protection	3-2
3.10	TDA 2640 Control and Protection Chip	3-2
3.11	Miller Integrator	3-2
3.12	Comparator	3-2
3.13	Pulse Width Modulator	3-2
3.14	Output Stage	3-5
3.15	Soft-Start and Cut-Out	3-5
3.16	Counting Circuit	3-5
3.17	Overvoltage Protection	3-5
3.18	Overcurrent Protection	3-5
3.19	Low Supply Voltage Protection	3-5
3.20	Loop Fault Protection	3-5
3.21	Inhibiting Circuit	3-5
3.22	+5V Regulator Card	3-5
3.23	+12V and -12V Regulators	3-5
3.24	+5V Regulator	3-5
3.27	Memory Supply Card	3-5
3.28	+5VM Regulator	3-5
3.31	+12VM Regulator	3-6
3.33	+10V Regulator	3-6
3.34	Sequence Card	3-6
3.35	Power Fail/Reset Logic	3-6
3.36	Real Time Clock	3-6
3.37	+10V Inhibit	3-6
3.38	Battery Control	3-6
3.43	Power On/Off Sequence	3-7
3.44	Backpanel Connections	3-7
3.45	Adjustments	3-7
	IV CONTROL PANELS	4-1
4.2	Mini Panel	4-1
4.3	Switches and Lamps	4-1
4.4	Logic Diagram	4-2
4.5	Start Logic	4-2
4.6	IPL/Program Loading	4-2
4.7	Interrupt Logic	4-3
4.12	Test	4-4
4.13	PROM Contents	4-4
4.14	Run Mode	4-5
4.15	Power Failure/Automatic Restart	4-5
4.16	Remote Loading	4-5
4.17	U Link Data	4-5
4.18	Back Panel Connector	4-5
4.19	Logic and Card Details	4-5
4.20	Full Panel	4-9
4.21	Control Panel Switches and Lamps	4-9
4.22	Safety-Key	4-10
4.23	Inter Panel Logic Description	4-10
4.30	Test	4-12
4.31	ROM Contents	4-12
4.32	U Link Data	4-12
4.33	Back Panel Connector	4-12
4.34	Control Panel Interface	4-12
4.35	Logic Diagrams	4-12
4.36	Components and Component Layout	4-12
	V CPU	5-1
5.1	Introduction	5-1
5.2	Terms Used in the Text	5-2
5.3	Interface Signals	5-2
5.4	Major Units of the CPU	5-2
5.5	CPU Clock	5-4
5.6	Clock Starts at Switch-On	5-4
5.7	Bus Control	5-4
5.8	Relationship of the Internal and I/O Bus	5-5
5.9	Internal Bus Transfers	5-5
5.10	I/O Bus Transfers	5-5

Paragraph		Page
5.11	Allocating the Bus for IOP Transfers	5-6
5.12	Allocating the Bus for Program Channel Transfers	5-6
5.13	Clock Inhibit Logic	5-7
5.14	IOP Transfers	5-7
5.15	Prevent CPU Taking the Bus	5-7
5.16	Program Channel Transfers	5-7
5.17	Prevent I/O Processor Taking the Bus	5-8
5.18	ANSF F/F Control Logic	5-8
5.19	Timeout	5-8
5.20	Timeout for a CPU Transfer	5-8
5.21	Timeout for IOP Transfer	5-8
5.22	Interrupts	5-9
5.23	Interrupt Received Routine	5-9
5.24	Service Interrupt	5-11
5.25	Return to the Interrupted Routine	5-11
5.26	Internal Interrupts	5-11
5.27	Planet	5-12
5.28	Program Status Word	5-12
5.29	Planet Decodes the RS Field	5-13
5.30	Control Lines	5-13
5.31	Microcommands	5-13
5.32	Setting up the PLA Lines	5-14
5.33	Bus Selection	5-14
5.34	Planet Checks for Fetch Interrupt or Panel Routines	5-15
5.35	SPALU	5-16
5.36	Spalu Operation	5-16
5.37	Arithmetic Logic Unit	5-16
5.38	Scratch Pad	5-16
5.39	Shift Operations	5-17
5.40	Switch-On	5-19
5.41	Master Clear at Control Panel	5-19
5.42	Control Panel Data to CPU	5-19
5.43	Control Panel to CPU Handshaking Routine	5-19
5.44	Load Register Button Routine	5-20
5.45	Read Register Button Routine	5-20
5.46	Read Memory Button Routine	5-21
5.47	Load Memory Button Routine	5-21
5.48	Instruction Step Button Routine	5-21
5.49	Read Program Status Button Routine	5-21
5.50	Initial Program Load Button Routine	5-22
5.51	Program Run Button Routine	5-22
5.52	Control Panel Interrupt Button Routine	5-22
5.53	Mini-Panel Data to CPU	5-22
5.54	Test CPU Button Routine	5-22
5.55	Initial Program Loading	5-22
5.56	CPU Operating Modes	5-25
5.57	Privileged Instructions	5-25
5.58	System and User Modes	5-25
5.59	Executing a Trap Routine	5-27
5.60	Internal Interrupts	5-27
5.61	Power Failure / Automatic Restart	5-27
5.62	Power Failure Routine	5-27
5.63	Automatic Restart Routine	5-29
5.64	Real Time Clock	5-29
5.65	Instruction Handling	5-29
5.66	Load Instructions	5-29
5.67	Store Instructions	5-29
5.68	Arithmetic Instructions	5-29
5.69	Logical Instructions	5-29
5.70	Character Handling Instructions	5-30
5.71	Branch Instructions	5-30
5.72	Shift Instructions	5-30
5.73	Control Instructions	5-30
5.74	Input Output Instructions	5-30
5.75	P851 List of Instructions	5-30
5.76	Control Instructions -- Detailed Description	5-32
5.77	Reset Internal Interrupt	5-32
5.78	Enable and Inhibit Interrupts	5-32
5.79	Halt CPU	5-33
5.80	Microinstruction	5-33
5.81	ROM Addressing Routine	5-33
5.82	RS Field Decodes	5-34
5.83	PLA Decodes	5-34
5.85	RAD Decodes	5-36
5.87	NA9,10 Decodes	5-36
5.88	Addressing ROM Bits	5-36
5.90	Addressing Modes	5-38
5.91	GM Decodes	5-38
5.93	Using the Test Tool	5-94
	VI MICRODIAGNOSTICS AND TROUBLESHOOTING	6-1
6.2	Fault Finding Procedure	6-1
6.3	Microdiagnostic Routines	6-3
6.4	CPU Tests	6-3
6.7	CU Test	6-3
6.9	Memory Test	6-3
6.11	Troubleshooting the CPU, CU, and Memory Cards	6-4
6.12	Internal Bus Fault	6-4
6.13	Spalu Flags and Flanet Faults	6-4
6.15	Data Path Test	6-4
6.16	GPBS Fault	6-5
6.17	Control Unit Fault	6-5
6.18	Memory Fault	6-5
6.19	Bootstrap ROM Fault	6-5

Paragraph		Page
6.21	Software IPL	6-6
6.22	Loading the IPL	6-6
6.23	Low Core IPL	6-6
6.24	High Core IPL	6-6
6.25	Verification of the Low Core IPL	6-7
6.26	Verification of the High Core IPL	6-7
6.28	Simple Test Programs	6-54
6.29	Memory Test Program MEMHAN	6-54
6.30	Program CHECK	6-56
6.31	ASR, PER3100, and Display	6-58
6.32	Program LINE	6-58
6.33	Programs NOECHO and ECHO	6-60
6.34	Program COPY	6-60
6.35	Program DUMP	6-60
6.36	Producing Programs on Paper Tape	6-66
6.37	Using the ASR Punch in LOCAL Mode	6-66
6.39	Program ASC4x4	6-66
6.42	Program HEXTAP	6-67
6.43	Suggestions for Sophisticated Programs	6-73
6.44	Routine Mini-IPL	6-73
6.45	Routine for using the INT Button	6-75
6.46	Program CPY4x4	6-75
6.47	Program MEM51	6-79
6.48	Programming for the Mini-Panel	6-83
6.52	Standard Test Programs	6-84
	VII SCUZ CU	7-1
7.1	Introduction	7-1
7.2	Interface Signals	7-2
7.3	V24 and Current Loop Interfaces	7-3
7.6	Major Logic Units	7-3
7.7	BIO Transceivers	7-3
7.8	UART Clock	7-3
7.9	Control Logic Clock	7-3
7.10	Interface Timing F/Fs	7-3
7.11	TranBuf Ready F/F	7-3
7.12	Interrupt F/F	7-4
7.13	Break Key Detection F/F	7-4
7.14	Status F/Fs	7-4
7.15	Address Recognition Logic	7-4
7.16	UART (Universal Asynchronous Receiver Transmitter)	7-4
7.22	Parity Error Logic	7-5
7.23	Control Word Generator	7-6
7.24	Input Operation	7-9
7.25	Operator Requests Transfer	7-9
7.26	Initiate	7-9
7.27	Device Inputs Character	7-9
7.28	Character Input to CPU	7-9
7.29	End of Transfer	7-9
7.30	Output Operation	7-11
7.31	Initiate	7-11
7.32	Character Output to SCUZ	7-11
7.33	End of Transfer	7-11
7.34	Installation	7-12
7.35	Device Interfaces	7-12
7.36	U Links	7-12
	VIII 4-8K DYNAMIC RANDOM ACCESS MEMORY	8-1
8.2	Physical	8-1
8.3	Block Diagram	8-1
8.4	Memory Voltages	8-1
8.5	Memory Array	8-2
8.6	Bit Selection	8-2
8.7	Data In/Out	8-3
8.8	Addressing Logic	8-3
8.9	Memory Cycle Addressing	8-3
8.10	Refresh Cycle Addressing	8-3
8.11	Memory Module Addressing	8-4
8.12	Control Logic	8-4
8.13	Refresh Cycles	8-4
8.14	External Cycles	8-4
8.15	Timing Logic	8-4
8.16	Address Multiplex Clock	8-5
8.17	Column Address Clock	8-5
8.18	End of Access Clock	8-5
8.19	Memory to CPU Clock	8-5
8.20	Timing Diagrams	8-5
8.21	Data In/Out Logic	8-6
8.22	Word Handling	8-6
8.23	Character Handling	8-6
8.24	Read/Write Modes	8-6
8.25	Bus Transceivers	8-6
8.26	D.C. Converter	8-6
8.27	Interface Connections	8-6
8.28	Reset Logic	8-6
8.29	Logic Diagrams, Cards, and Card Components	8-6
	IX 0.5, 1, 2K STATIC RANDOM ACCESS MEMORY	9-1
9.2	Physical	9-1
9.3	Block Diagram	9-1

Paragraph		Page
9.4	Memory Voltages	9-1
9.5	Memory Array	9-3
9.6	Bit Selection	9-3
9.7	Data In/Out	9-3
9.8	Addressing Logic	9-3
9.9	Module Addressing	9-3
9.10	Control Logic	9-4
9.11	Read/Write	9-4
9.12	Word/Character Mode	9-4
9.13	Timing Logic	9-4
9.14	Start Access Time	9-4
9.15	Delay Write Enable	9-4
9.16	Memory to CPU Clock	9-4
9.17	Stop Internal Clock	9-4
9.18	Timing Diagram	9-4
9.19	Data In/Out Logic	9-5
9.20	Word Handling	9-5
9.21	Character Handling	9-5
9.22	Bus Transceivers	9-5
9.23	Memory Inhibit	9-5
9.24	Reset Logic	9-5
9.25	Interface Connections	9-5
9.26	Logic Diagram, Cards, and Card Components	9-5
	X 16k - 32k DYNAMIC RANDOM ACCESS MEMORY	
10.1	Introduction	10-1
10.2	Construction	10-1
10.3	Voltages	10-2
10.4	Functional Schematic Diagram	10-2
10.6	Memory Cycle	10-4
10.12	Refresh Cycle	10-4
10.13	Logic Description	10-4
10.14	Operation of a 16k MOS Dynamic RAM Chip	10-4
10.16	Addressing Logic	10-5
10.21	Timing Logic	10-6
10.30	External Cycle Control Logic	10-9
10.38	Refresh Cycle Control Logic	10-11
10.40	Voltage Regulator	10-11
10.41	Power Failure	10-11
10.42	Interface Connections	10-11
10.43	Logic Diagram and Card Components	10-11
	APPENDIX A INTEGRATED CIRCUITS	A-1

LIST OF ILLUSTRATIONS

Figure		Page
1.1	Eurorack with Mini Control Panel	1-4
1.2	Interconnections for Both Control Panels	1-6
1.3	Programmed Channel Transfers	1-6
1.4	Multiplexed I/O Processor Channel Transfer	1-7
2.1	Eurorack Installation Data	2-3
2.2	Eurorack Layout Details	2-4
2.3	Ventilator Units	2-5
2.4	10 Ampere Distribution Panel	2-6
2.5	10 Ampere Panel Circuit	2-7
2.6	25 Ampere Distribution Panel	2-8
2.7	25 Ampere Panel Circuit	2-9
2.8	Configuration Examples	2-11
2.9	Block Diagram of Interface	2-13
2.10	Exchange Timing at Master Level	2-16
2.11	Exchange Timing GPBS Level	2-17
2.12	Interrupt System	2-17
2.13	General Purpose Card	2-20
2.14	General Purpose Card Logic Diagram	2-23
2.15	General Purpose Card Component Layout	2-24
3.1	Mechanical Layout of Sub-Assemblies	3-3
3.2	Power Supply Block Diagram	3-4
3.3	Block Diagram of TDA 2640 Chip	3-4
3.4	Power Sequence Timing	3-7
3.5	Position of Potentiometers	3-8
3.6	+5V Regulator Card Schematic	3-9
3.7	Memory Supply Card Schematic	3-10
3.8	Sequence Card Schematic	3-11
3.9	Mains Transformer Chassis Component Layout	3-12
3.10	+5V Regulator Card Component Layout	3-13
3.11	Memory Supply Card Component Layout	3-14
3.12	Sequence Card Component Layout	3-15
3.13	+5VR Regulator Card Schematic	3-16
3.14	+5VR Regulator Card Component Layout	3-16
4.1	Mini Panel	4-2
4.2	Data Switches	4-3
4.3	IPL Timing	4-3
4.4	Timing for INT, PWF, ARF, or RTC	4-4
4.5	U Link Data	4-6
4.6	Logic Diagram	4-7
4.7	Mini Panel Card	4-8
4.8	Full Panel	4-9
4.9	Control Panel Timing	4-11
4.10	U Link Data	4-14
4.11	Logic Diagram of Inter Panel	4-16
4.12	Layout of Inter Panel Card	4-17

Figure		Page
4.13	Logic Diagram of Control Panel	4-18
4.14	Control Panel Layout	4-19
5.1	Interface Signals and Major Units	5-3
5.2	Relationship of Internal and I/O Bus	5-5
5.3	Clock is Inhibited during Program Channel Transfer	5-7
5.4	No Data Transfer so Timeout Occurs	5-9
5.5	Interrupt Received Routine	5-10
5.6	Data Save Microinstruction Routine	5-11
5.7	Planet Block Diagram	5-12
5.8	Spalu Block Diagram	5-15
5.9	Spalu Shift Connections for 16 Bit Configuration	5-17
5.10	Spalu Carry Interconnections for 16 Bit Configuration	5-17
5.11	Right Shifts	5-18
5.12	Left Shifts	5-18
5.13	Switch-On and Execute IPL Instruction Routine	5-19
5.14	Full Control Panel to CPU Handshaking Routine	5-20
5.15	Initial Program Loading	5-23
5.16	Execute LKM -- CPU goes to System Mode	5-26
5.17	Execute RTN -- CPU returns to User Mode	5-26
5.18	Power Failure and Automatic Restart	5-28
5.19	Timing -- Power Failure / Automatic Restart	5-28
5.20	ROM Address Timing	5-33
5.21	ROM Addressing Routine	5-34
5.22	Addressing ROM Bits 24-39	5-37
5.23a	SPALU, Data In/Out, IPL Logic Diagram	5-39
5.23b	Planet and ROM Control Logic Diagram	5-40
5.23c	P851 Clock, Bus Control, Timeout Logic Diagram	5-41
5.23d	Interrupt Logic Diagram	5-42
5.24	P851 CPU Component Layout	5-43
5.25	Microinstruction Routines	5-55
	1 Load Constant	5-63
	2 Load Constant to P	5-63
	Absolute Branch with Constant	5-63
	3 Addition Constant	5-63
	4 Addition Constant to P	5-63
	5 Enable / Inhibit Interrupts	5-64
	Link to Monitor	5-64
	Reset Internal Interrupt	5-64
	6 Subtract Register from Register	5-69
	Negate Register	5-69
	7 Logical Shifts -- Double, Single, Left and Right	5-68
	Circular Shifts -- Double, Single, Left and Right	5-68
	8 Arithmetic and Normalise Shifts -- D, S, L and R	5-65/66
	9 Arithmetic and Normalise Shifts -- D, S, L and R	5-65/66
	10 Output from Register (OTR)	5-69
	11 Input to Register (INR)	5-69
	Send Status (SST)	5-69
	Test Status (TST)	5-69
	12 Write External Register (WER)	5-69
	13 Read External Register (RER)	5-69
	14 RT1.OR2	5-62
	15 See Routine 8	
	16 Button Routines	5-57
	17 HLT.INST; Idle 1, Idle 2	5-56
	18 Subtract Constant from P	5-63
	Subtract Constant	5-63
	Logical AND with Constant	5-63
	Logical OR with Constant	5-63
	Exclusive OR with Constant	5-63
	19 Exchange Character Register to Register	5-70
	20 Call Function with Register	5-70
	21 Execute Register	5-72/73
	22 Ones Complement	5-70
	23 Addressing Routines	5-61/62
	24 Load T3B Type Instructions	5-74
	25 Increment Memory	5-75
	26 Twos Complement	5-75
	27 Halt	5-64
	Relative Forward Conditional Branch	5-64
	Relative Backward Conditional Branch	5-64
	28 Load Register with Register	5-69
	Absolute Conditional Branch to Register	5-69
	Addition Register with Register	5-69
	Logical OR Register with Register	5-69
	Test Not Mask	5-69
	Exclusive OR Register with Register	5-69
	29 Load Register with P	5-70
	Addition Register with P	5-70
	Subtract Register from P	5-70
	Compare Word-Register to P	5-70
	Call Function with P	5-70
	Compare Word-Register to Register	5-70
	30 Call Function	5-71
	31 Store T3B Type Instructions in P	5-74
	Load P	5-74
	Load Register	5-74
	Store Register	5-74
	Store P	5-74
	32 Addressing	5-61
	33 Store Without CR Change	5-62
	34 Store With CR Change	5-62
	35 Addressing Routine	5-61
	36 Addressing Routine	5-61
	37 Addressing Routine	5-61

Figure		Page
38	Addressing Routine	5-61
39	Store T3B Type Instructions	5-74
40	Load Character	5-76
	Store Character	5-76
41	Compare Word	5-76
	Compare Character	5-76
42	Return	5-71
43	Return with A15	5-71
44	Ones Complement	5-76
45	Multiple Load Constant	5-77
46	Multiple Store	5-77
	Multiple Load	5-77
47	Absolute Conditional Branch	5-75
	Addition	5-75
	Subtract	5-75
	Logical AND	5-75
	Logical OR	5-75
	Exclusive OR	5-75
	Clear Memory	5-75
48	Multiple Load/Store T3B Type	5-78
49	See Routine 48	
50	Double Add	5-79
	Double Add with Constant	5-79
51	Double Add Register to Register	5-81
52	Double Subtract	5-80
	Double Subtract with Constant	5-80
53	Double Subtract Register with Register	5-81
54	Multiply	5-82
	Multiply Register with Register	5-82
55	Compare Word with P	5-76
56	Divide	5-83/84/85
57	See Routine 56	
58	Control Panel Handshaking Routine	5-56
59	Instruction Step Button Routine	5-57
60	Read Memory Button Routine	5-57
61	Load Memory Button Routine	5-57
62	Master Clear Button Routine	5-57
63	Fetch 1, Fetch 2	5-56
64	Trap Routine	5-58
65	Power Fail	5-58
66	Automatic Restart	5-58
67	Interrupt Data Save	5-59
68	Control Panel Interrupt Button Routine	5-57
69	Initial Program Loader	5-60
70	Read Program Status Button Routine	5-57
71	Diagnostics -- Spalu Flags and Bus Tests	5-86
72	Diagnostics -- Condition Register Tests	5-88
73	Diagnostics -- Control Unit Test	5-89
74	Diagnostics -- Memory Tests	5-90/91/9
75	Diagnostics -- Memory Tests Continued	
76	Diagnostics -- Memory Tests Continued	
77	Call Function with A15	5-71
78	Multiply Register with Register	5-82
79	Divide Register with Register	5-83
6.1	Fault Finding Flow Chart	6-2
7.1	Interface Signals and Major Units of SCUZ	7-2
7.2	Timing for Input Mode	7-8
7.3	Timing for Output Mode	7-10
7.4	SCUZ Logic Diagram	7-13
7.5	SCUZ Component Layout	7-15
8.1	Memory Card Layout	8-2
8.2	Logic Blocks of Memory Card	8-2
8.3	Functional Block Diagram of a 4K Dynamic RAM Chip	8-3
8.4	Timing Diagram	8-5
8.5	Logic Diagram (a)	8-8
8.6	Logic Diagram (b)	8-9
8.7	4K Memory Card (RAMDZ4 — 16 Bits) Component Layout	8-10
8.8	8K Memory Card (RAMDZ8 — 16 Bits) Component Layout	8-12
9.1	Memory Card Layout	9-2
9.2	Logic Blocks of Memory Card	9-2
9.3	Block Diagram of Memory Chip	9-3
9.4	Read/Write Cycle Timing	9-4
9.5	Static RAM Logic Diagram	9-7
9.6	0.5K Memory Component Layout	9-8
9.7	1K Memory Component Layout	9-10
9.8	2K Memory Component Layout	9-12
10.1	Memory Card Layout	10-2
10.2	Functional Schematic Diagram of 16k - 32k RAM Memory	10-3
10.3	Functional Block Diagram of a 16k Dynamic RAM Chip	10-5
10.4	16k/32k Memory Card Linking Diagram	10-6
10.5	Cycle Timing Waveforms	10-7
10.6	Refresh Cycle Waveforms	10-8
10.7	Bus Timing Waveforms	10-9
10.8	Memory Logic Diagram	10-13
10.9	Memory Card Component Layout	10-15
Appendix A	Integrated Circuits	A-1

LIST OF TABLES

Table		Page
2.1	Configuration Possibilities	2-10
2.2	GPBS Signals	2-14
2.3	Types of Data Transfer	2-15
2.4	Back Panel Connector 1 Pinning	2-18
2.5	Back Panel Connector 2 Pinning	2-19
2.6	General Purpose Card Parts List	2-25
3.1	Mains Transformer Chassis Parts List	3-12
3.2	+5V Regulator Card Parts List	3-13
3.3	Memory Supply Card Parts List	3-14
3.4	Sequensor Card Parts List	3-15
4.1	PROM Contents	4-5
4.2	Connector Data	4-6
4.3	Mini Panel Components	4-8
4.4	ROM Contents	4-13
4.5	Back Panel Connector Data	4-15
4.6	Control Panel Interface Data	4-15
4.7	Component List of Inter Panel	4-17
4.8	Full Control Panel Parts List	4-19
5.1	General Purpose Bootstrap Contents	5-23
5.2	Dedicated Bootstrap Contents	5-24
5.3	P851 List of Instructions	5-31
5.4	RS Field Decodes	5-35
5.5	P851 CPU Parts List	5-44
5.6	ROM Contents	5-45
5.7	Meanings of Terms and Signal Names Used in the Text	5-93
6.1	Standard Test Programs	6-84
7.1	Sequensor 1 Contents	7-6
7.2	Sequensor 2 Contents	7-7
7.3	Sequensor 3 Contents	7-7
7.4	V24 Interface	7-12
7.5	Current Loop Interface-2 Wire	7-12
7.6	Current Loop Interface-4 Wire	7-12
7.7	U Links	7-14
7.8	SCUZ Parts List	7-14
8.1	Memory Module Addressing Links	8-4
8.2	Interface Connections	8-7
8.3	4K Memory Card (RAMDZ4 — 16 Bits) Parts List	8-11
8.4	8K Memory Card (RAMDZ8 — 16 Bits) Parts List	8-13
9.1	Interface Connections	9-6
9.2	0.5K Memory Parts List	9-9
9.3	1K Memory Parts List	9-11
9.4	2K Memory Parts List	9-13
10.1	Command Decoder Functions	10-10
10.2	Write/Read Command Decoding	10-10
10.3	Interface Connections of Card Socket	10-12
10.4	16k/32k Dynamic RAM Parts List	10-16

SECTION I

SYSTEM DESCRIPTION

1.1 GENERAL

The P851 mini-computer system is modular in design and may be used either by itself as a general purpose mini-computer or as part of a larger system. The following facilities are available with the P851:

- Set of 160 software instructions.
- Monitor control.
- Two data channels: Program, Input/Output Processor or Direct Memory Access.
- Real Time Clock interrupt: every 20mS for 50c/s, every 16.67mS for 60c/s supply.
- 16 Scratch Pad registers, 14 fully programmable.
- Power Failure Routine.
- Automatic Restart after power failure.
- Battery Back-up after power failure.
- 16 interrupts (4 internal).
- Initial Program Loader -- General Purpose Bootstrap, loads a user or system program into RAM from an I/O device.
- Initial Program Loader -- Dedicated Bootstrap, loads a user or system program from REPROM.
- Microdiagnostics.
- Trap Routine if an attempt is made to execute an invalid instruction.
- Timeout -- frees the Bus after 6.4μS if a reply has not been received by CPU.

Peripheral Devices include:

- Typewriter -- ASR 33, 10 characters/sec.
- Matrix Printer and Keyboard -- PER 3100, 50 characters/sec.
- Paper Tape Punch, 75 characters/sec.
- Paper Tape Reader, 333 characters/sec.

1.2 Eurocards

- Central Processor -- multiply, divide, double shift, add, subtract, IPL, 16 interrupts, Bus control, interfaces full or mini control panel, 1024 microprogram words, timeout, cycle time 640 nS.
- Input/Output Processor -- 300K words/sec, interfaces up to 8 devices, addresses up to 32k of memory, block transfers maximum 4k.
- Serial Control Unit -- synchronous or asynchronous, speeds 50-19,200 baud.
- Synchronous/Asynchronous Line Control Unit -- half or full duplex, speeds 110-9,600 baud.
- High Level Data Link Control Unit -- synchronous, half or full duplex, speed 100k bits.
- Paper Tape Reader/Punch Control Unit -- interfaces one punch and one reader device.
- Input/Output Translator -- enables P851M to interface P852,56,57M controllers.
- Programmable Real Time Clock -- either 4 standard rates 1,5,10,20mS or 2^{15} rates from 1mS to 32 seconds.
- Digital Clock -- only interfaces on Programmed Channel without interrupts.
- Static Memory -- 0.5k, 1k, 2k, 4k, or 8k 16-bit words, cycle time 0.75μS, access time 0.55μS max.
- Dynamic Memory -- 4k or 8k 16-bit words without parity, 18-bit words with parity, cycle time 0.660μS, access time 0.480μS.
- Reprogrammable Memory -- 1 to 4k 16-bit words, cycle time 0.75μS, access time 0.55μS max.
- General Purpose Card -- contains the basic logic for interfacing the GP Bus S and gives the customer the opportunity of designing his own Control Unit.
- Extender Cards -- available to give engineering test facilities.
- Floppy Disc CU -- transfer rate 250,000Hz (4μsec/bit), access time 260mS (average.

1.3 Cabinet

The P851 cards are housed on a Eurorack containing:

- Logic cards for processing data and controlling the system.
- Power Supply.
- Control Panel, either a Full Control Panel or Mini-Control Panel is available.

Two sizes of logic cards are used:

- 160x100mm. Five card positions of this size are available and will normally contain --

  Inter-Panel or Mini-Control Panel card,

  Sequence card,

  Battery Pack card,

  Memory Power Supply card,

  Regulator card.

- 160x233.4mm. Thirteen card positions are available as follows --

  Position 1. Central Processor card must be in this position.
  2. IOP must be in this position. (If not fitted a Memory card may be inserted.
  3. ⎫ Memory and Control Unit Cards
  4. ⎪
  ⎪
  12. ⎪
  13. Input/Output Translator Card must be in this position if present. ⎭

1.4 Control Panels

The P851 may be supplied with either a Full Control Panel or Mini Control Panel. The Mini Panel is mounted on a Eurocard and plugs directly into the General Purpose Bus S (Figure 1-1). When the Full Control Panel is supplied instead of the Mini Panel then an Inter-Panel Card interfaces it to the GP Bus S. For details of both control panels see the Control Panel section and for the basic differences in the interconnections see Figure 1-2.

1.5 DATA TRANSFERS

All data transfers within the P851 system use either the Master/Slave principle of transfer or transfer data between two slaves under the control of a master. All data transfers are via the General Purpose Bus S and take the form of either a 16-bit word transfer or an 8-bit character transfer. The basic transfer channel is the Programmed Channel which uses the data path between CPU and Control Unit. One Input/Output Processor Channel may be connected to the system as an option and can use the same Control Units as Programmed Channel. Up to eight subchannels may be connected to the Input/Output Processor and be used simultaneously. Priority is given initially to the Input/Output Processor as the Master and then allocated to one of the subchannels. Priority of the subchannels is decided at system installation time.

1.6 Programmed Channel (Figure 1-3)

The Programmed Channel is the basic transfer channel and is standard with all systems. Apart from providing an exchange path between a Control Unit and CPU it also enables the initialisation between CPU and Input/Output Processor to take place. Data is transferred by input/output instructions and external register instructions in a program. Each word or character transferred requires a separate instruction. Apart from the instructions used to carry out the exchange, Input/Output instructions may be used to start and stop a device, and TST and SST to check the status of a device. Two modes of operation, Wait and Interrupt Mode, may be used with Programmed Channel.

1.7 Wait Mode. This is the simplest but slowest form of transfer and so is only used normally for Stand Alone Programs. Each word or character is exchanged separately and the complete program is held up in a waiting loop between individual exchanges. In this mode the maximum transfer rate is either the time to execute the program loop or the time taken by the device to execute a single exchange whichever is slowest.

1.8 Interrupt Mode. Each word or character is exchanged separately and each instruction executed forms part of an interrupt routine. When a Control Unit is ready to exchange a character it raises an interrupt and the main program is stopped while the new exchange is initiated by the interrupt routine. On completion of this routine the main program is restarted. To enable the operation of a Control Unit via the Programmed Channel the following instructions are available:

- CIO Start
- CIO Stop
- INR Input to Register
- OTR Output to Register
- SST Sense Status
- TST Test Status

The responses given to these commands are set in the Condition Register of the Central Processor. If certain conditions exist after the execution of one of these instructions then this gives software the facility to jump into another routine using a Branch instruction.

1.9 Input/Output Processor Channel (Figure 1-4)

The Input/Output Processor Channel is mounted on its own card and plugs directly into the General Purpose Bus. This channel provides a fast method of block transfer between the system memory and up to 8 Input/Output Control Units at a transfer rate of up to 300k words per second. These subchannels to the 8 devices serve the same purpose as the interrupt routine of the Programmed Channel. A data path is provided between the device Control Unit and Memory via the Bus. The normal sequencing and states of the device control units apply, but the interrupt is replaced by a Break signal wired directly to the Input/Output Processor priority system. The priority system initiates each hardware exchange sequence.

1.10 IOP Priority System. Each control unit connected to the IOP uses only one Break line which enables the IOP to allocate priority to each control unit. The Break lines are connected via a priority system. Subchannel 0 has the

EURORACK FRONT VIEW

For details of Mini Panel see Section IV

For details of Power Supply see Section III

Notes: When the Full Control Panel is fitted the Mini Panel Card is replaced by the Inter Panel Card

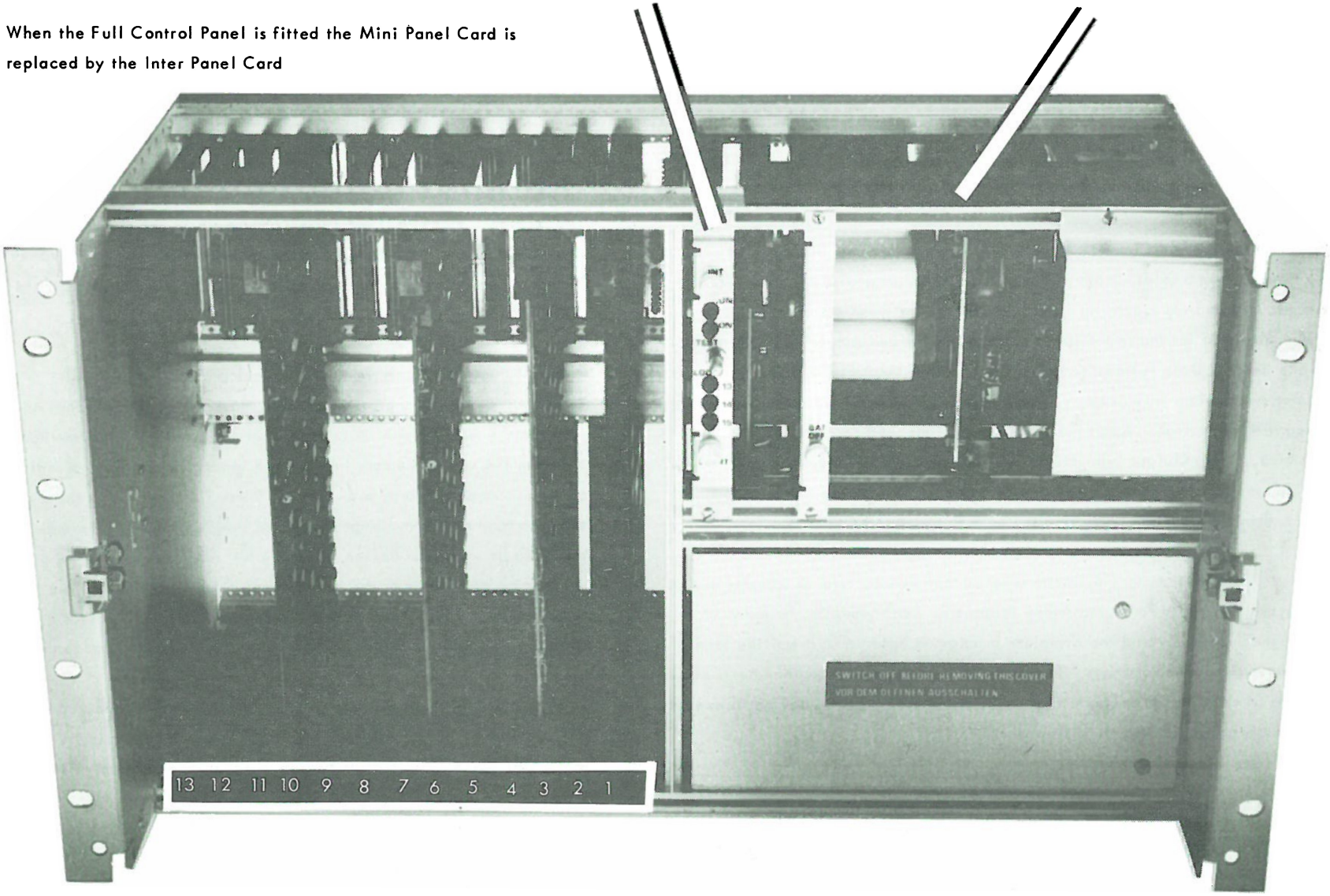


Figure 1-1 Eurorack with Mini Control Panel

EURORACK REAR VIEW

Notes: Details of No. 1 and No.2 connectors are given in Section II

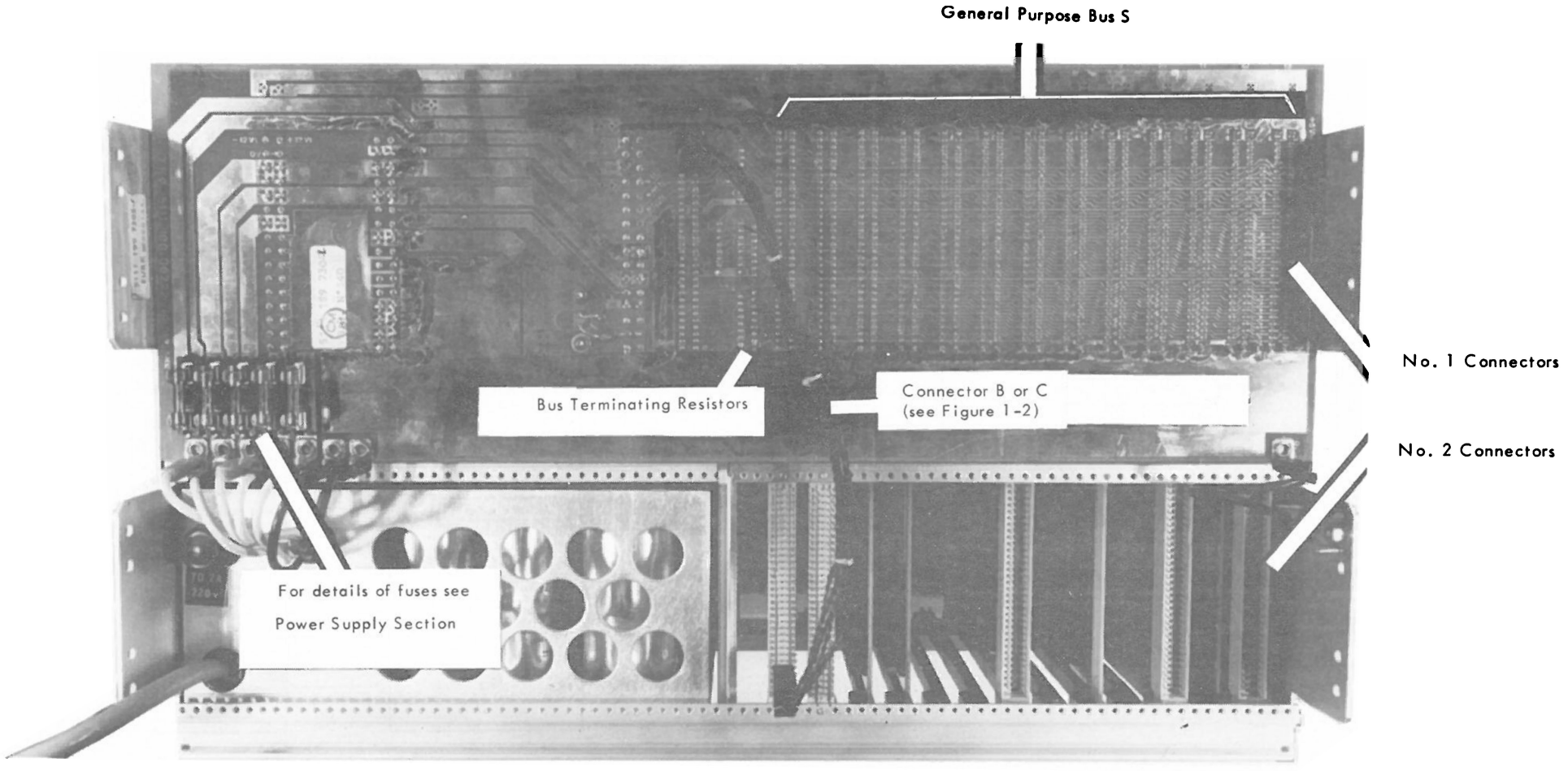
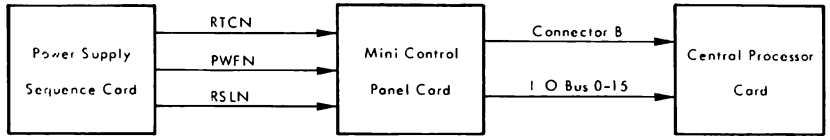
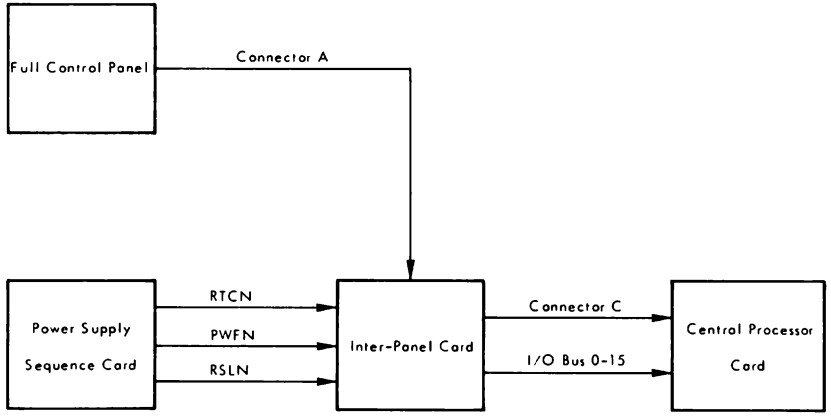


Figure 1-1 Eurorack with Mini Control Panel

Mini Panel



Full Panel



NOTES: Signals RTCN, PWFN, RSLN and I/O Bus 0-15 are printed circuit connections via the Eurorack back panel

For Connector A details see Table 4-6
For Connector B details see Table 4-2
For Connector C details see Table 4-5

Figure 1-2 Interconnections for Both Control Panels

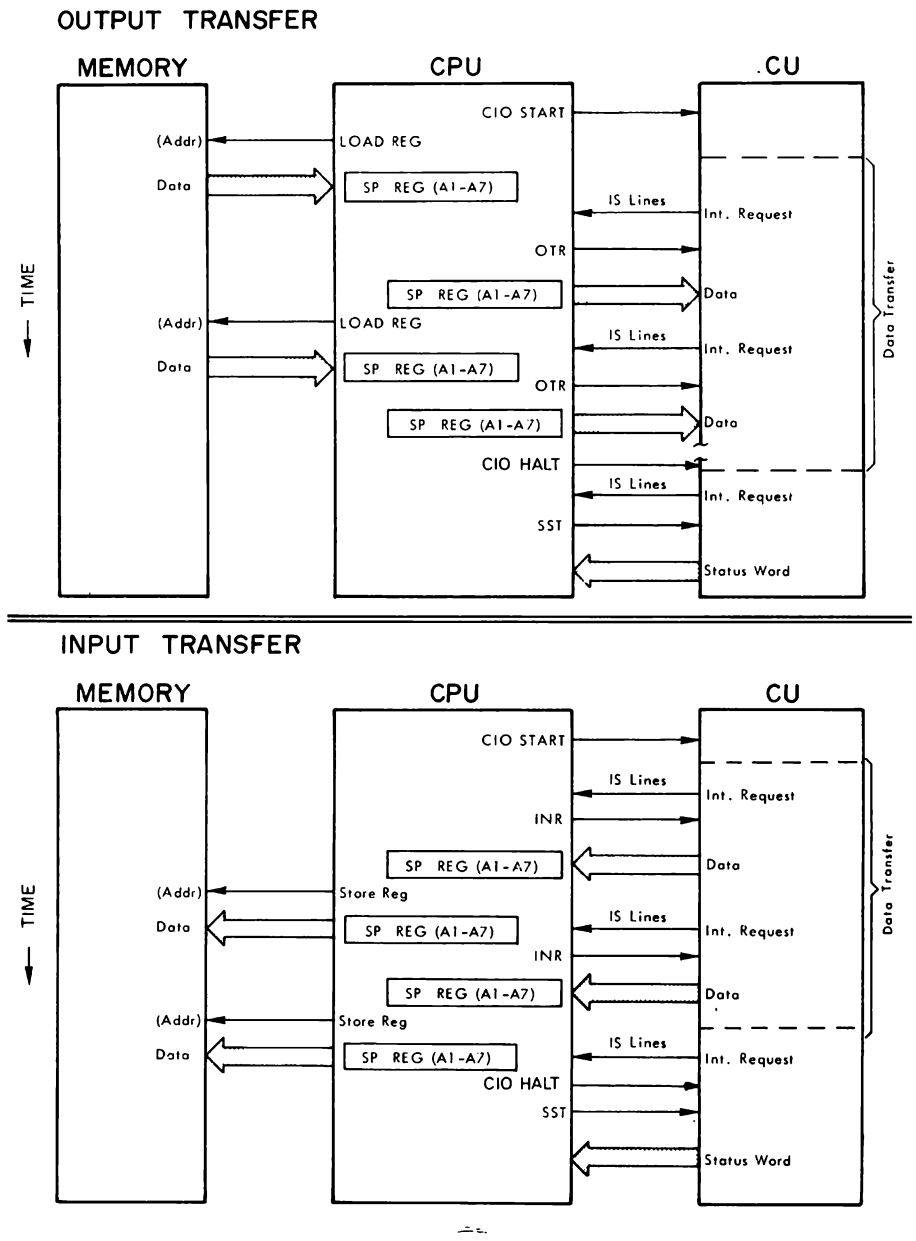
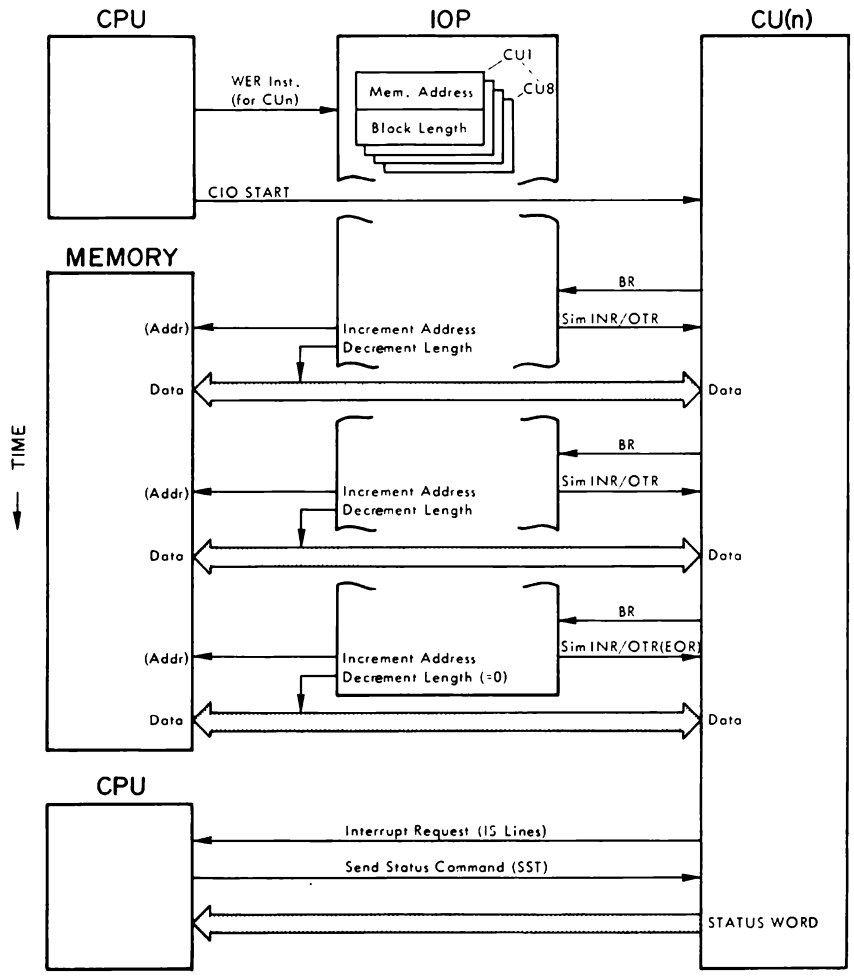


Figure 1-3 Programmed Channel Transfers



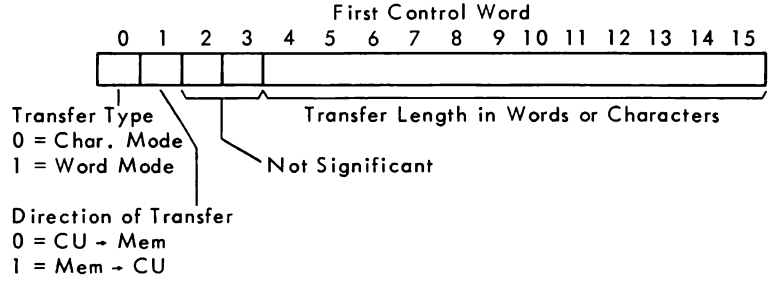
Note -- This diagram shows the sequence for transferring a block of data between memory and a single CU, (n). The IOP can multiplex up to eight CUs; the operation shown is thus duplicated for each CU.

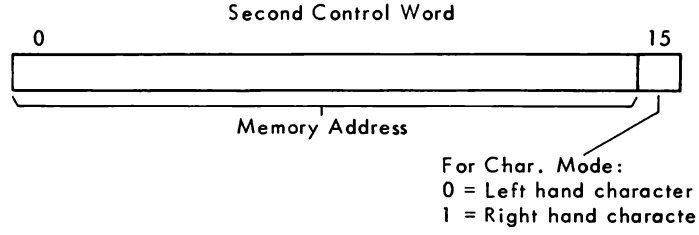
Figure 1-4 Multiplexed I/O Processor Channel Transfer

highest priority and subchannel 7 the lowest. IOP has 16 working registers, 2 for each subchannel. These are used to contain control words with the following information:

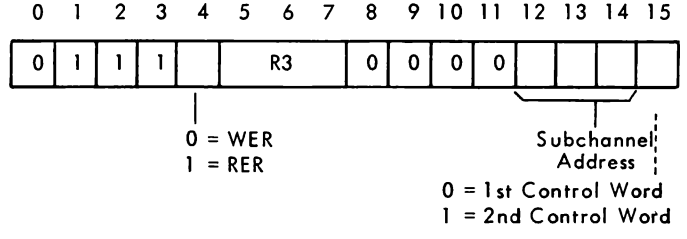
- Character or Word transfer.
- Direction of transfer.
- Length of a block to be transferred (max length 4k).
- Memory start address.

IOP is addressed by instruction WER which transfers the following parameters:





Each control word is loaded in the next subchannel register by a WER instruction with the following format:



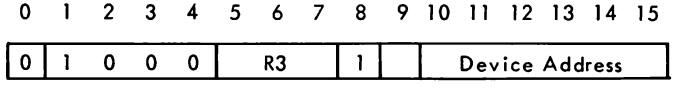
1.11 IOP Uses Three Data Paths. Three separate control Paths are used during an IOP transfer.

- CPU to IOP. CPU is master of the exchange which is executed using RER or WER instructions. WER sets up the IOP and RER reads the contents of the working registers during an End or an Error Routine.
- CPU to Control Unit. CPU is master of the exchange and can start, stop or request the status of a device connected to the IOP. These instructions are sent via the Programmed Channel.
- IOP to Memory and Control Unit. IOP is master of the exchange which is between Memory and a Control Unit. This type of exchange is initiated after the reception of a Break signal and after the IOP has been allocated a Bus cycle by the CPU. When the last exchange for a Control Unit has taken place an interrupt is sent from Control Unit to CPU.

1.12 P851 Control Instructions

The control instructions available for P851 enable the starting and stopping of a peripheral device, testing for error or inoperable conditions and transferring data between CPU and external registers. This information must be considered as of a general nature as not all Control Units use the facilities available.

1.13 Start and Halt Instructions. The Command Input/Output instruction is interpreted by the P851 system as follows:

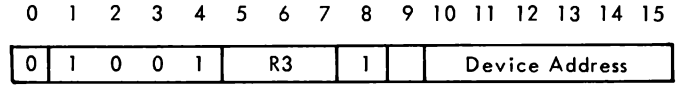


Bit 9 = 1 Start an I/O operation on a peripheral device.
Bit 9 = 0 Stop a data transfer or reset the state of a Device Control Unit.
R3 = Scratch Pad Address A1-A7.

During the execution of the instruction the DA field is sent to the Device Controller via the I/O Bus. In some cases the contents of the 16-bit register specified by R3 may also be sent to provide further information. A CIO Start will not be accepted if the device address is unknown or if the device is busy. CIO Halt is always accepted.

1.14 Status Instructions

There are 2 status instructions, SST and TST, the difference between the two being specified by bit 9. The instruction is interpreted as follows:

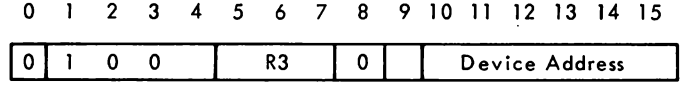


Bit 9 = 1 (SST) Send status of the device to the CPU register specified by R3. When status is received it means the following:
Bit 15 = 1 Not Operable
Bit 14 = 1 Throughput Error
Bit 13 = 1 Parity Error
Bit 12 = 1 Incorrect Length
Bit 10 = 1 Tape Low (TRPZ)

Bit 9 = 0 (TST) Test status of the device by sending the relevant data to the register specified by R3.
Bit 15 = 1 Device Busy
Bit 14 = 1 Previous Command was accepted (TRPZ)

1.15 Data Exchange Instructions

There are 2 exchange instructions, INR and OTR, the difference between them being specified by bit 4. The instruction is interpreted as follows:



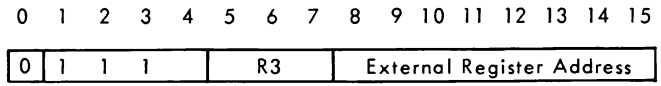
Bit 4 = 1 (INR) Data is input from the Control Unit specified in the DA field to the CPU register specified by R3.

Bit 4 = 0 (OTR) Data is output from the CPU register specified by R3 to the Control Unit specified in the DA field.

Note that bit 9 of the instruction field may be used to specify a particular function e.g. Binary or ASCII code.

1.16 External Register Instructions

There are two external register instructions, WER and RER, the difference between them being specified by bit 4. The instruction is interpreted as follows:



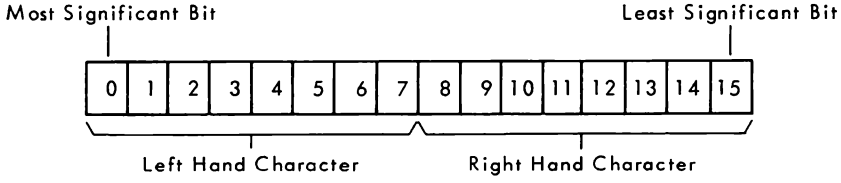
Bit 4 = 1 Read External Register (RER)

Bit 4 = 0 Write External Register (WER)

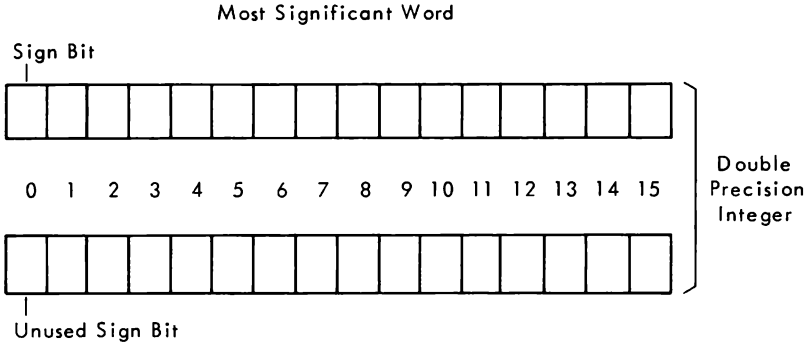
For P851 these instructions are associated with IOP Channel. The WER takes the content of the CPU Scratch Pad register specified by R3 and transfers it to the IOP Scratch Pad address specified by bits 8 to 15. The RER reads the content of the IOP Scratch Pad register and transfers it to the CPU Scratch Pad to check on the progress of a transfer.

1.17 Data Formats

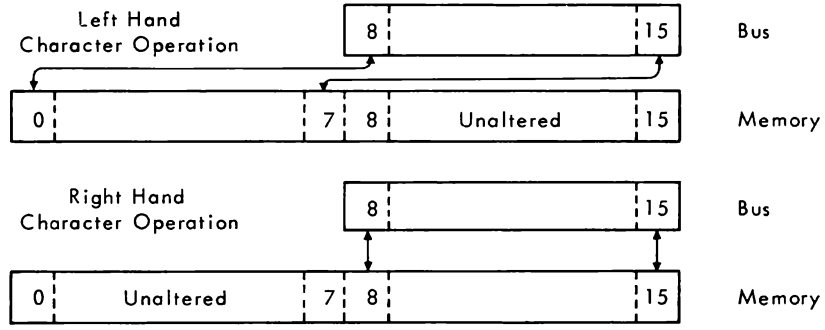
The basic data format within the system is shown below. The bit positions are numbered from 0-15 where bit 0 is the most significant bit and bit 15 the least significant bit. Bit 0 is the sign bit. When at logic level 0 it indicates positive data and when at 1 negative data. The data word may be divided into two characters and can be used independantly by certain instructions.



A double format is used for extended arithmetic operations. The sign bit of the least significant word is not used and is usually reset to zero leaving 31 bits to represent data.



All memories can operate in word or character mode. Character mode operation enables the content of the least significant input or output lines to be either read or written by the left or right-hand character of the addressed memory location. In all operations of this type the unused character is left unaltered.

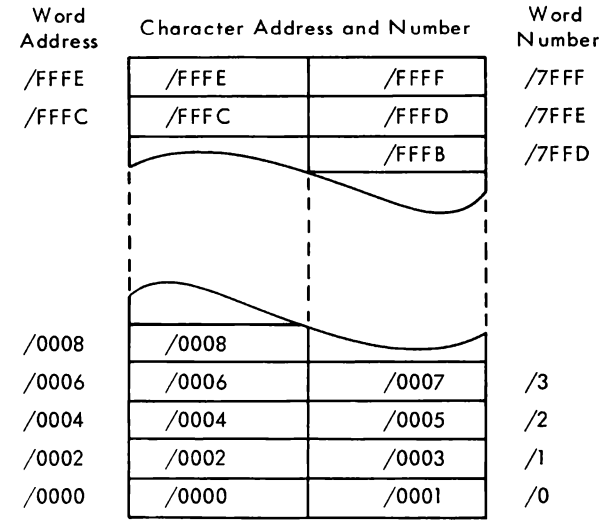


1.18 Memory Addressing

Depending on the type of instruction or the type of operation the memory can be addressed in word or character mode. Bits 0-14 address memory in word mode, bit 15 is unused. In character mode all 16 address bits are used. Bits 0-14 address the location and bit 15 selects the character either left or right.

Bit 15 = 0 addresses left-hand character

Bit 15 = 1 addresses right-hand character



SECTION II

INSTALLATION, INTERFACE and GENERAL PURPOSE CARD

2.1 This section contains information to enable the engineer to install and interface the P851 system. For customers who want to design their own Control Units a General Purpose Card is available already containing some logic to simplify the design task for the customer. If certain configurations require specialist assistance then contact the service office quoted in the Service Contract or the nearest Philips office.

2.2 SITE PREPARATION

Extensive site preparation is not normally required for a P851 system. In the case of simple configurations or where the Eurorack is part of a much larger system no site preparation is necessary at all. Therefore, the following recommendations (with the exception of the Electrical Requirements) need not necessarily apply to your system.

2.3 Environmental Conditions

The CPU can operate through a wide range of environmental conditions so the need for air conditioning will depend on the peripherals used by the system. The P851 CPU and its peripherals will normally be located indoors in an office/factory environment where the air is not contaminated by industrial processes. The conditions for storage (with or without packaging), transport, and operating are:

- Storage and Transport.

Temperature Range	-30°C to +70°C
Dynamic Change	1°C/mm
Relative Humidity	$\leqslant$90%
Air Pressure	600mb to 1100mb

- Operating (CPU).

Temperature Range	0°C to +50°C
Dynamic Change	1°C/mm
Relative Humidity	90%
Air Pressure	600mb to 1100mb

For most system applications, especially where operating personnel are in the same environment, the following conditions are recommended:

Temperature Range	+18°C to 24°C
Relative Humidity	50% to 60%
Air Pressure	860mb to 1060mb

2.4 Electrical Requirements

A single phase a.c. supply complying with the following specification should be provided for the exclusive use of the computer system. It should be either a 3 wire single phase, neutral and earth, or two phases and earth; its current carrying capacity must be adequate for the total power requirements of both the CPU and the peripherals. The requirements are:

Voltage	115V, 220V or 240V r.m.s. ±10%
Frequency	50±2Hz or 60±3Hz
Voltage Transients	1.5kV peak amplitude for a duration not exceeding 10 microseconds between half peak points at a repetition frequency of 3 to 10Hz for not more than 10 minutes
Transient State	The time between the voltage falling from nominal to zero and then returning to nominal must be not greater than 0.5 secs
Power Interrupts	Less than one every 10 seconds for a duration not exceeding 10mS.

If the mains supply is subject to frequent interruptions or failures it is advisable to install an alternative supply source, such as a rotary motor/generator.

2.5 INSTALLATION

Prior to packing and dispatch to the site, the equipment will have had an overall system check during which all the necessary settings and adjustments will have been made. When installing the equipment reference must be made to the Configuration Sheets supplied with the equipment which give details of special connections, board links etc. After unpacking the various items check that nothing is missing or damaged, then assemble the racks and install the various units in position making all the required interconnections. A few preliminary checks are then made and the system is ready for commissioning with the test programs devised for your system configuration. The following paragraphs should be used as a supplement to the Configuration Sheets or as an aid to enlarging the present system at a later date.

2.6 Eurorack

Figure 2-1 shows the Eurorack which contains the power supply, CPU, Memory, CU cards, etc. The rack is divided vertically into two equal parts, the lefthand half has thirteen vertical slots to hold the double format Eurocards of the the CPU, Memory etc., and the righthand half is divided into two horizontal compartments which contain the power supply. The lower of the two compartments holds the mains transformer, rectifiers and reservoir capacitors; the upper half holds the three power supply control cards and the Mini Panel or Inter Panel card (all single format Eurocards). Figure 2-2 shows layout details of the Eurorack including the two dedicated slots for the CPU and IOP cards. The positions of the other cards will be given in the Configuration Sheets. On the rear upper half of the rack the printed circuit board, which provides the Simplified General Purpose Bus (GP-Bus-S), is mounted; the power supply connections to the back panel connectors which are mounted internally in the rack. The Eurorack is designed to be mounted in a standard 19" rack and it is held in position by four screws.

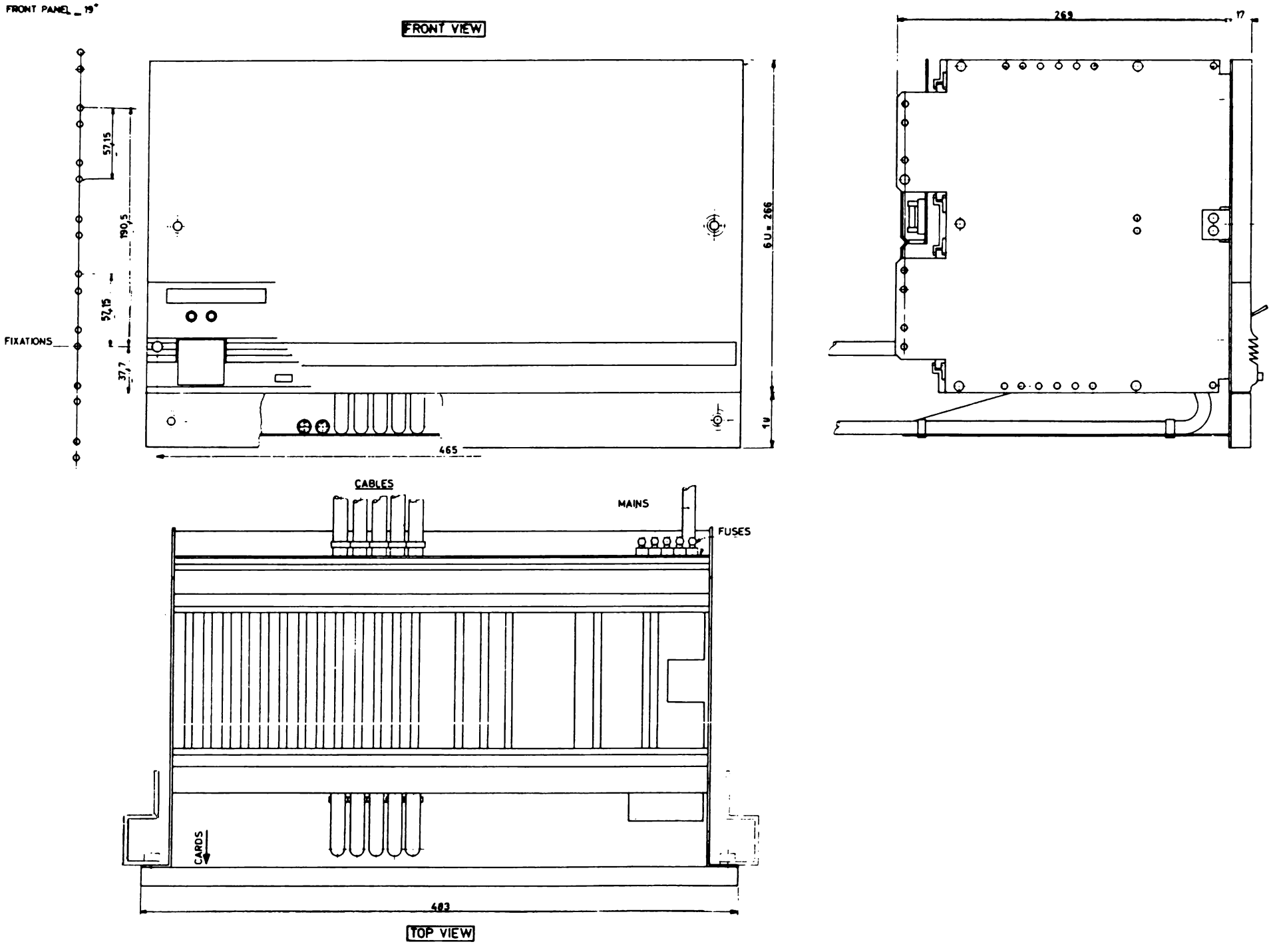


Figure 2-1 Eurorack Installation Data

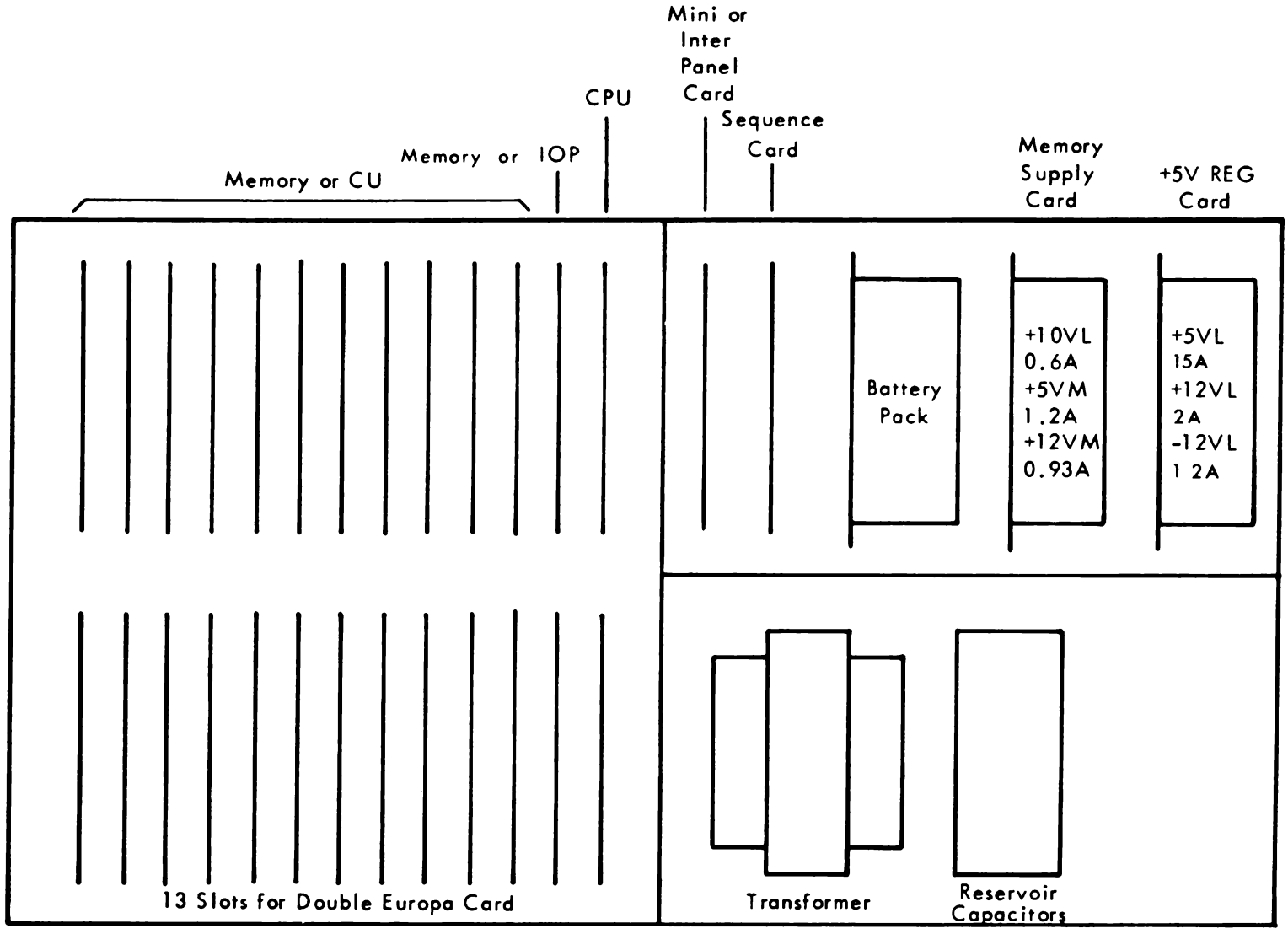


Figure 2-2 Eurorack Layout Details

2.7 Air Ventilation Units

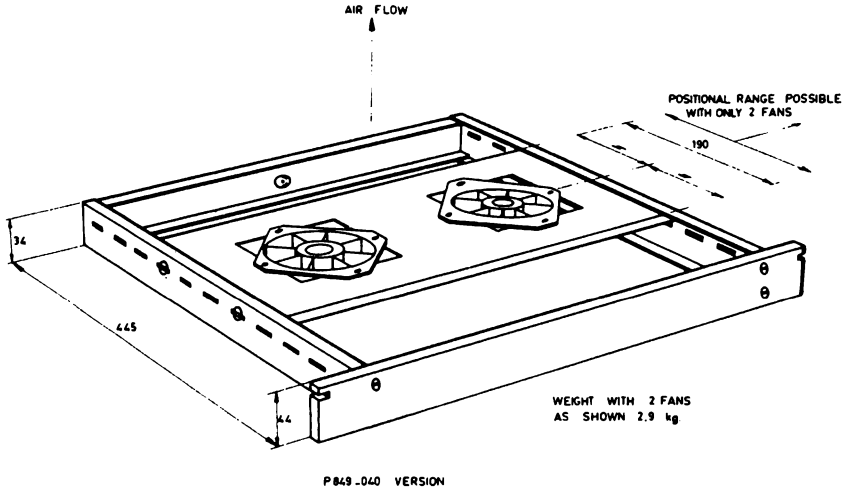
Convection cooling is sufficient if the total heat dissipation of the equipment in the rack is less than 700 kcal/h (800 watts). Above this level forced air circulation by one of the following ventilation units is necessary. Both units are shown in Figure 2-3 and are similar except for the number of fans. The ventilator units need 1U rack space and are:

- P849-040 -- intended for heat dissipation levels of between 700 and 1030 kcal/h (800 to 1200 watts) and has two fan units.
- P849-041 -- intended for heat dissipation levels of between 1030 and 1300 kcal/h (1200 to 1500 watts) and has four fan units.

2.8 Power Distribution Panels

Two types of power distribution panel are available to suit different power consumption levels. Both panels are 2U high and are:

- P849-042 -- (shown in Figures 2-4 and 2-5) a power distribution panel with mains filter, mains isolation switch, neon indicator, fuse, and terminal block with a current capacity of 10 amperes.
- P849-043 -- (shown in Figures 2-6 and 2-7) a power distribution panel with the same features as for the P849-042 except that the current carrying capacity is 25 amperes.



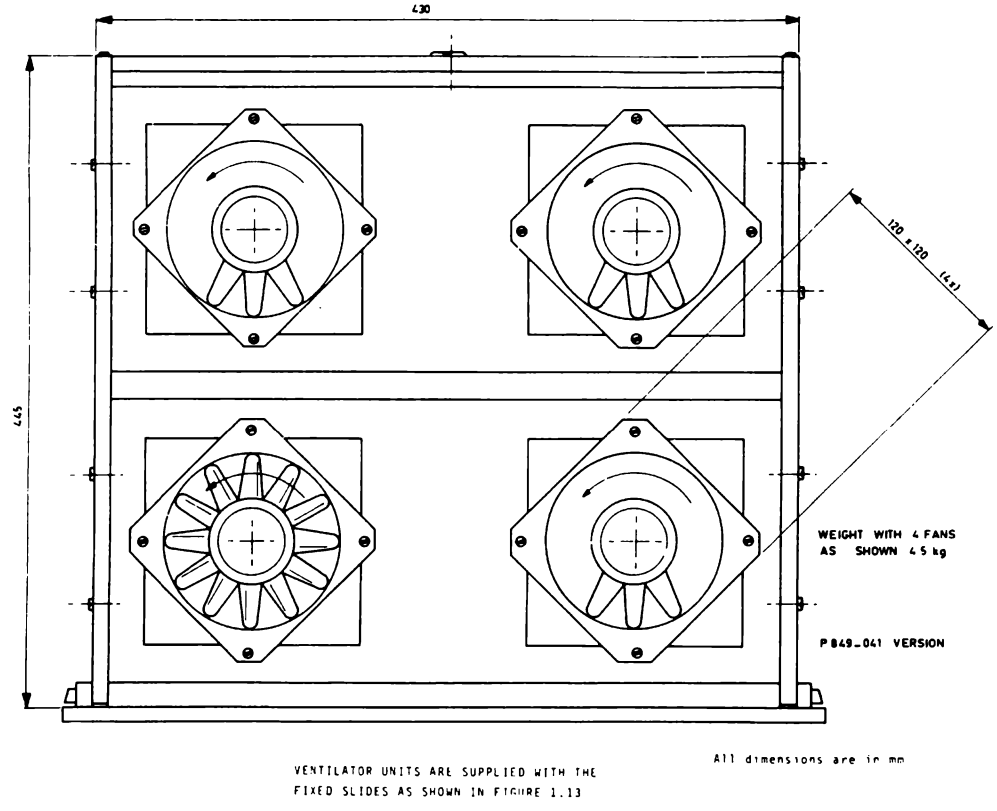


Figure 2-3 Ventilator Units

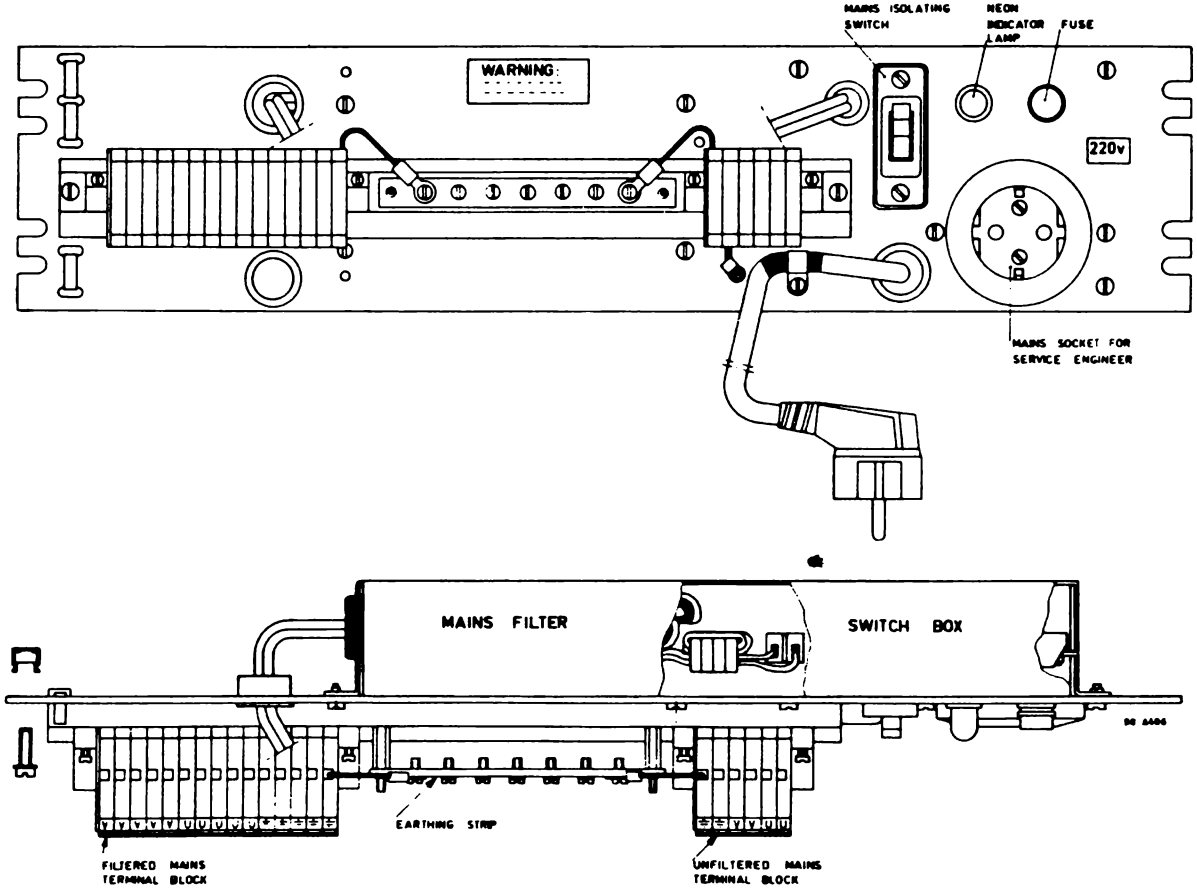


Figure 2-4 10 Ampere Distribution Panel

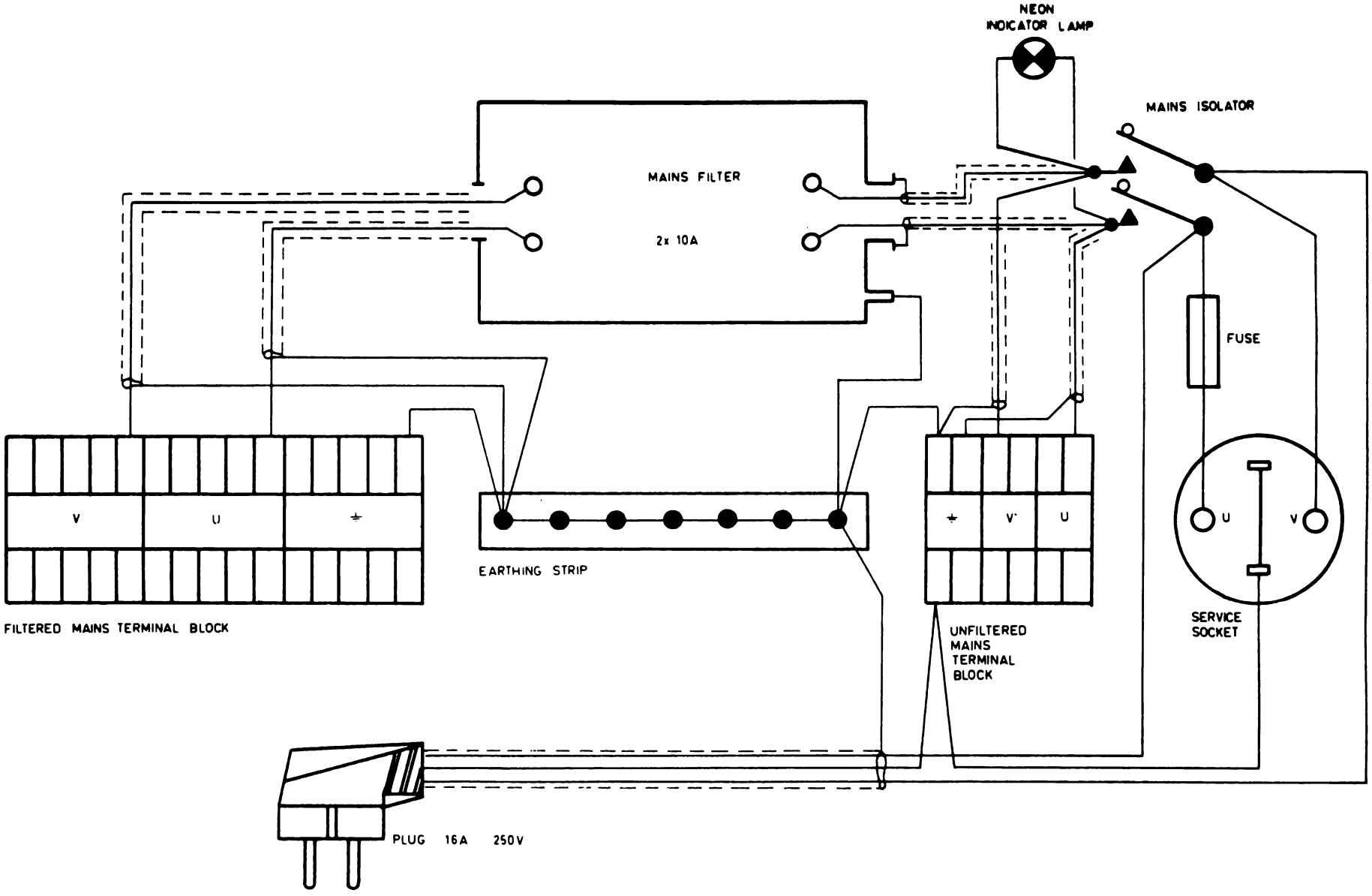


Figure 2-5 10 Ampere Panel Circuit

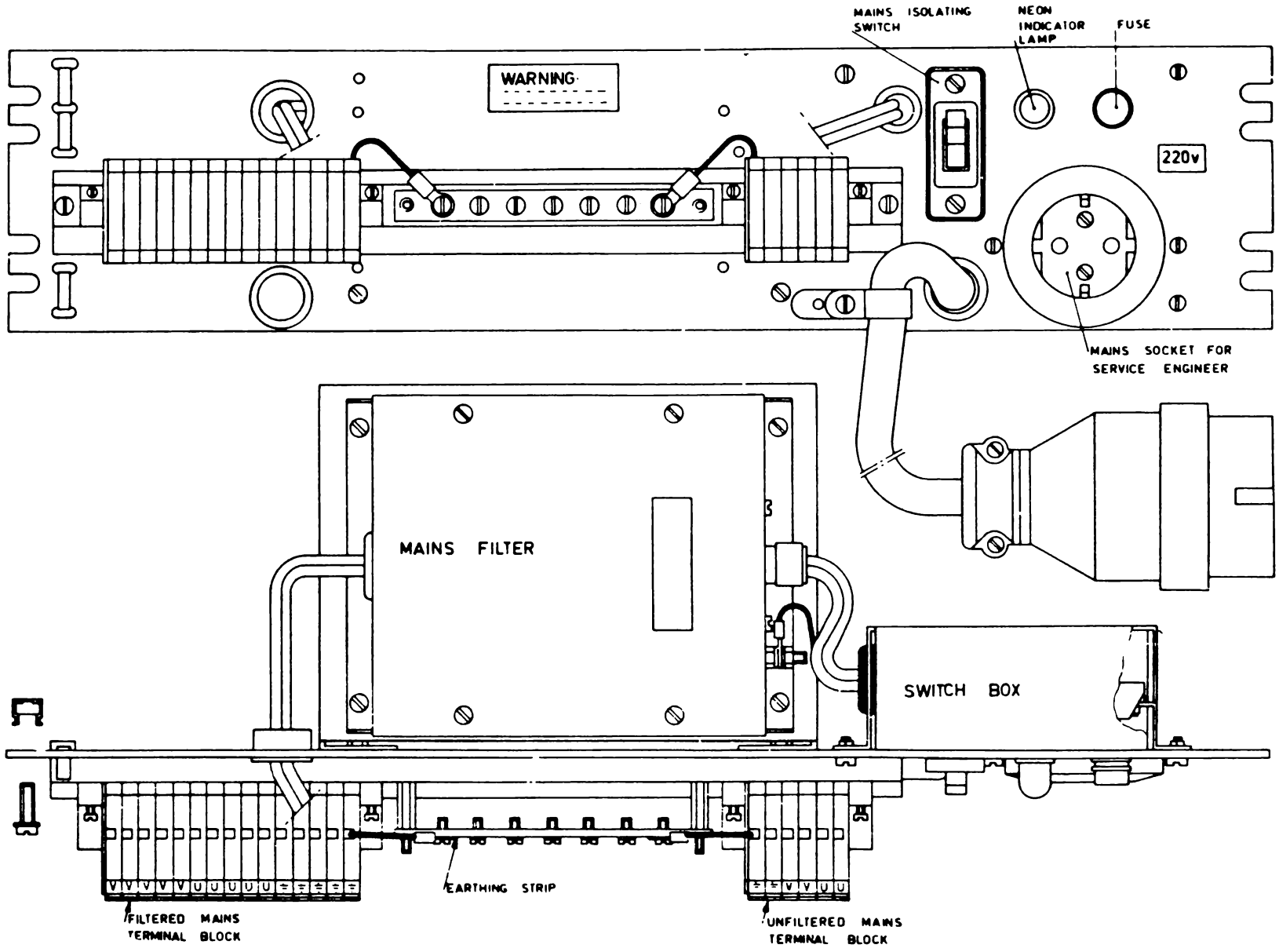


Figure 2-6 25 Ampere Distribution Panel

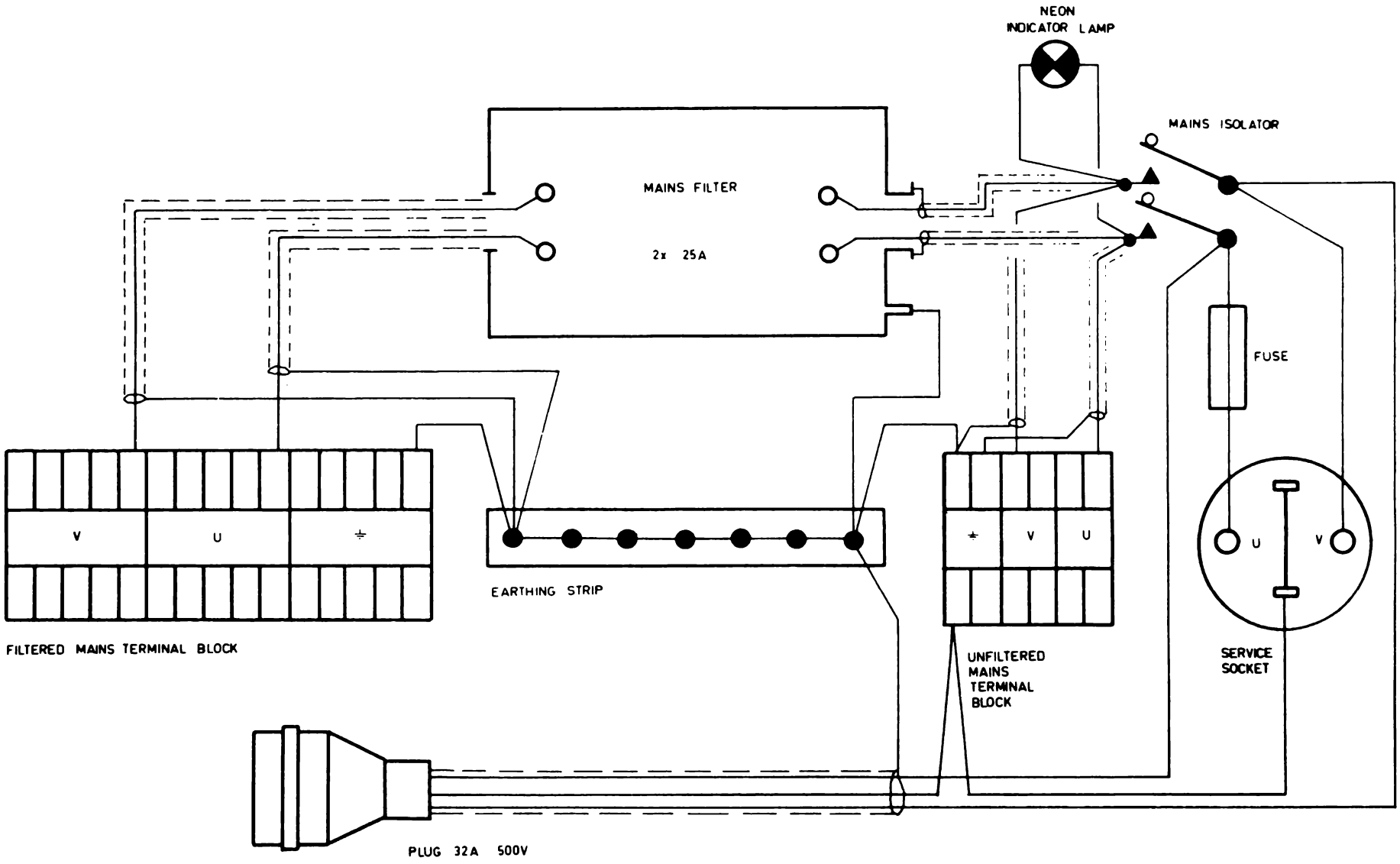


Figure 2-7 25 Ampere Panel Circuit

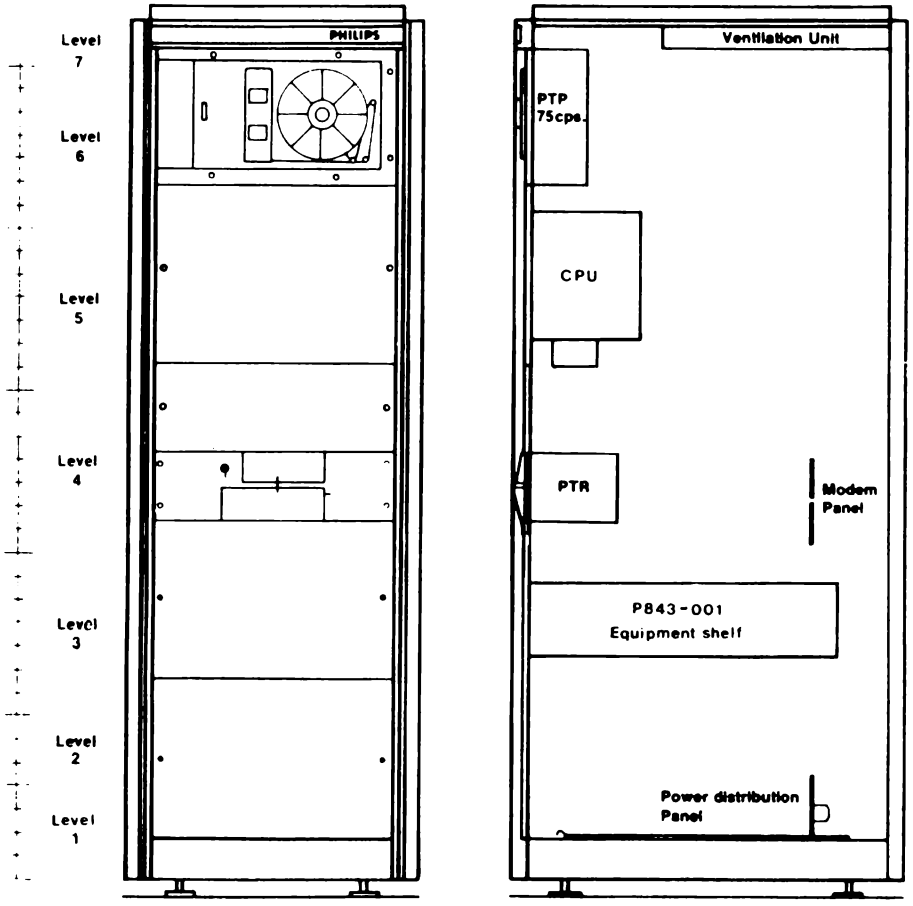
2.9 Configuration Guide

For a customer planning a single or multi-rack layout the following should be taken into account.

- Single Rack Layout — Care should be taken to ensure a uniform appearance, operational access to the cabinet and peripherals, length of I/O signal cables and weight distribution of the different levels. The rack is divided into seven imaginary levels and except when the Tape Punch is mounted between levels 1 and 2 all equipment must be confined to the levels shown in Table 2-1. Any space left between the levels must be filled with an appropriately sized blanking panel. When complete systems are sent direct from the factory the 7 Unit (7U) blank panels are fitted wherever possible but if separate items are delivered they may come with the appropriate blank panel. Two possible configurations are shown in Figure 2-8.
- Multi-Rack Layout — To use the P851 with other systems there are restrictions which must be considered. The GP Bus S is not designed to be extended out of the basic cabinet except when using the Input/Output Translator (IOT) which connects Control Units of the P852/6/7. When using the IOT the IOT card must be fitted into position 13 of the Eurorack, a maximum of 7 external racks may be connected, cable length between 2 racks must not be less than 2 metres and must not exceed 15 metres. Total cable lengths for all extensions together must not exceed 15 meters. For program Channel the cable connecting Breaks to and from IOT should be a twisted pair, and the length must not exceed the length of Bus connecting that same rack. IOT only interfaces Control Units (not memories) which operate as slaves in the P851 system.

Table 2-1 Configuration Possibilities

Location	Item	Units Required Front	Rear
Level 7 1U	Nameplate	1	-
	Ventilator Unit	-	1
Level 6 7U	P803-001 Tape Punch	6	-
	P833-152 Cassette Shelf	4	4
	P843-001 Equipment Shelf	3	3
Level 5 7U	P851 Eurorack	7	-
	P801-001 or P802-001 Punched Tape Reader	3	-
	P803-001 Tape Punch	6	-
	P843-001 Equipment Shelf	3	-
Level 4 7U	P851 Eurorack	7	-
	P801-001 or P802-001 Punched Tape Reader	3	-
	P803-001 Tape Punch	6	-
	P824-002 Moving Head or P830-10 Floppy Disc	6	6
	P843-001 Equipment Shelf	3	3
Level 3 7U	P803-001 Tape Punch	6	-
	P824-002 Moving Head or P830-10 Floppy Disc	6	6
	P843-001 Equipment Shelf	3	3
Level 2 3U	P843-001 Equipment Shelf	3	3
Level 1 4U	Power Distribution Panel	-	2
Level 1 + Level 2	P803-001 Tape Punch	6	-
	Power Distribution Panel	-	2



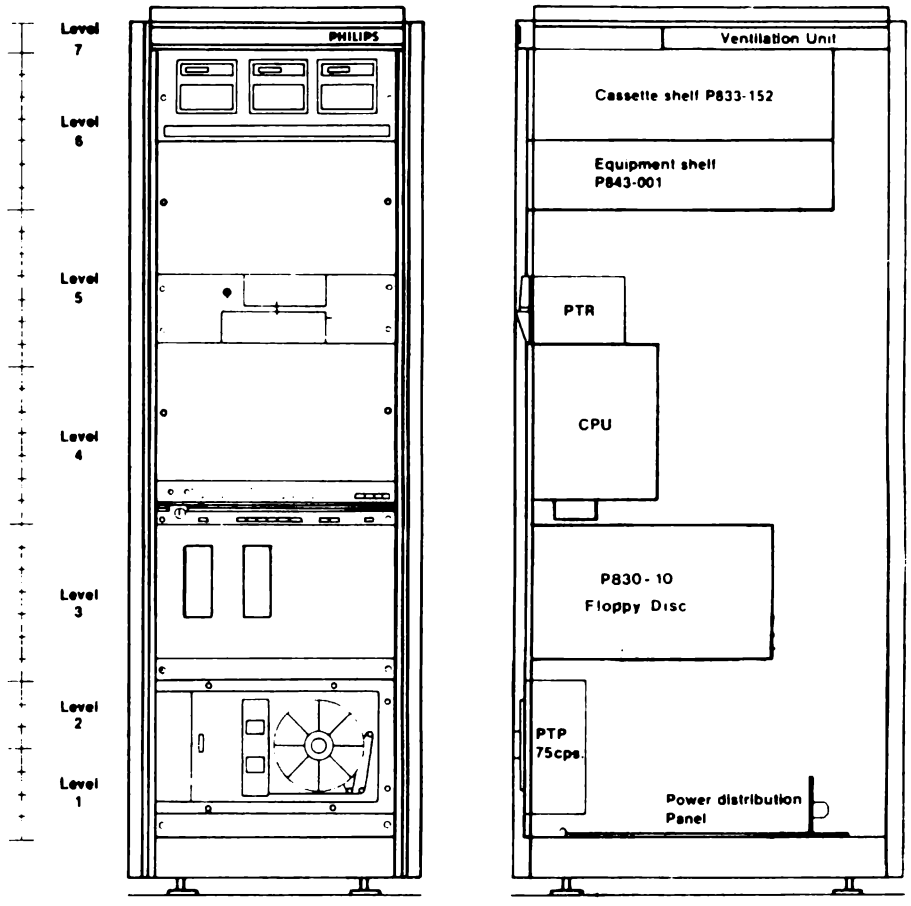


Figure 2-8 Configuration Examples

2.10 Rack Earthing and Cabling Rules

Each rack must have a mechanical ground or safety earth point to which the safety earth points of all the units mounted in the rack must be connected. The rack safety earth point must be electrically connected between the rack frame and the earth wire of the power cable supplying the rack from the site mains. Each rack must also have an electrical ground of 0 volts and the shields of cables must not be used as inter-unit safety earth conductors. Cable shields must be connected to the safety earth using earthing leads as short as possible and of the same cross sectional area. To avoid either internal or inter-rack electro-magnetic interferance, the mains cables to the different units should be mounted against the lefthand side of the rack frame and the I/O cables should be mounted against the righthand side of the rack frame.

2.11 Installing

The following steps are for general guidance only, specific information about the I/O locations, rack layout, cabling, etc. will be given in the Configuration Sheets which are delivered with your system or equipment.

- Unpack and inspect each item to ensure that circuit cards and connectors etc. have not been damaged or become loose during transit.
- Check that the mains input setting of each item corresponds to the site supplies. If not refer to the power supply section of this manual or the relevant part of the manufacturer's handbook (for peripherals), and make the appropriate changes to the input connections.
- Check that the U links have been fitted in accordance with the Configuration Sheets and have not been lost during transit.
- Check that the internal wiring and connectors of the rack are undamaged. If telescopic slides are fitted remove the restraining tape and check that they move freely, and if a pedestal is included tilt the rack backwards and secure the pedestal to the rack using the bolts provided.
- For the peripherals consult the manufacturer's handbook and remove any items fitted to secure moving parts during transit.
- Assemble the Eurorack and the other rack mountable items in the rack in accordance with the pre-determined layout, and move the free standing peripherals into position.
- Make sure that the site supply is not connected to the rack, then make the a.c. power connections to the items installed in the rack using the distribution panel or intermediate terminating blocks as prescribed in the Configuration Sheets.
- Connect the I/O cables to the appropriate CU cards, and the Break and Interrupt wires to the appropriate pins of the CPU and IOP back panel connectors in accordance with the Configuration Sheets.
- Using the Configuration Sheets as a guide connect the I/O cables to the peripherals.
- Stow and secure excess lengths of cables and wires ensuring that items mounted on slides can be withdrawn for servicing purposes.
- Ensure that each item of equipment is switched OFF, then connect the rack and free standing equipment to the site mains supply.
- Switch ON the mains input to the rack and check that mains is present at the distribution panel. If not check fuses and connections.
- Switch ON each device in turn starting with the CPU to ensure that the peripheral controllers are activated before the peripherals become operable. As each device is switched ON check that mains is present by observing the indicator lamps.
- When all the devices are operable check the system using the Test Programs delivered with the system.

If any difficulties are encountered during installation or if advice is needed contact your nearest Philips Office.

2.12 INTERFACE

The interfacing of the different elements of the P851M system is achieved by the General Purpose Bus Simplified (GPBS), which has 46 data and signal lines operating in asynchronous mode. All data transfers between the system elements take place via the GPBS under control of either the CPU, the IOP or DMACU which act as masters of the exchange, all other system elements being slaves. Figure 2-9 shows a block diagram of the interface.

2.13 Physical Aspects of the GPBS Connections

There are 42 data and signal lines common to all 13 slots of the Eurorack and these are implemented on the printed circuit back panel. Two other signals (DMARQN and DMANSN) are common only to the first two slots of the Eurorack, which are used by the CPU and IOP or DMACU cards, and are also implemented on the back panel. The last two signals of the GPBS are for the Interrupt and Break signals. These are dedicated connections made with jumper wires, the interrupts being connected to socket 2 of the CPU card and the breaks being connected to socket 2 of the IOP card.

2.14 GPBS Signals

There are three types of bus signals: command signals, timing signals, and data signals. Any of these signal names followed by the suffix letter N indicates that for command and timing signals the action will become valid when the logic level is low (0), and for data signals indicates that the value transmitted on the bus will be the complementary value. The signals and their functions are described as follows:

- Bus Input/Output -- All elements connected to the Bus receive data signals on these 16 bidirectional lines (BIO00N to BIO15N). The Bus is validated by one of the bus timing signals.
- Address and Function -- These 16 lines represent different data depending on the type of exchange. For memory exchanges the 16 lines (MAD00 to MAD15) represent a memory address which is sent to the memory by one of the masters (CPU, IOP, or DMACU). For other types of exchange the 16 lines represent a function command and the address of either a CU or external register.
- WRITE -- This signal is sent by the masters and has two meanings depending on the type of exchange. For memory exchanges this signal at logic level 1 indicates to the memory a write operation and at logic level 0 indicates a read operation. Exchanges with control units or external registers a logic level 1 indicates an output operation and a logic level 0 indicates an input operation.
- CHA -- This signal is sent by the masters to memory and at logic level 1 indicates a character operation and at logic level 0 indicates a word operation. For exchanges using control units or external registers the signal level will always be 0.
- DMARQN -- This signal is sent to the CPU by either the IOP or DMACU to request control of the bus. The request is made when DMARQN is set to 0 logic level and it remains at 0 as long as one of these masters is using the bus.

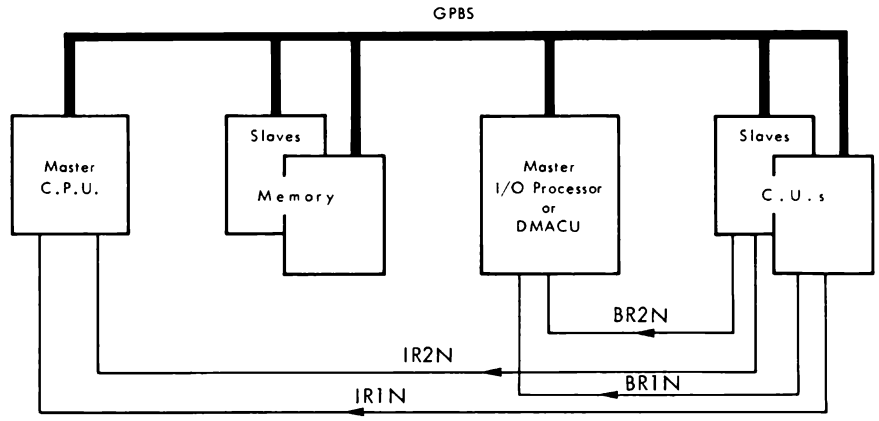


Figure 2-9 Block Diagram of Interface

- DMANSN -- This is the signal sent by the CPU to the other master in reply to the DMARQN signal requesting use of the bus. If the CPU is not using the bus (or will not need it to finish an instruction) it sets DMANSN to logic level 0 and remains at this level until the request has ended.
- TMRN -- This signal is sent to the memory by a master and is used to validate the address on the address lines, to validate data into memory, and to clock the exchange. The master expects the signal TSMN from memory to validate data output from memory and to terminate the exchange.
- TMPN -- This signal is sent to a peripheral control unit by the master and is used to validate the control unit address, to validate data to the control unit, and to clock the exchange. The control unit responds with the signal TSMN which validates data from the control unit and terminates the exchange.
- TMEN -- This signal is sent to an external register by the master and is used to validate the address and data to the register and to clock the exchange. The external register replies with the TSMN signal to validate data from the register and to terminate the exchange.
- TSMN -- This signal is sent by the slave to the master and validates data from the slave to the master and terminates any exchange with the master.
- ACN -- This signal is sent by the control unit to the CPU when a command is accepted.
- CLEARN -- This signal is sent by the CPU to all the elements connected to the bus and is used as a general reset.
- PWFN/RSLN -- These signals are sent from the power supply to ensure a correct switch ON / switch OFF sequence in case of a mains power failure.
- IRN -- These are the dedicated lines, comprising jumper wires between the control units and socket 2 of the CPU card, which signal to the CPU that a data or status transfer is required.

Table 2-2 GPBS Signals

No	Name	Mnemonic	Source	End	Function
16	Bus Input Output Lines	BIOXXN	All	All	Data Path
16	Address or Function Lines	MADXX	Master	Slaves	Slave addressing
1	Character	CHA	Master	Memory	Character mode
1	Write	WRITE	Master	Memory	Transmission direction
1	Timing Master to Control Unit	TMPN	Master	CU	Exchange timing pulse
1	Timing Master to Memory	TMRN	Master	Memory	Exchange timing pulse
1	Timing Master to External Registers	TMEN	Master	Register	Exchange timing pulse
1	Accept	ACN	Slave	CPU	I/O dialogue with CU
1	Timing Slave to Master	TSMN	Slave	Master	Exchange timing pulse
1	Power Failure	PWFN	Power Supply	All	Power control
1	Reset Line	RSLN	Power Supply	All	Power control
1	Clear	CLEARN	CPU	All	Initialization
1-16	Interruption	IRN	Control Unit	CPU	I/O dialogue with CU
1-8	Break	BRN	Control Unit	IOP	I/O dialogue with CU
1	DMA Request	DMARQN	IOP	CPU	Master requests the Bus
1	DMA Answer	DMANSN	CPU	IOP	CPU gives the Bus to the other master

- BRN -- These are the dedicated lines comprising jumper wires between the control units and socket 2 of the IOP which signal that the control unit is ready for a data transfer.

Table 2-2 shows the relationship of the bus signals.

2.15 Data Transfers

All transfers on the bus are effected by a master/slave relationship between two of its elements. The CPU is always a master but the IOP or DMACU can be either a master or slave depending on the type of exchange. CPU always has the Bus but if CPU is 'not busy' and IOP or DMACU requests the Bus with DMARQN then CPU hands over the Bus with DMANSN. At the end of the exchange the bus is handed back to the CPU. The different types of data transfer are shown in Table 2-3. All the types of data transfer are initiated by program instructions which are automatically fetched from memory by the CPU and during the execution of certain instructions memory access may again be necessary. Because every memory access takes place via the bus, using the MAD lines for addressing and the BIO lines for data, any master requiring the bus for a transfer must ask the CPU for bus control before starting an exchange. Data transfers to and from peripherals are initiated by the CPU using I/O instructions and they take place under control of either the Programmed, IOP or DMACU channels.

2.16 I/O Instructions

Any exchange between either the CPU, IOP or DMACU and one of the CU's is started by a CIO Start instruction, then the transfer can take place via either the Programmed Channel or the IOP Channel. The actual transfer instructions (INR, OTR) are also sent by the CPU to the CU's for a Programmed Channel transfer but for IOP transfers they are simulated by the logic of the IOP card. The remaining instructions (SST, TST, and CIO Halt) are also sent out by the CPU to the CU's for any type of transfer. The CIO Halt instruction is always accepted no matter what state the CU may be in. The other instructions will only be accepted if the CU is in the appropriate state. Note that the TST instruction is not accepted by all Control Units. For a description of I/O instructions see System Description.

Table 2-3 Types of Data Transfer

Master of Exchange	Addressed Element	Signal Used	
		by Master of Exchange	by Element Addressed
CPU	Memory	TMRN	TSMN
CPU	Control Unit	TMPN	TSMN
CPU	I/O Processor	TMEN	TSMN
CPU	DMACU	TMEN	TSMN
CPU	External Register	TMEN	TSMN
I/O Processor	Memory	TMRN	TSMN
I/O Processor	CU	TMPN	TSMN
DMACU	Memory	TMRN	TSMN

2.17 Programmed Channel

The Programmed Channel uses the CPU to control transfers of data characters or words to or from a CU. The CU is initiated with a CIO Start instruction and then each character or word is requested by interrupt to enable an INR or OTR instruction to effect an exchange. Normally a TST instruction is sent to the CU before the exchange to check that the CU is not busy, and at the end of an exchange an SST instruction is sent to check that the transfer has been successful. When the transfer is complete the CPU sends a CIO Halt instruction which switches the CU to Wait state and then an SST checks the status and switches the CU to Wait state and then an SST checks the status and switches the CU to the Idle state.

2.18 IOP Channel

The IOP Channel enables blocks of data (either words or characters) to be transferred between the CUs and memory without using the CPU. The CPU initiates the exchange with a CIO Start but the IOP controls the exchange. To initiate an exchange the CPU must first load two of the registers of the IOP with the relevant parameters for the exchange. This is done using the WER instruction (Write External Register). With this information the IOP is able to control the exchange without aid from the CPU. After the WER instruction the CPU activates the CU with a CIO Start

instruction and from then on the IOP becomes master of the exchange. Each time a transfer is required the CU signals the IOP with the Break Request signal. At the end of transfer IOP sends MAD03 to end the exchange, the CU switches to Wait state and waits for the SST which switches the CU to the Idle state.

2.19 Exchange Timing

The timing of the data exchanges is shown in Figure 2-10 which shows the timing at the Master level and in Figure 2-11 which shows the timing at the GPBS level.

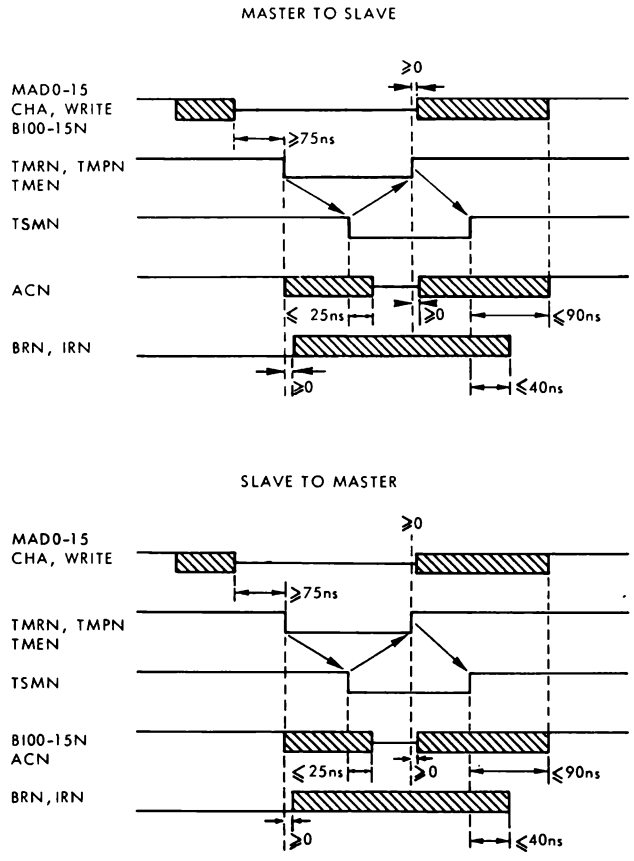


Figure 2-10 Exchange Timing at Master Level

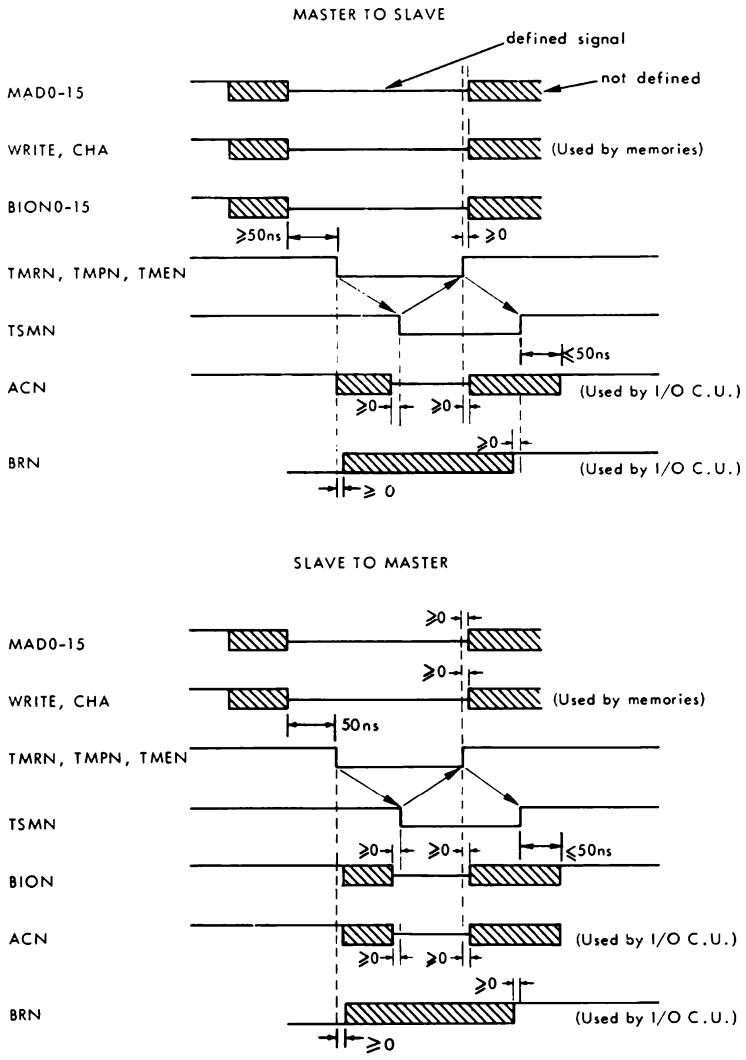


Figure 2-11 Exchange Timing GPBS Level

2.20 Interrupt Signals

The interrupt signals are dedicated connections made by jumper leads between the CUs and connector 2 on the back panel for external interrupts and between pins of connector 2 for internal interrupts. The internal interrupts are for Power Failure (PFFN), Real Time Clock (RTCAN), Control Panel Interrupt (CPFN), and Program Interrupt (PIFN). The priority for each interrupt signal is determined by the position of the jumper lead on connector 2, IS00N having the highest priority and IS15N the lowest priority. The position of the jumper lead is selected by the customer at system generation time or can be changed at a later date to meet new system requirements. There is a 6 bit Program Level Register (PLR) in the CPU which contains a number from 0 to 63 (0 is the highest level) representing the priority level of the program. When an interrupt is received by the CPU it is passed to a priority encoder whose output is compared with the priority level in the PLR. If the interrupt has a higher priority than the PLR the program is interrupted and the device causing the interrupt is processed by the CPU. Figure 2-12 shows a block diagram of the interrupt system.

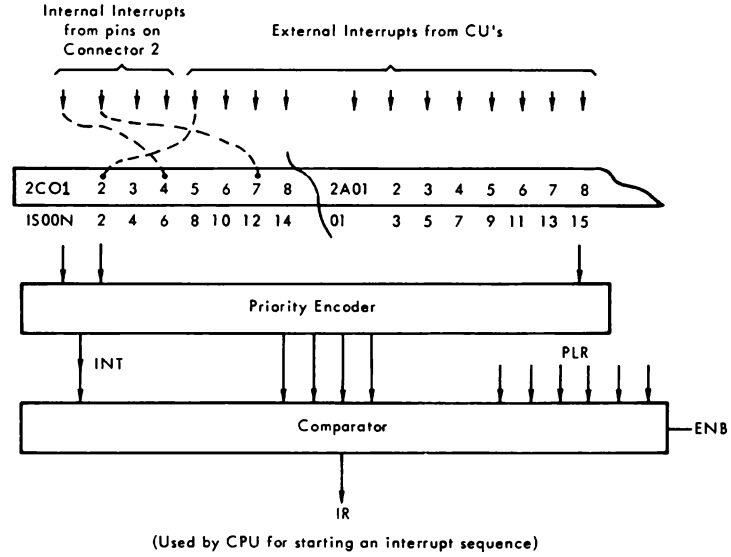


Figure 2-12 Interrupt System

2.21 Break Signals

The break signals are also dedicated connections made by jumper leads between the CUs and connector 2 of either the IOP or DMACU card. The priority of each break is determined by the position of the jumper lead on connector 2, BR0N having the highest priority and BR7N the lowest priority. The break with the highest priority is determined by logic similar to that used by the interrupt system but is entirely independant from the CPU logic.

2.22 Back Panel Connector Pinning

The interface signal pinning of the back panel connectors is given in Tables 2-4 and 2-5.

2.23 Interfacing for Customer Designed Control Units

For customers who wish to design their own control units there are two possibilities: they can either buy a General Purpose card direct from Philips which contains all the interface logic or they can design the whole card themselves.

2.24 Card Size

The slots available for control unit cards are designed to hold double format Eurocards whose size is 160mm x 233.4mm. The card connector which plugs into the back panel GPBS connector must comply with the pinning given in Table 2-4. Other connectors can be pinned according to the customer's requirements.

Table 2-4 Back Panel Connector 1 Pinning

Pin Number	Signal Name		Pin Number	Signal Name
1C01	+12VL		1A01	-12VL
1C02	+5VM		1A02	+5VM
1C03	Not Used		1A03	Not Used
1C04	* DMANSN (IR1N)		1A04	* DMARQN (BR1N)
1C05	* * +10VL (IR0N)		1A05	* * +10VL (BR0N)
1C06	0V		1A06	0V
1C07	PWFN		1A07	RSLN
1C08	CLEARN		1A08	ACN
1C09	0V		1A09	0V
1C10	TMPN		1A10	TSMN
1C11	TMEN		1A11	TMRN
1C12	0V		1A12	0V
1C13	MAD00		1A13	MAD01
1C14	MAD02		1A14	MAD03
1C15	MAD04		1A15	MAD05
1C16	MAD06		1A16	MAD07
1C17	MAD08		1A17	MAD09
1C18	MAD10		1A18	MAD11
1C19	MAD12		1A19	MAD13
1C20	MAD14		1A20	MAD15
1C21	CHA		1A21	WRITE
1C22	0V		1A22	0V
1C23	BIO08N		1A23	BIO00N
1C24	BIO09N		1A24	BIO01N
1C25	BIO10N		1A25	BIO02N
1C26	BIO11N		1A26	BIO03N
1C27	BIO12N		1A27	BIO04N
1C28	BIO13N		1A28	BIO05N
1C29	BIO14N		1A29	BIO06N
1C30	BIO15N		1A30	BIO07N
1C31	+12VM		1A31	+12VM
1C32	+5VL		1A32	+5VL

* DMANSN and DMARQN are only connected at the first two positions of the back panel connector for use by the CPU and either the IOP or DMACU cards.

* * The +10VL supply is available at the first position of the back panel connector and is only used by the CPU card.

() For the eleven sockets available for use by memory and CUs, the signals within parenthesis may be sent to the CPU, IOP, or DMACU via the pins adjacent in the Table.

Table 2-5 Back Panel Connector 2 Pinning

Pin Number	Signal Name CPU Card	Signal Name IOP Card	Pin Number	Signal Name CPU Card	Signal Name IOP Card
2C01	IS00N	BR0N	2A01	IS01N	BR1N
2C02	IS02N	BR2N	2A02	IS03N	BR3N
2C03	IS04N	BR4N	2A03	IS05N	BR5N
2C04	IS06N	BR6N	2A04	IS07N	BR7N
2C05	IS08N		2A05	IS09N	
2C06	IS10N		2A06	IS11N	
2C07	IS12N		2A07	IS13N	
2C08	IS14N		2A08	IS15N	
2C09	PFFN		2A09	CPFN	
2C10	RTCAN		2A10	PIFN	
2C11	CPZ0N		2A11		
2C12			2A12		
2C13			2A13		
2C14			2A14		
2C15			2A15		
2C16			2A16		
2C17			2A17		
2C18			2A18		
2C19			2A19		
2C20			2A20		
2C21			2A21		
2C22			2A22		
2C23			2A23		
2C24			2A24		
2C25			2A25		
2C26			2A26		
2C27	RUNFA		2A27	0V	
2C28	RTCFZ1N		2A28	0V	
2C29	PUPN		2A29	0V	
2C30	MCLN		2A30	0V	
2C31	PUPS		2A31	0V	
2C32	BIOEKEY		2A32	0V	

Note: The ISxxN signals of the CPU card are connected to the IRN signals of the CUs by jumper leads, and the BRxN signals of the IOP card are connected to the BRN signals of the CUs by jumper leads.

2.25 GENERAL PURPOSE CARD (Figure 2-13)

The General Purpose Card (GPZ) contains the basic logic to interface to the General Purpose Bus S. The card has the following facilities:

- BIO Transceivers to interface the bidirectional lines.
- Command Decoder to decode the Control Instructions
- Address Recognition to recognise the Control Unit address, the address being selected with links on the card.
- Request a transfer via the Program Channel or IOP Channel, selectable by strap.
- Storage of an end of transfer signal for IOP Channel.

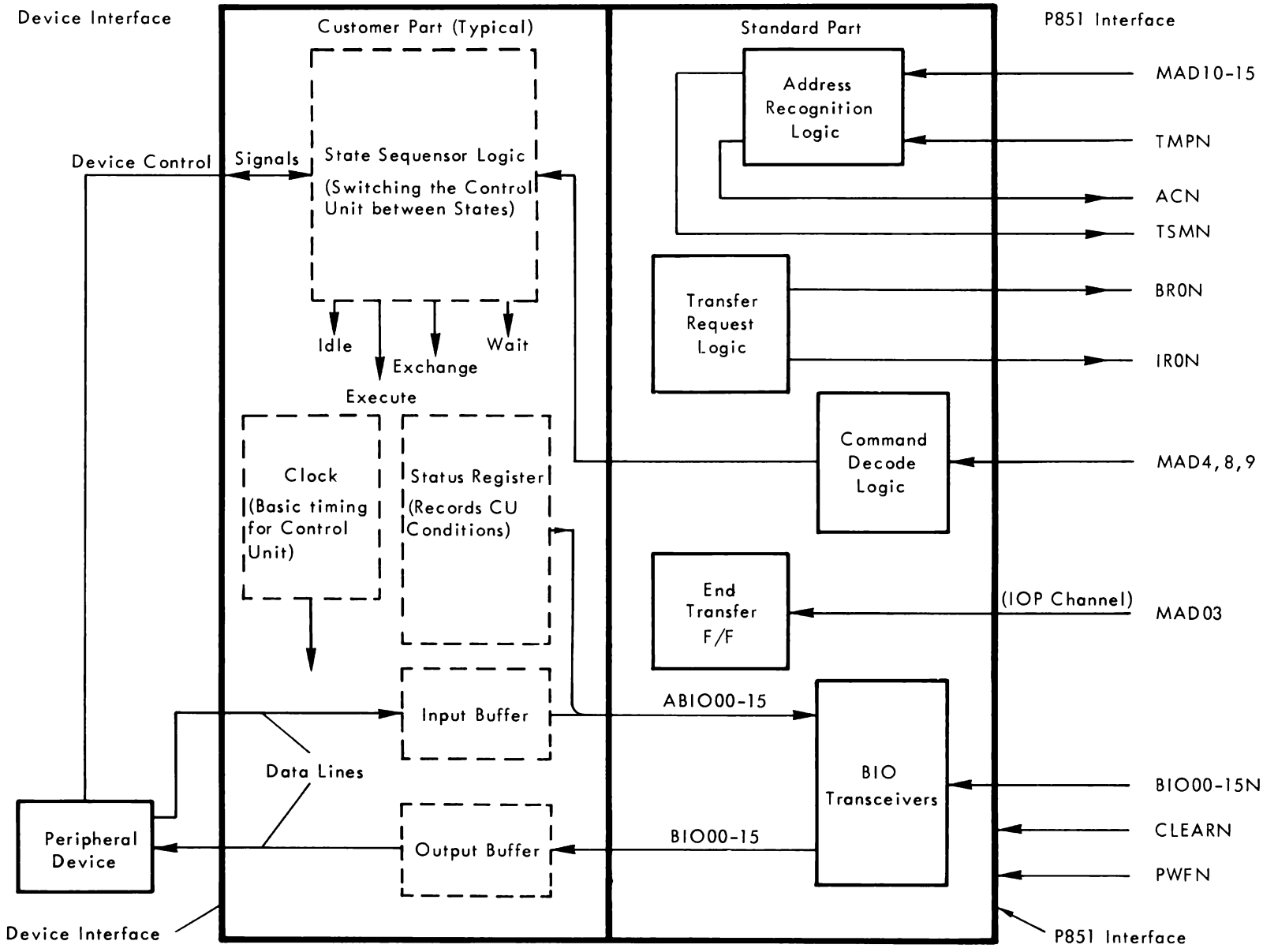
The operation of the finished Control Unit is restricted in that it may only interface one device and only operate as a slave in the P851 system. For the typical operation of a Control Unit see Serial Control Unit (SCUZ), Volume 1, Section 7. Information on the Control Instructions are given in Section 1 but if more detail is required then refer to the software manuals.

2.26 Signals (Figure 2-14)

The Interface signals between the P851 and the General Purpose Card are already described in the Interface part of this section. The following paragraphs describe the signals between the standard part of the card and the customer designed part. Note that all fan-ins and fan outs are for either LS (Low Power Schottky) logic of 0.25 or for TTL (Transistor Transistor Logic) of 1 (1.6mA).

2.27 Signals Standard Logic → Customer Logic.

- ARE — Address recognised is active high if the address on the MAD10-15 Lines is identical to the address selected by the 'U' links of the Card. Fan-out = 2.75.
- TSM — Enabled at the same time that TSMN is sent to validate the transfer, active high. Fan-out = 5.
- PWFN — System power fail signal, active low. Fan-out = 1.
- CLEAR — Enabled from the Master Clear signal, active low. Fan-out = 4.75.
- CLIRN — Enabled from the Master Clear signal, active low. Fan-out = 4.25.



Note: The logic shown for the Customer Part is not mandatory and is only intended to give an indication of what is required.

Figure 2-13 General Purpose Card

- EORF — End of Exchange signal for Multiplex Channel, active high. Fan-out = 5.
- BIO00-15 — Output data from BIO Transceivers, active high. Fan-out = 10.
- ACCION — CIO Start Command accepted, active low. Fan-out = 4.75.
- HALTN — CIO halt command accepted, active low. Fan-out = 4.75.
- ACOTRAN — OTR command accepted, active low. Fan-out = 4.75.
- ACOTRBN — OTR command accepted, active low. Fan-out = 4.75.
- ACINRAN — INR command accepted, active low. Fan-out = 4.75.
- ACINRBN — INR command accepted, active low. Fan-out = 4.75.
- ACSSTN — SST command accepted, active low. Fan-out = 4.75.
- ACTSTN — TST command accepted, active low. Fan-out = 4.75.
- AMAD041 — This signal must be used by the customer to validate BIO Lines during an input to CPU (INR, SST, TST).

2.28 Signals Customer Logic - Standard Logic. If the customer decides not to use these signals then they must be connected to an inactive level.

- INCTN — Validate CIO Start, active low. Fan-in = 0.25.
- EXCHAN — Validates INR and OTR commands, active low. Fan-in = 0.5.
- EXCHBN — Validates INR and OTR commands, active low. Fan-in = 0.5.
- WSTN — Validate SST, active low. Fan-in = 0.25.
- CLOKBRF — clocks the Break Request F/F on the low to high transition. Fan-in = 0.5.
- DATABR — Data is ready so set Break Request F/F, active high. Fan-in = 0.25.
- BRFZ0N — Direct reset of the Break Request F/F, active low. Fan-in = 0.75.
- BRFZ1N — Direct set of the Break Request F/F, active low. Fan-in = 0.5.
- INR — Interrupt Request, active low. Fan-in = 0.25.
- ABIO00-15 — Input Data Lines to the BIO Transceivers, active high. Fan-in = 1.
- VAL2 — Validates the input Data Lines for ABIO00-07, active high. Fan-in = 0.25.
- VAL1 — Validates the input Data Lines for ABIO08-15, active high. Fan-in = 0.25.

2.29 Major Logic Units (Figure 2-14)

There are 5 major logic units on the GP Card.

2.30. Address Recognition Logic. The address of the Control Unit is selected by 6 'U' Links on the card. If the address is correct the outputs of the OR-Gates are high so signal TMP and thus the generation of TSMN and TSM. Signals INCTN, WSTN, EXCHAN and EXCHBN must be stabilised within 90nS of signal ARE going high.

2.31 Accept Command Logic. The MAD04,08,09 Lines are decoded from the instruction to enable the output signals in the following table:

Command	MAD04	MAD08	MAD09
CIO Start	0	1	1
CIO Halt	0	1	0
OTRA	0	0	0
OTRB	0	0	1
INRA	1	0	0
INRB	1	0	1
SST	1	1	1
TST	1	1	0

When the command is decoded ACC enables ACN to be sent to CPU indicating that the command is accepted. If the instruction is not accepted ACN stays high and an error is recorded at CPU. Note that TST (ACTSTN) and CIO (Halt) (ACHALTN) are always accepted but for the other inputs it is essential that the Control Unit is in the correct state. Note that for the INR/OTR bit 9 may be used to specify a particular function and so may be at either logic level.

2.32 Break Request F/F. This F/F is only used by IOP Channel to send a Break Request to IOP. The F/F is clocked by CLOKBRF when the Control Unit is ready to send or receive data (DATABR). Direct set and reset signals may also be used.

2.33 End Exchange F/F. For IOP Channel the IOP sends MAD03 to the Control Unit to end the exchange. Signal ARE clocks the F/F to record that the exchange has ended and enables signal EORF. The F/F will normally reset the next time that it is clocked with MAD03 high or by Master Clear at the direct reset input.

2.34 BIO Transceivers. The BIO Transceivers connect the Control Unit to the bi-directional lines of the General Purpose Bus S. The direction of the Transceivers is decided by pins 7 and 9 at the device. To enable data in the direction Bus → Control Unit, signals VAL1/2 are low inhibiting BIOVAL1/2. To enable data Control Unit → Bus, VAL1/2 are high.

2.35 Design Rules

The General Purpose card is supplied to give the customer freedom in design but there are certain rules which must be adhered to in order to ensure that the Control Unit operates correctly in the P851 system.

2.36 Accepting a CIO Start.

- On the trailing edge of the ACCION pulse, signal INCTN must be made inactive.
- Signals BRN and IRN must already be inactive high.
- If Master Clear was received it must make INCTN inactive.

2.37 Input or Output Data with INR or OTR.

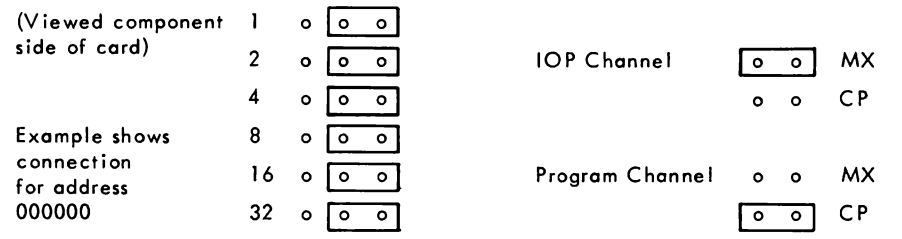
- Enable EXCHAN/EXCHBN active low to accept the exchange command.
- Enable an exchange request for IOP Channel BRN active low and for Program Channel IRN active low.
- For output data Bus → Control Unit, ACOTRAN/ACOTRBN active low means that data is on the BIO Lines and must be stored.
- For input data Control Unit → Bus ACINRAN/ACINRBN active low means that data must be sent to Bus. Note that VAL1 or VAL2 must be enabled to validate data at the BIO Transceivers.
- While processing the INR/OTR the Break Request F/F must be reset on the leading edge of the TSM signal.
- On the trailing edge of ACINRAN/ACINRBN, ACOTRAN/ACOTRBN the EXCHAN/EXCHBN must be deactivated.

2.38 Transfer Ends. For continuous operation the exchanges continue until an end of transfer is indicated. For Program Channel a CIO Halt and for IOP Channel MAD03 is the indication. After signal ACCN goes high signal WSTN should be activated (Control Unit is in the wait for SST state) and signal IRN activated which requests the SST instruction.

2.39 Send Control Unit Status. On the leading edge of ACSSTN the Status Word of the Control Unit must be present on the ABIO Lines and enabled by VAL1/VAL2. During the processing of this command IRN must be deactivated. On the trailing edge of the ACSSTN signal WSTN is deactivated and INCTN goes active low (Control Unit Inactive).

2.40 Executing TST Instructions. When a TST is executed, testing if the Control Unit is busy (bit 15) signal INCTN may be used to indicate this condition enabled by VAL1.

2.41 Card Links. There are 2 sets of link positions on the card for selecting the Control Unit address and the Data Channel.



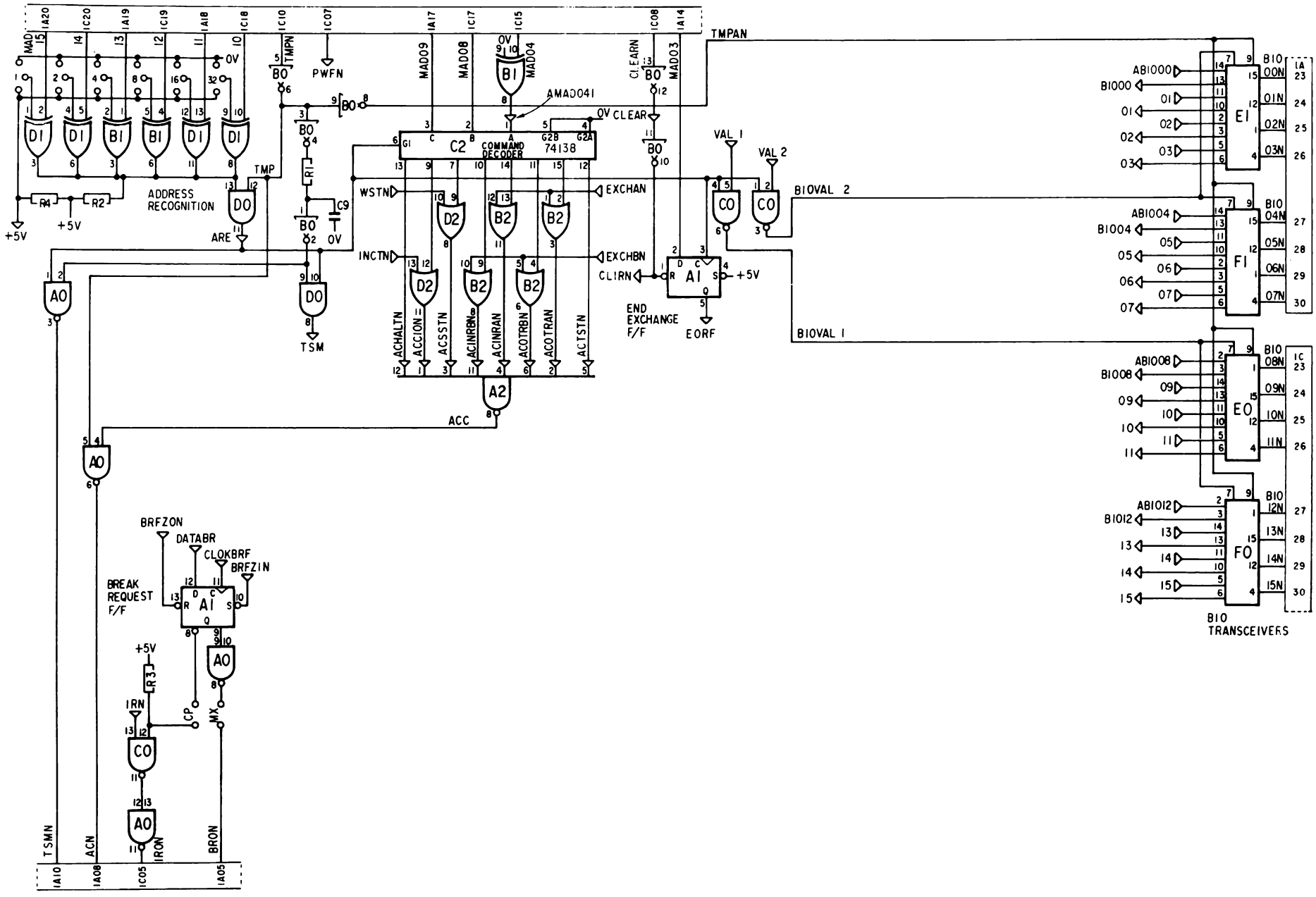


Figure 2-14 General Purpose Card Logic Diagram

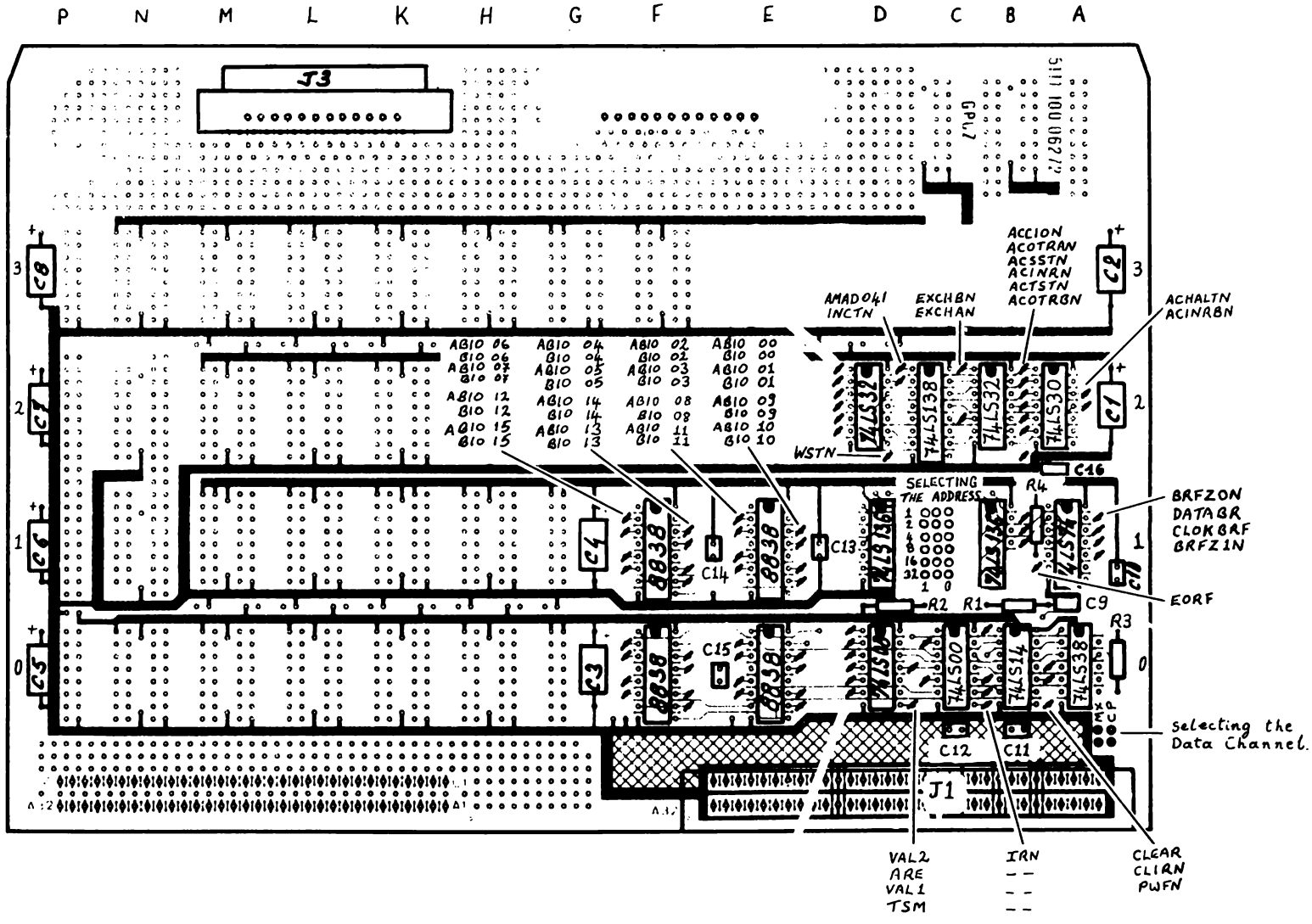


Figure 2-15 General Purpose Card Component Layout

Table 2-6 General Purpose Card Parts List

Reference	Description	12NC Code
	Printed circuit	5111 100 06272
E0,E1,F0,F1.	Integrated circuit 8838.	
A0.	Integrated circuit 7438.	
C0.	Integrated circuit 74LS00.	
D0.	Integrated circuit 74LS08.	
B0.	Integrated circuit 74LS14.	
A2.	Integrated circuit 74LS30.	
B2,D2.	Integrated circuit 74LS32.	
A1.	Integrated circuit 74LS74.	
B1,D1.	Integrated circuit 74LSi36.	
C2.	Integrated circuit 74LS138.	
R2,3,4.	Resistor 1Kn, 1/4W, 5%.	
R1.	Resistor 100n, 1/8W, 1%.	
C9.	Capacitor 680pF, 100V, 10%, ceramic.	
C10-16.	Capacitor 10nF, ceramic.	
C1-8.	Capacitor 10µF, 25V, Fitco.	
J1.	Connector 100 964 053.	
J3.	Connector 206 604 1.	

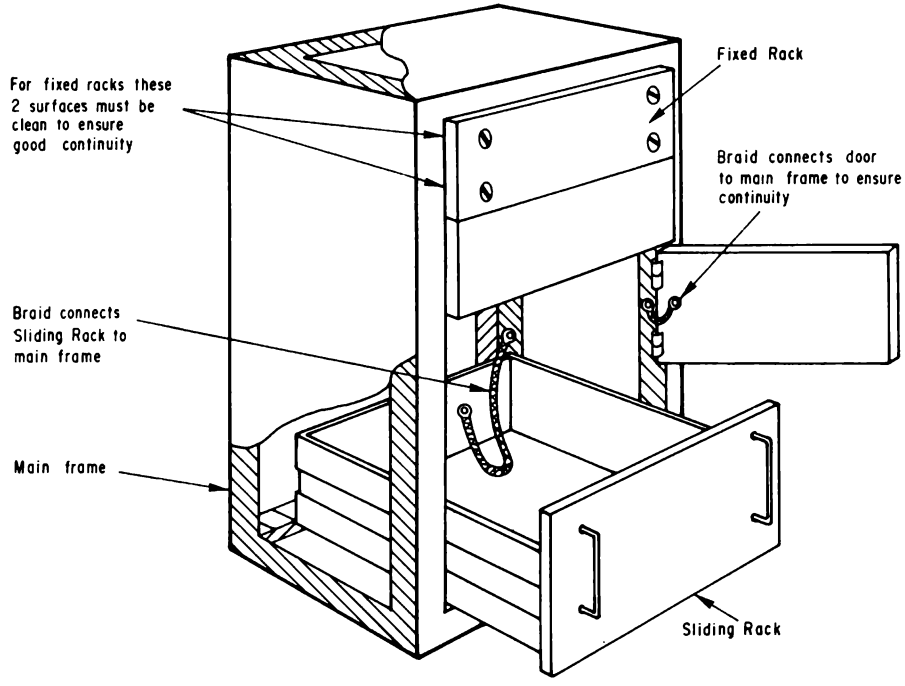
2.42 RULES FOR CONNECTING GROUNDS IN A SYSTEM

These rules must be observed for all installations to ensure that external interference is reduced to a minimum and that the electrical safety regulations are complied with.

2.43 Grounding for Cabinets and Racks

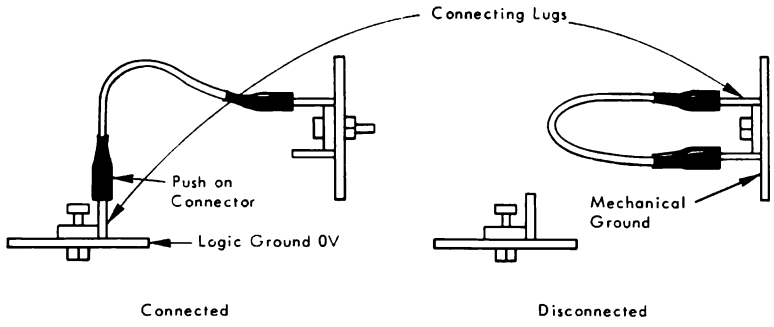
The following rules should ensure that the cabinets and racks have good ground continuity.

- Fixed Racks — The rack is fitted to the cabinet mainframe by screws. A good continuity between cabinet and rack is ensured by keeping the mating surfaces clean.
- Sliding Racks — A rack mounted on telescopic rails does not have a reliable ground path so it is essential that an electrical link is made using metal braid. Note that it is not acceptable to use the ground conductor of the mains lead or cable shielding for the ground connection.
- Cabinet Panels and Doors — All doors and panels must have a good electrical link with the mainframe and in particular the doors must be connected with metal braid.

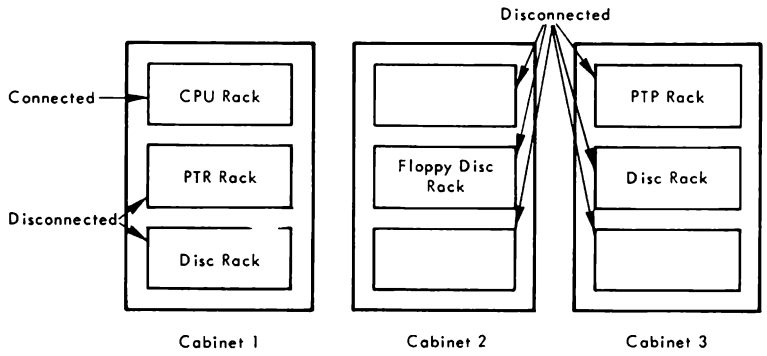


2.44 Logic Ground and Mechanical Ground

Logic ground and mechanical ground points are provided for each rack. They are situated close to each other and may be connected or not.



A system may use several cabinets each containing a number of racks but only one rack may have this link connected. All other rack links are unconnected.

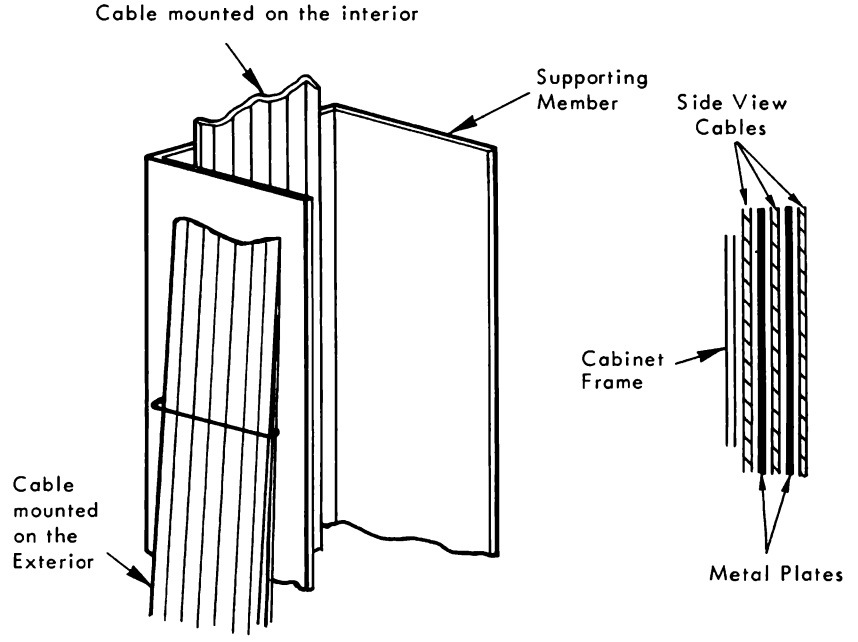


In a normal configuration the CPU rack is connected and all other racks are unconnected. If more than one CPU is being used in the system then the user decides on which CPU rack to have the link connected.

2.45 Flat Cables

Flat cables without shielding may cause electromagnetic fields to be set up in the equipment so the use of non-shielded flat cables should be avoided. If it is essential to use non-shielded flat cables then special precautions must be taken to avoid these fields being set up.

- If channel type supporting members are in the cabinet then they should be used as they provide a shield around the cable.
- If these supporting members are not available then the cables should be run separately, either clamped flat against the panels or against the outside of other supporting members.
- Mounting flat cables one on top of the other should always be avoided. If this method must be used then a metal plate should be inserted between the flat cables, suitably supported and electrically connected to the cabinet.



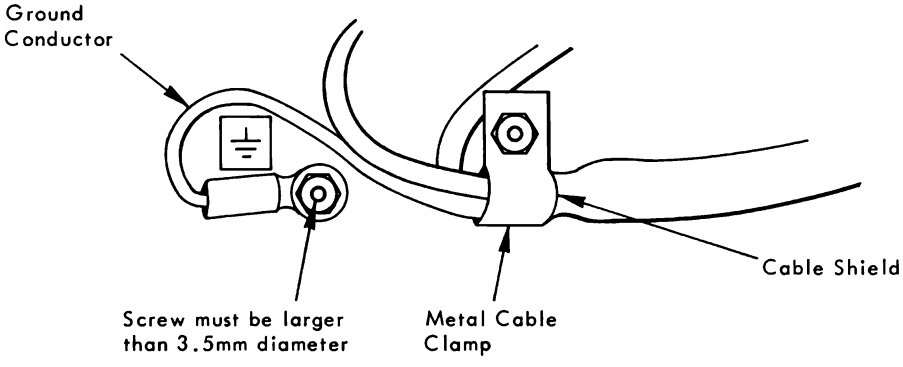
2.46 Mains Cables

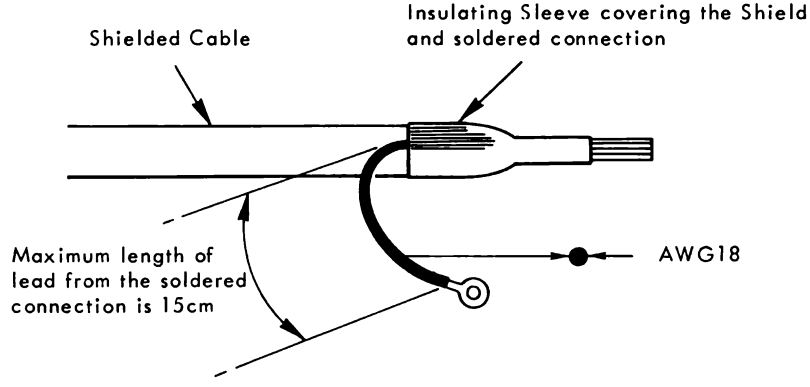
When planning an installation an area along the mainframe should be reserved for the mains cables and all other cables should be kept as far as possible from the area. It is recommended that the cable be shielded either before or after filtering and that this shielding be connected to a mechanical ground at each end of the cable. The mains cable must be secured to the chassis by a metal clamp at the point where the shielding extends out of the cable sheath. Note that the screw fixing the ground conductor to the chassis must be as close as possible to the metal cable clamp.

2.47 General Rules for Connecting Shielded Cables

All twisted pairs must be contained inside a shielded cable and this shield must be connected to mechanical ground at each end. Specific lead dimensions are given for some peripherals but as a general rule the following dimensions should be adhered to.

- the lead must be as short as possible and never exceed a maximum length of 15cm.
- Connection to the shielding must be soldered and connection to the mechanical earth must be with either an eye-lug (for a 3mm screw) or a push-on type connector (Faston 6.35 or equivalent).
- The lead cross section must have a minimum guage of AWG18.





2.48 Special Rules for Connecting Shielded Cables

For some devices the lead dimensions for connecting the shielded cables at the device and at the CPU are different to the general rules. These devices are listed in the following table.

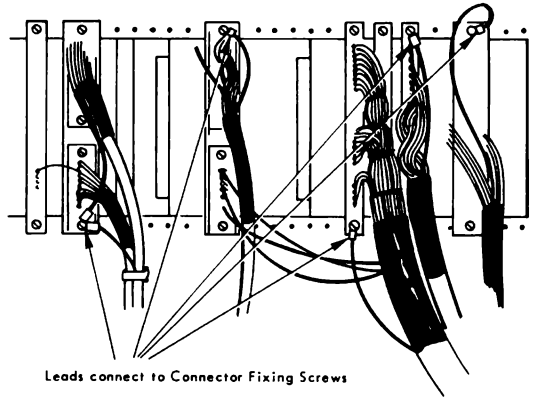
Table of Special Lead Dimensions

	At the Device			At the CPU		
	AWG	Max Length	Eye Lug	AWG	Max Length	Eye Lug
Mag Disc X1215/16	16	15cm	4mm	16	15cm	3mm
Mag Disc 9760/9762	16	08	3	16	10	3
Line Printer X1415/25	16	10	3	16	10	3
PER 3100 (V24 Interface)	16	15	4	16	10	3
PER 3100 (Curr. Loop Int.)	16	15	4	18	03	Molex pin 3
ASR33 (V24 Interface)	16	10	3	16	10	3
ASR33 (Curr. Loop Int.)	18	03	Molex pin 6	18	03	Molex pin 3
Tape Reader 2540EP	16	10	3	16	10	3
Card Reader CM300L	16	15	4	16	10	3
Tape Punch 4070	16	10	4	16	10	3
Display P817	16	10	4	16	10	3
	At the Terminal Box			At the CPU		
Remote Control Cables	18	10	-	18	03	Molex centre pin
	At the Extension Rack			At the CPU		
Break Cables	16	10	3	16	15	3
	At the Cassette Rack			At the CPU		
Break Cables	16	10	3	16	15	3

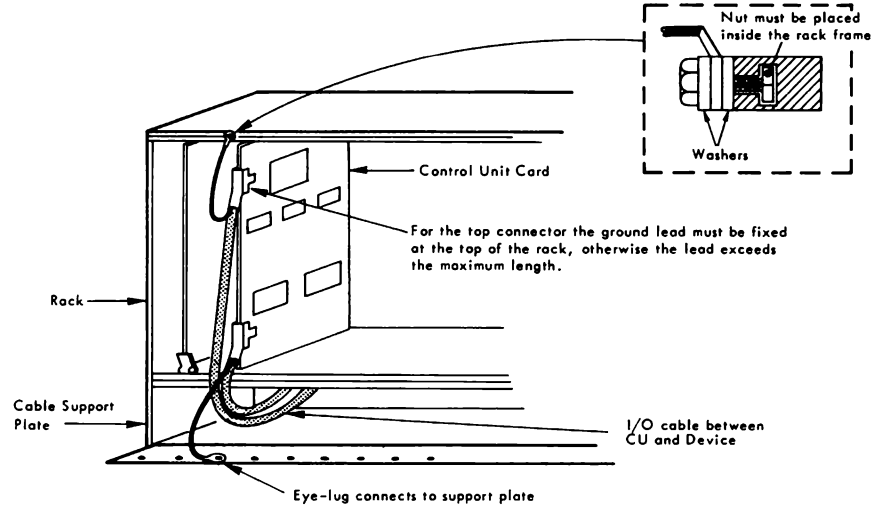
2.49 Connecting the Ground Lead in a Cabinet or Rack

The connection of ground leads for a Cabinet or Rack depends on the type of installation. In the following examples different types of ground connections are given.

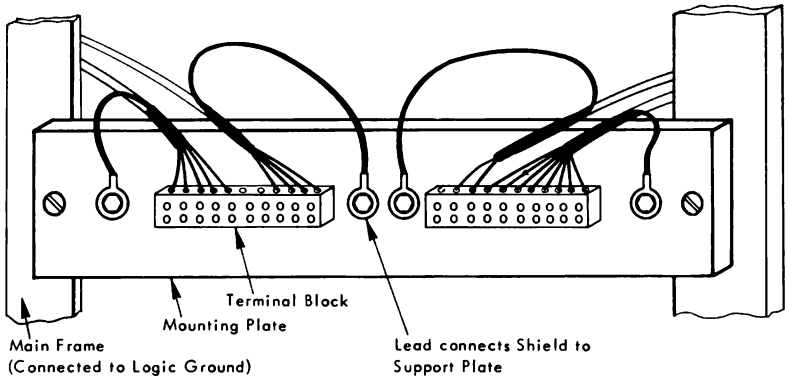
Example using Connector Fixing Screws



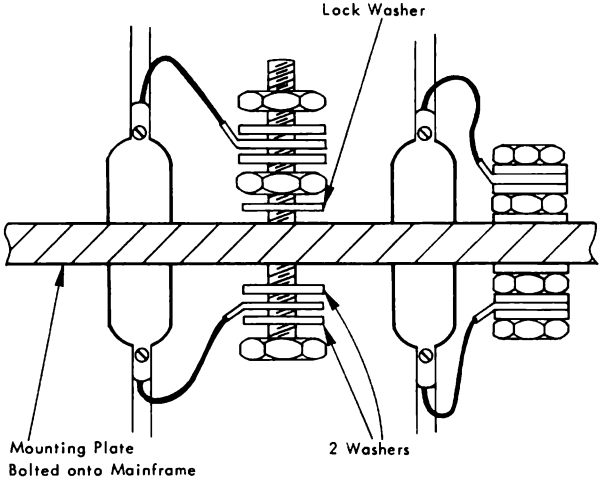
Example using Cable Support Plate



Example using Terminal Blocks



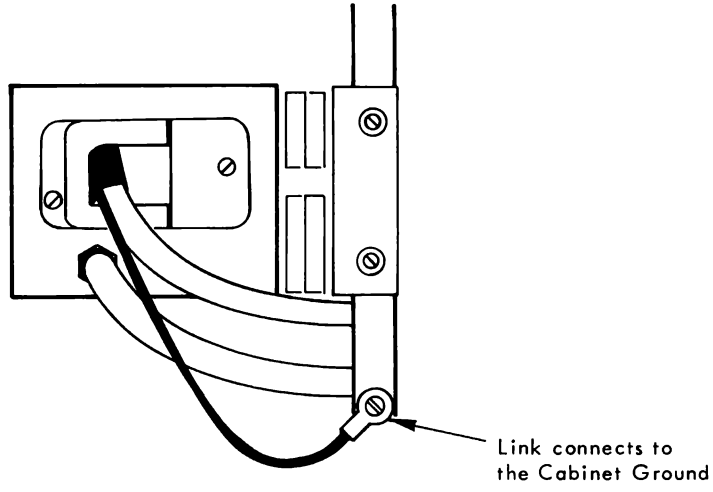
Example using Connectors (Top View)



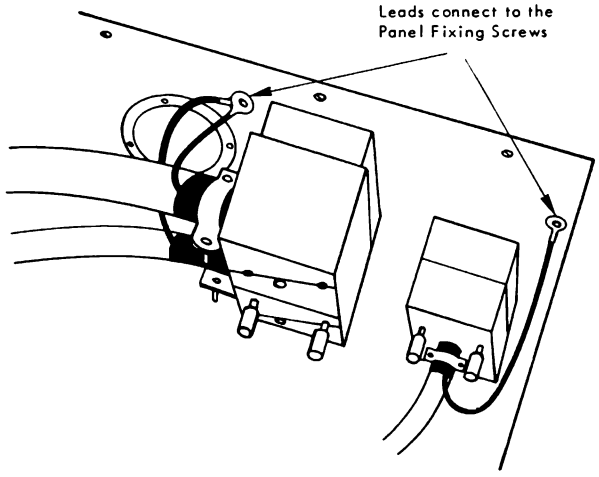
2.50 Connecting a Ground Lead at the Devices

The following illustrations show how a ground lead is connected at some peripheral devices.

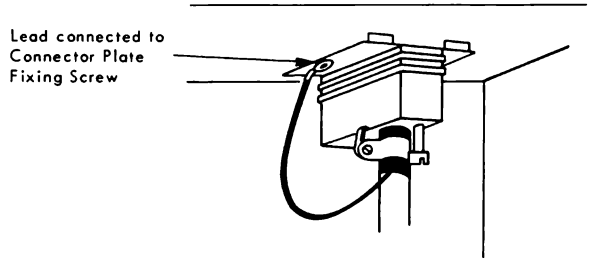
Connecting to the Mag. Disc Control Units (CDD X1215 and X1216)



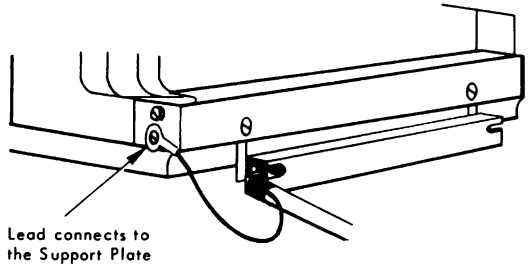
Connecting to the Mag Disc Control Units (CDD CDC 9760 and 9762)



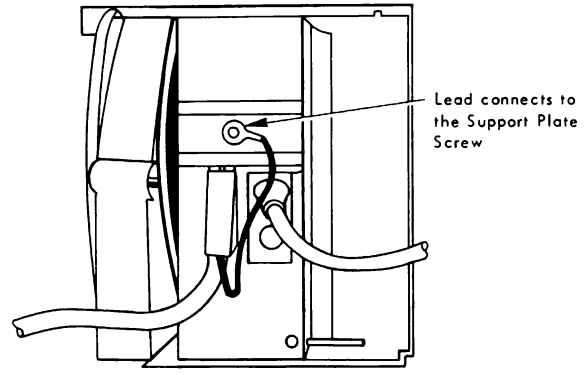
Connecting to the Line Printers (X1415 and X1425)



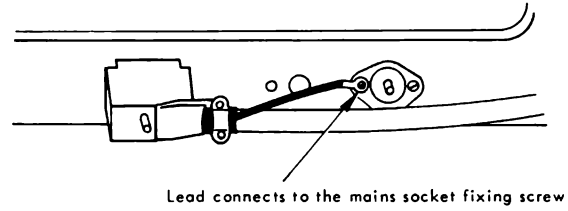
Connecting to the Paper Tape Reader (Digitronics 2540 EP)



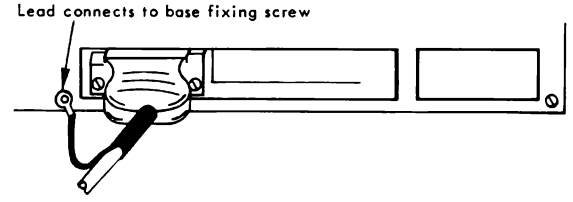
Connecting to the Paper Tape Punch (Facit 4070)



Connecting to the Card Reader (CM300L)

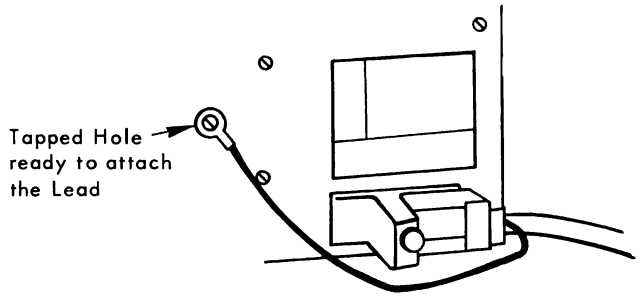


Connecting to the Display Console (P817)



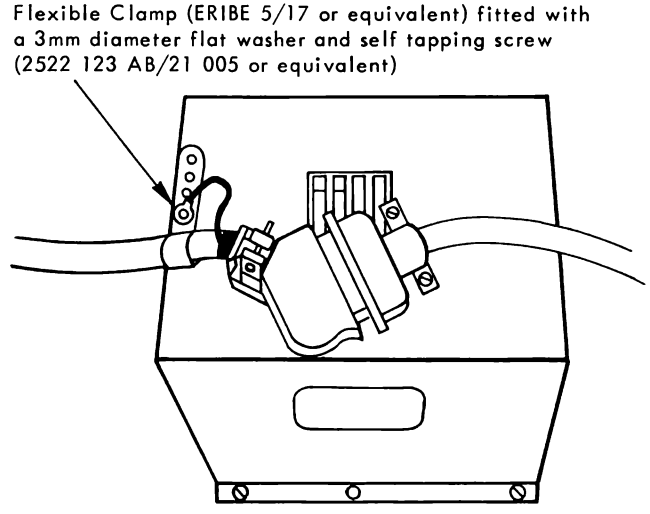
Connecting to the Serial Control Unit PER3100

V24 Interface: At the device a tapped hole is near the connector which should be used to attach the lead

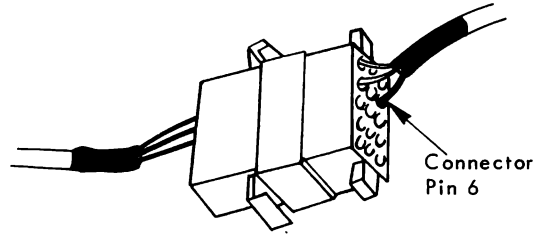


Connecting to the Serial Control Unit ASR33

V24 Interface: At the device the cable is attached to the metal cover of the transformer situated in the base of the ASR.

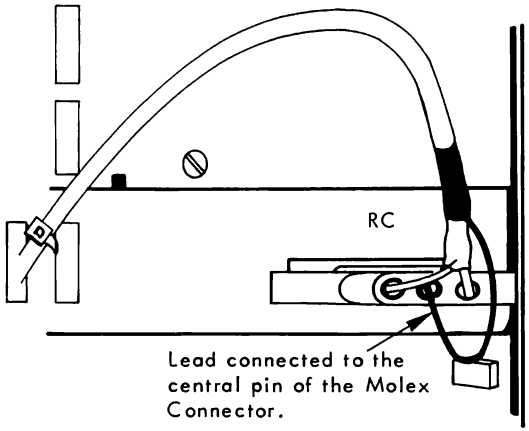


Current Loop Interface: At the device when using the extension cable the connection between Molex connector pin 6 and the ASR mechanical ground is made with the cable shielding. Note that under no circumstances must the Molex connectors be situated outside the ASR base.

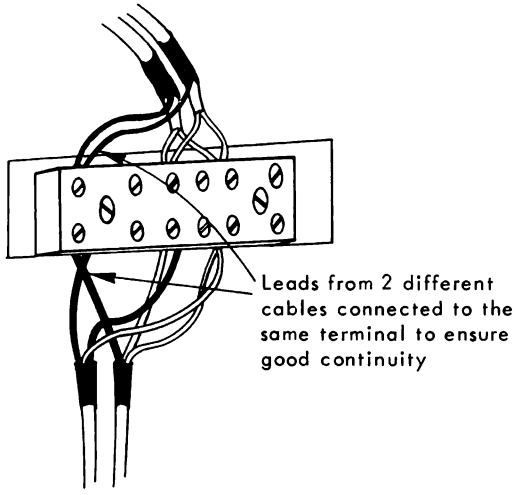


Connecting the Remote Control Cables

Rack Side: The shielding is connected to the central pin of the Molex connector. Note that for racks M1, M2, M4, M5, E2 and K7 a link connects the mechanical ground to the central pin of the Molex connector.

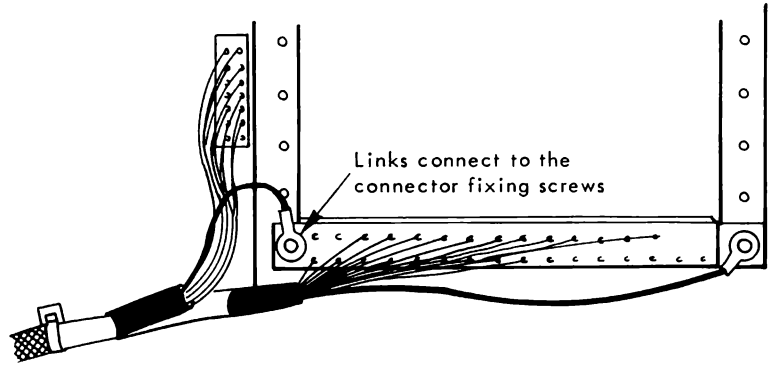


Connection at the Cabinet Terminal Box: When connecting the shielding of two cables at the Terminal Box connect to the same terminal to ensure good continuity.

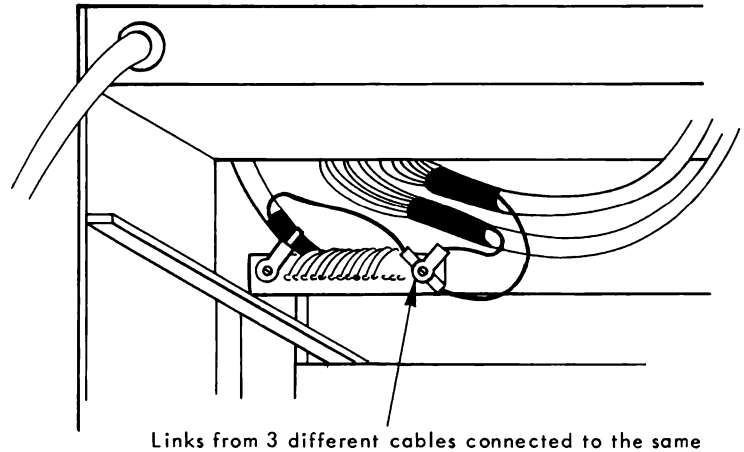


Connecting the Break Cables

At the Extension Rack:



At the Cassette Rack:



SECTION III

POWER SUPPLY

3.1 DESCRIPTION

The power supply provides six regulated d.c. voltages and three logic signals. Four of the voltages are used by the logic circuits of the cards and the other two voltages are used exclusively by the memory modules. The power supply also has battery back-up to maintain the two memory voltages during a power failure. The battery is recharged when power is restored.

3.2 Input Voltages

The power supply operates on voltages of 115V, 220V, or 240V ±10% single phase, at either 50Hz ±2Hz or 60Hz ±3Hz.

3.3 Output Voltages

The output voltages and maximum currents from the power supply are:

- +5V (Logic), 15A
- +12V (Logic), 2A
- -12V (Logic), 1.2A
- +10V (Logic), 0.6A

±3% due to ±10% mains variation and to load variations from 10% to 100% of full load.

- +5VM (Memory), 1.2A
- +12VM (Memory), 0.93A

±3% due to ±10% mains variation and to load variations from 30% to 100% of full load.

3.4 Logic Signals

The three logic signals produced by the power supply are Power Fail (PWFN), Reset Logic (RSLN) (both active low signals), and the Real Time Clock (RTCN) which provides a 1µs ±30% pulse at the frequency of the mains.

3.5 MECHANICAL

The power supply is mounted in the right-hand half of the CPU chassis and comprises four sub-assemblies (excluding the battery pack). These are:

- Mains Transformer chassis, which also contains the mains filter, the rectifiers, and the reservoir capacitors.
- The +5V Regulator card (REG +5V), containing +5V, +12V, and -12V regulators which supply logic voltages to the backpanel bus.
- The Memory Supply card (ALIMOS), containing +5VM, +12VM, +10V regulators and a +12V auxiliary supply regulator.
- The Sequence card (SEC Z), which provides the three logic signals and the battery back-up circuits.

The mains transformer chassis is secured to the bottom of the CPU chassis by six screws, the nuts being fastened to the mains transformer chassis. The three power supply cards are slide mounted above the mains transformer chassis and there is a slot provided for the battery pack. Figure 3-1 shows the physical position of the sub-assemblies.

3.6 BLOCK DIAGRAM

Figure 3-2 is a block diagram of the power supply sub-assemblies and their interconnections with a detailed schematic diagram of the Mains Transformer chassis. Figures 3-6 to 3-8 are the schematic diagrams for the circuit cards.

3.7 MAINS TRANSFORMER CHASSIS

The a.c. mains is connected to the power supply input terminals via fuse F01 and is then fed through a line filter to the mains transformer.

3.8 Mains Voltage Connections

The power supply can operate on either 240V, 220V, or 115V and the connections are shown in Figure 3-2.

3.9 Rectifier and Transformer Protection

Fuses F02 to F06 protect the two diode bridge packs and the transformer in case of a failure by one of the regulator circuits. When the power supply is operating normally these fuses can be ignored because the regulators provide electronic protection.

3.10 CONTROL AND PROTECTION CHIP (TDA2640 or TDA1060)

The control and protection for the +5V logic, the +5VM, and the +12VM voltages are based on three of these chips. These provide control for the output transistors, soft-start and cut-out circuits, overvoltage and overcurrent protection, low supply voltage protection, and loop fault protection. Figure 3-3 shows a block diagram of the chip and its pin numbers. The component designations on the block diagram, and in the following description, are simply the pin numbers to which they are connected and should not be confused with the real designations on the schematic diagrams.

3.11 Miller Integrator

This circuit provides a free running saw tooth waveform whose frequency is dependant on the value of C3-4 and R4-5 (20kHz at full load).

3.12 Comparator

This circuit compares the voltage on pin 10 with the reference voltage on pin 9, the result being used to stabilize the output voltage.

3.13 Pulse Width Modulator

The width of the output pulses is modulated by the comparator output. The modulator output is also controlled by the soft-start and cut-out circuit. Because this circuit is designed to work primarily with voltages greater than 6V, the sense voltage for the two regulators is fed via a diode.

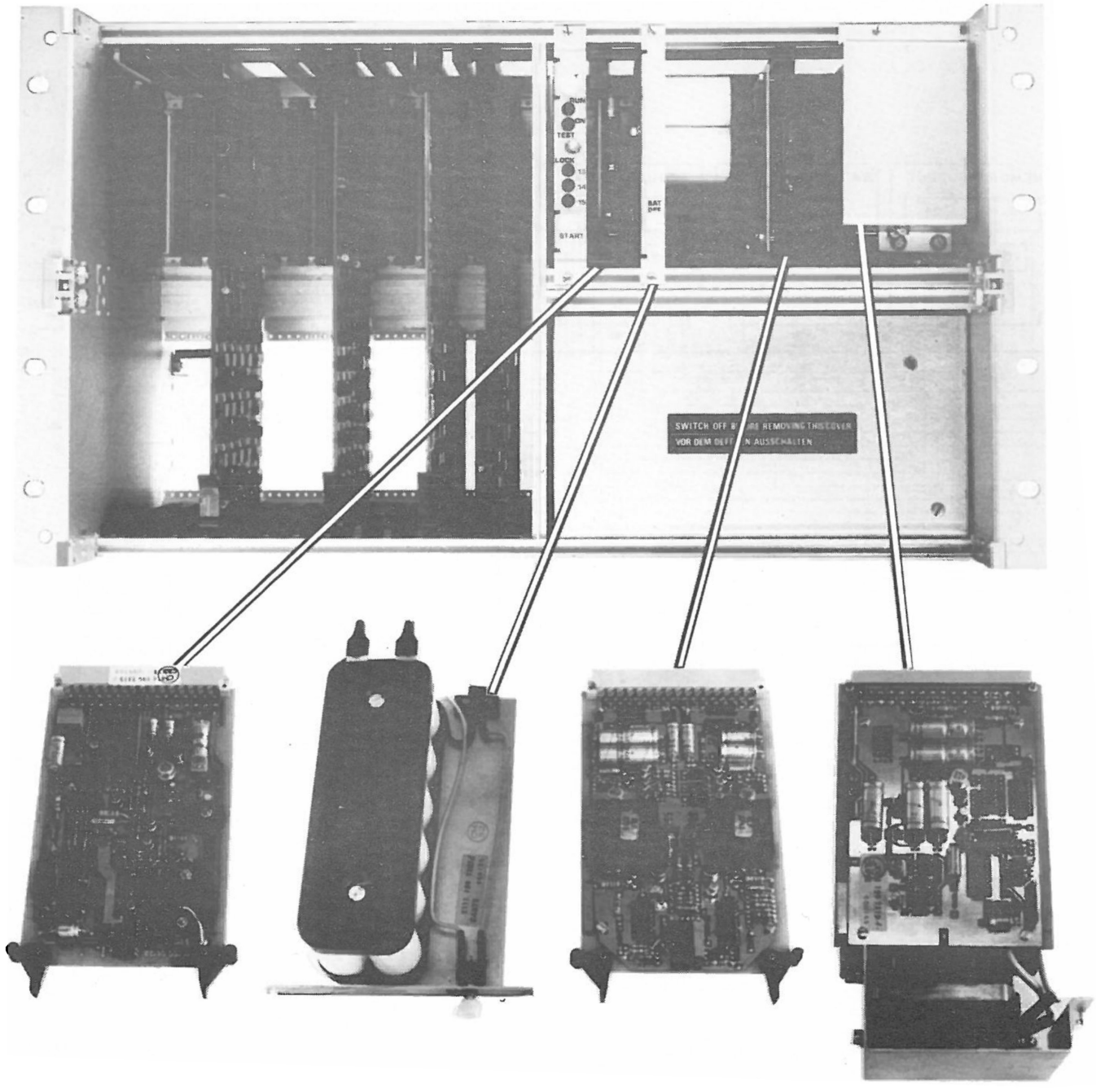


Figure 3-1 Mechanical Layout of Sub-Assemblies

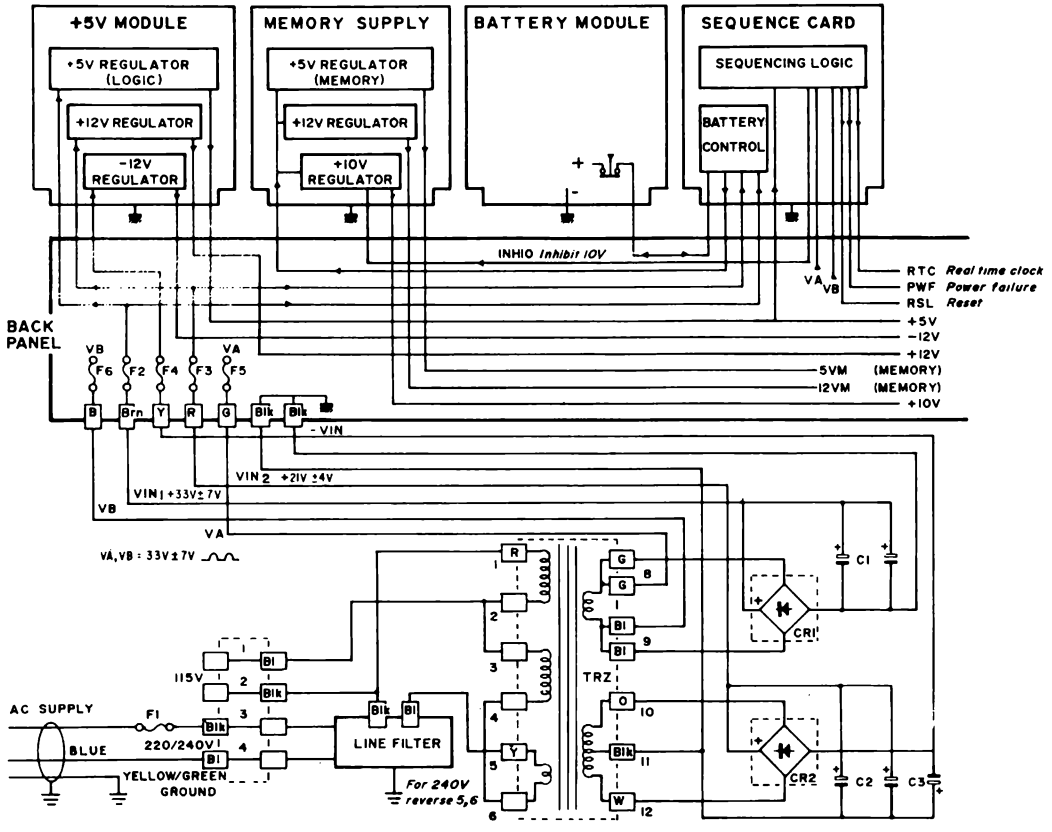
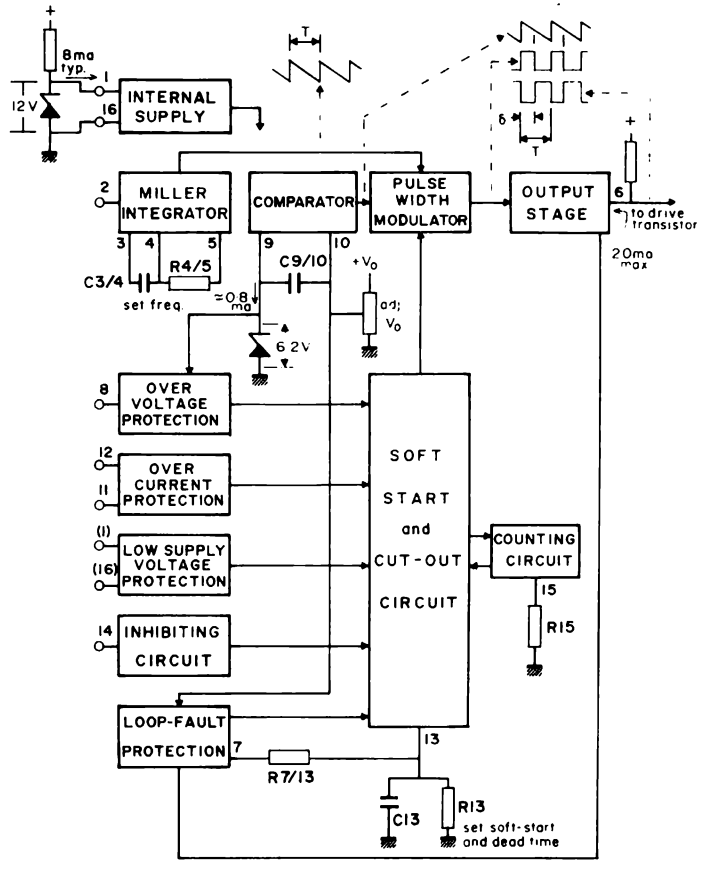


Figure 3-2 Power Supply Block Diagram



NOTE: for TDA1060 see page A-13

Figure 3-3 Block Diagram of TDA 2640 Chip

3.14 Output Stage

The output stage is used to drive the switching transistor via associated transistor circuits, the number and complexity of these circuits depends on the input requirements of the switching transistor.

3.15 Soft-Start and Cut-Out

Commands from the protection circuits are fed to the soft-start and cut-out circuit which in turn controls the pulse width modulator. A fault condition causes instantaneous switch off; then, after a dead time, determined by the product of R13 and C13, the soft start is initiated.

3.16 Counting Circuit

The counting circuit is not used by this power supply and is suppressed by R15.

3.17 Overvoltage Protection

This circuit is not used by this power supply, protection being provided by an external circuit.

3.18 Overcurrent Protection

If the voltage between pins 11 and 12 exceeds one volt the overcurrent protection circuit is tripped, which activates the cut-out and soft-start procedure.

3.19 Low Supply Voltage Protection

This circuit protects the switching transistor if the supply voltage falls below 8V.

3.20 Loop Fault Protection

This circuit causes immediate switch off if the feedback voltage on pin 10 falls to zero either because of an open or short circuit feedback loop.

3.21 Inhibiting Circuit

This circuit inhibits the chip and the regulator if the voltage on pin 14 is less than 5V.

3.22 +5V REGULATOR CARD

This card provides the +5V, +12V, and -12V regulated supplies that are used by both the power supply and CPU logic circuits. The schematic diagram of the card is shown in Figure 3-6.

3.23 +12V and -12V Regulators

The three terminal regulators used for these supplies have internal current limiting and shut-down facilities.

3.24 +5V Regulator

The +5V supply is a switching regulator with control and protection, except for overvoltage protection, provided by a TDA 2640 chip (TDA1060 for later versions). The output voltage is adjusted by PR1.

3.25 Overvoltage Protection. Overvoltage protection is provided by zener diode CR1 which monitors the current through the shunt and operates the overcurrent circuit of the chip if the breakdown voltage of 7.5V is exceeded.

3.26 Overcurrent Protection. The overcurrent circuit of chip U1 is driven by comparator U2 for the +5V supply (the overcurrent adjustment is by R2 which is normally set for a value of 16.5A at 25°C ambient temperature) and transistors Q5 and Q6 which monitor the Vin1 and Vin2 lines respectively. If either of these three circuits cause the voltage on pin 12 to exceed one volt the overcurrent circuit trips out the card.

3.27 MEMORY SUPPLY CARD

This card provides the +5VM and +12VM memory voltages, the +10V logic voltage for the CPU, and a +12V series regulated supply to drive the logic on the card. Figure 3-7 shows the schematic diagram of the card.

3.28 +5VM Regulator

This supply has a switching regulator with control and protection, except for overvoltage protection, provided by a TDA 2640 chip. The output voltage is adjusted by P1.

3.29 Overvoltage Protection. The overvoltage protection is by zener diode CR9. When the breakdown voltage of 7.5V is exceeded, the increase of current through R23 causes the overcurrent circuit of the TDA chip to operate and cut off the supply.

3.30 Overcurrent Protection. The overcurrent circuit of chip U1 is driven by Q13 for the +5VM supply (the overcurrent value is fixed at 2A by R23 and R8) and Q7 which monitors the Vin MOS line. If either of these two circuits cause the voltage on pin 12 to exceed one volt the overcurrent trips and cuts off the supply.

3.31 +12VM Regulator

This supply has a switching regulator with control and protection (it has no overvoltage protection) provided by TDA 2640 chip U2. The output voltage is adjusted by P3.

3.32 Overcurrent Protection. The overcurrent circuit of chip U2 is driven by Q14 which samples the current accross R24. The overcurrent value is fixed at 2.6A by R24 and R6.

3.33 +10V Regulator

This supply is series regulated and the output voltage is adjusted by P2. During normal operation the INH10N line is floating via relay contacts on the Sequence Card and Q10 is on. When a power failure occurs or a normal power off takes place, the INH10N line is grounded and Q10 is switched off which cuts off the 10V supply and prevents unnecessary drain on the battery.

3.34 SEQUENCE CARD

This card contains the Power Failure, Reset, and Real Time Clock logic signals, the Inhibit logic for the +10V supply and the Battery Control facilities for the memory supplies. Figure 3-8 shows the schematic diagram of the card.

3.35 Power Fail/Reset Logic

The VA and VB inputs to the card drive the transistors of the mains detector circuit which is set up to detect mains failures longer than 10ms by adjusting R2. The output from this circuit drives two monostables which produce the Power Failure (PWFN) and Reset Logic (RSLN) signals in the correct sequence during either normal power ON/OFF or power failure conditions; it also controls the INH10N signal to the Memory Supply card.

3.36 Real Time Clock

The VA input is used to drive the transistor and monostable multivibrator to produce 1μS pulses at mains frequency (every 20mS at 50Hz). The pulse width is determined by R29 and C15. The output signal (RTCN) is sent to the CPU via the control panel and is used to set one of the internal interrupt F/Fs.

3.37 +10V Inhibit

This circuit is controlled by the Power Fail/Reset logic. When the power supply is plugged into the mains supply, signals from the two logic circuits enable the transistors to energise and latch relay K1 whose contacts open removing the inhibit from the +10V supply and enabling the RSLN line to go high. If the power supply is unplugged or a power failure occurs, the PWFN and RSLN logic cuts off the latching transistor of the relay whose contacts close grounding the RSLN line and inhibiting the +10V supply.

3.38 Battery Control

The purpose of this circuit is to supply the Vin MOS to the Memory Supply Card (either from the Mains Transformer Chassis or the Battery Pack), a Battery Protection circuit, and a Battery Charger circuit.

3.39 Mains Operation. Vin 2 provides the Vin MOS supply and also energises and latches on relay K2. With the relay contacts closed the battery circuit is completed but does not affect Vin MOS due to the voltage differential.

3.40 Battery Operation. If the power supply is unplugged or there is a power failure, the battery pack comes into operation as soon as the voltage through CR12 falls below 16V (approximately) maintaining the Vin MOS supply and holding relay K2 latched on. When the battery voltage drops below 13.5V the battery protection circuit operates and drops out relay K2 disconnecting the battery supply.

3.41 Battery Protection. This circuit prevents the battery from discharging below a level which would cause permanent damage to the battery pack. When the voltage from the battery drops below 13.5V the zener diode CR9 stops working thus cutting off the transistors and dropping out relay K2. A push OFF switch is provided on the battery pack which will drop out relay K2 when the power supply is unplugged.

3.42 Battery Charger. Vin 1 supplies this circuit whose output is controlled by zener diode CR7 to give a constant trickle charge of 50mA. It takes 48 hours to recharge a fully discharged battery.

3.43 POWER ON/OFF SEQUENCE

The timing for the power ON/OFF sequence for the power supply output voltages and the Powerfail/Reset logic is shown in Figure 3-4.

3.44 BACKPANEL CONNECTIONS

The output connections from the Mains Transformer Chassis to the backpanel are made with Faston connectors and the supplies may be identified by the coloured sleeves. These are:

- Brown (Brn) -- Vin 1
- Red (R) -- Vin 2
- Yellow (Y) -- Vin
- Blue (Bl) -- VB
- Green (G) -- VA
- Black (Blk) -- Ground

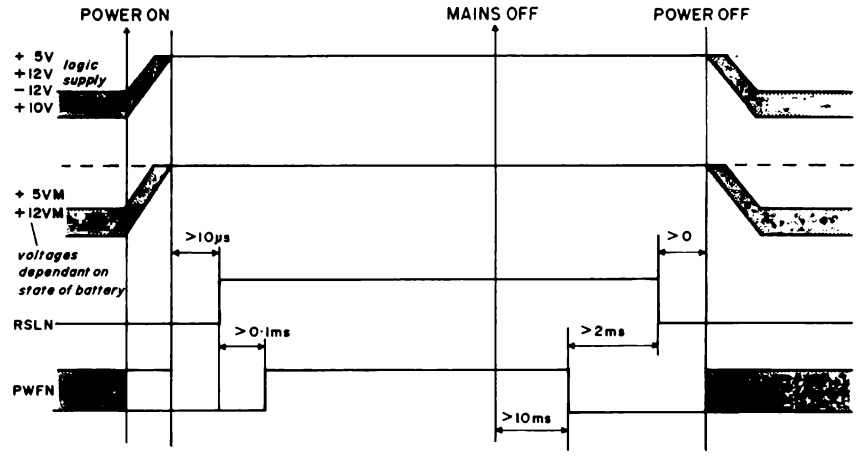


Figure 3-4 Power Sequence Timing

3.45 ADJUSTMENTS

There are six adjustments provided to set up the power supply. Two of the adjustments are made using fixed resistors selected on test; the other four are made by potentiometers which are accessible from the front of the rack with the cards in position. The two fixed resistors can be easily recognised because they are mounted on stand-off turret lugs; Figure 3-5 shows the position of the potentiometers. WARNING: The following information on adjustments should only be made after an emergency repair when no spare card is available. When a spare card is available the repaired card should be sent back to the factory to be accurately adjusted. All adjustments for voltage and current should be made using a digital type meter with four or more digits or a meter with an accuracy better than 0.5%, and should be made at an ambient temperature of 25°C. To adjust:

- +5VL output voltage select PR1 = 5V±3%
- +5VL overcurrent select R2 = 16.5A±3%
- +5VM output voltage select P1 = 5V±3%

- +12VM output voltage select P3 = 12V±3%
- +10VL output voltage select P2 = 10V±3%
- PWFN 10mS delay select R2.

3.46 COMPONENTS

Figures 3-9 to 3-12 show the component layout of the power supply sub-assemblies and Tables 3-1 to 3-4 list the components. Figure 3-13 shows the battery pack card.

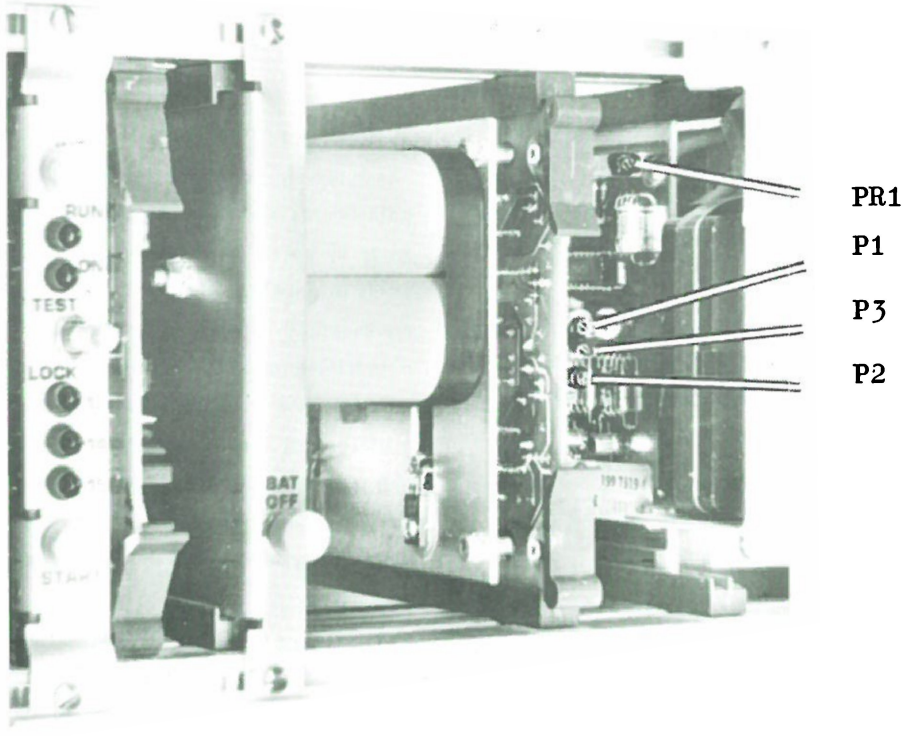


Figure 3-5 Position of Potentiometers

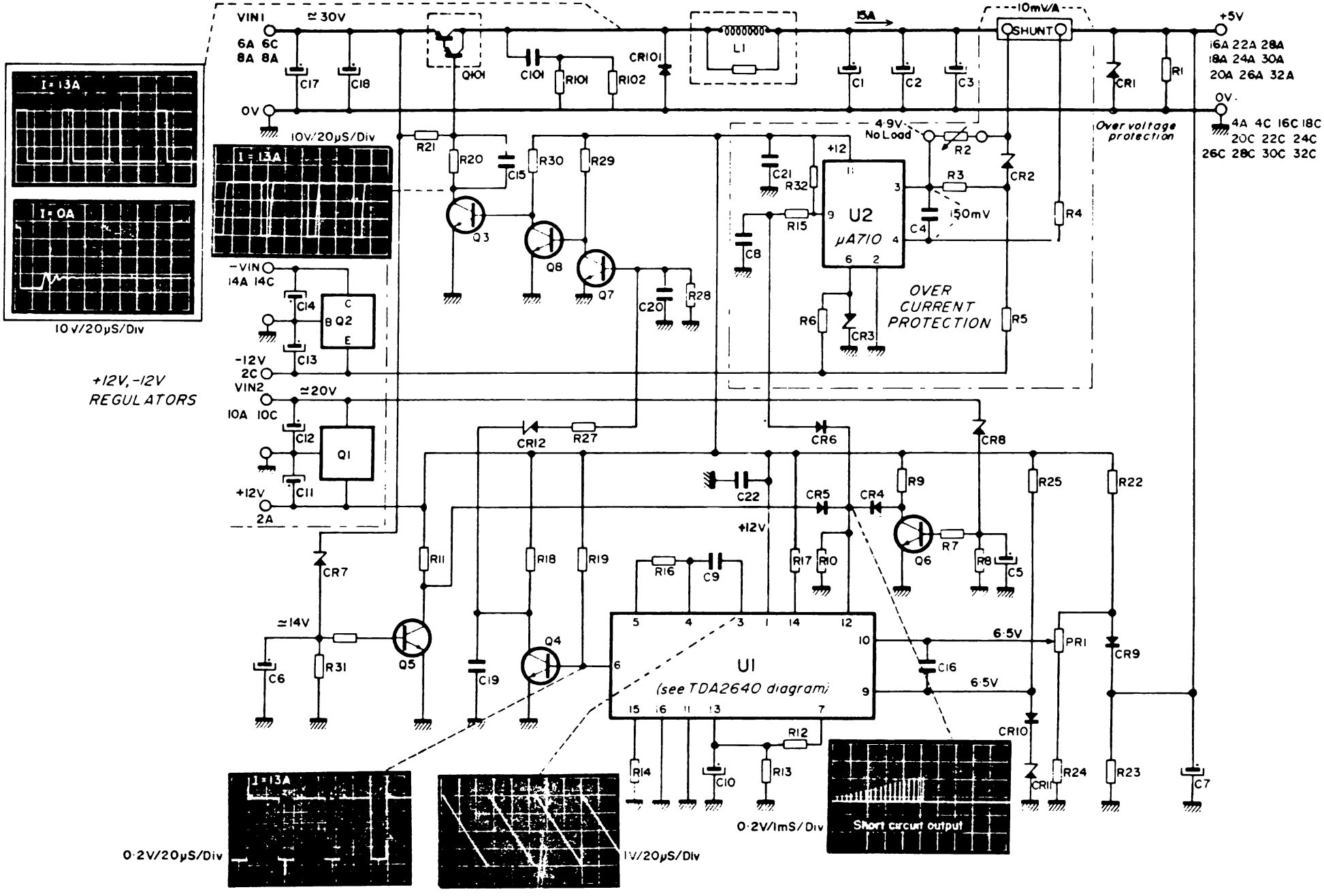


Figure 3-6 -5V Regulator Card Schematic

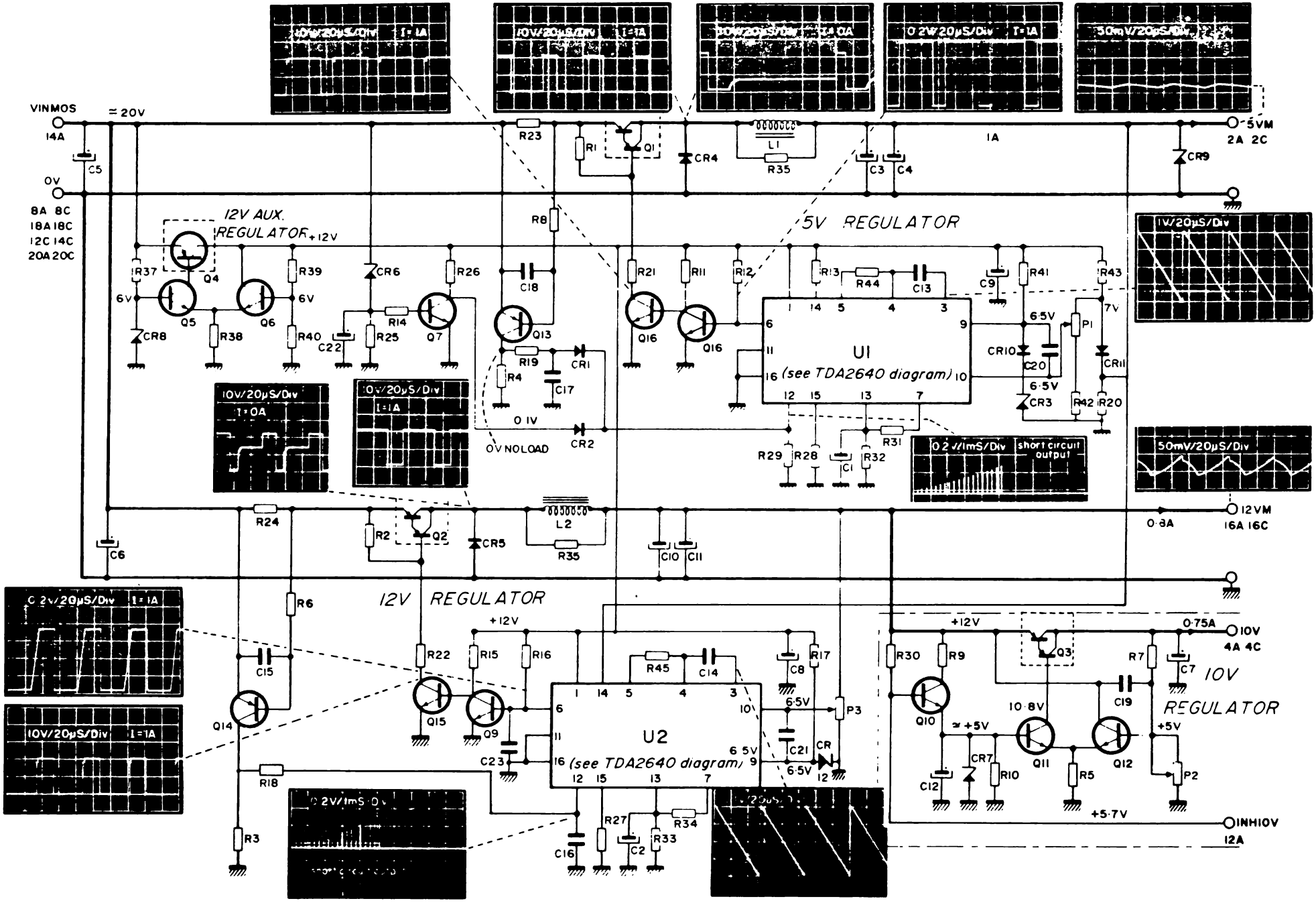


Figure 3-7 Memory Supply Card Schematic

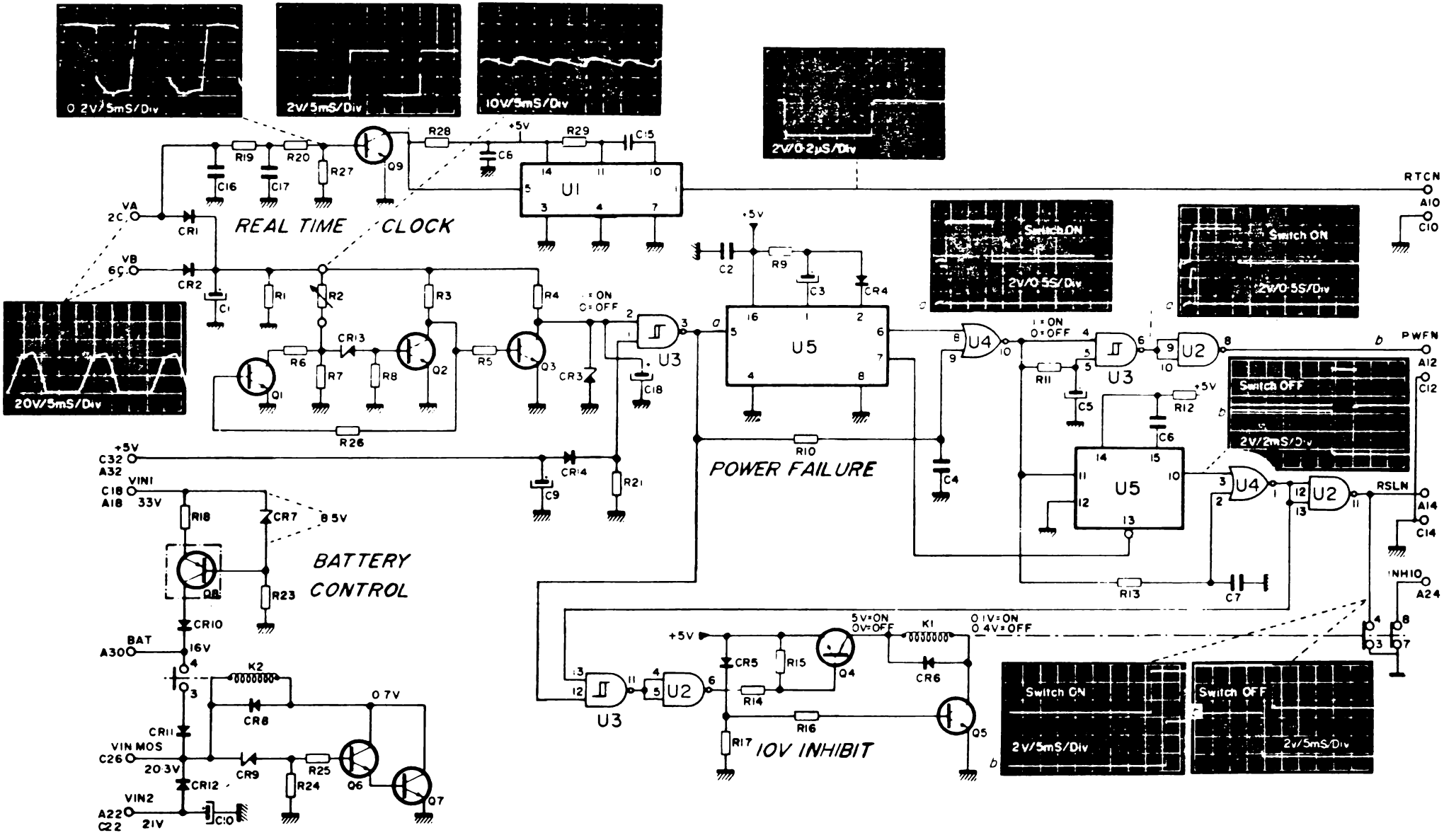


Figure 3-8 Sequence Card Schematic

Table 3-1 Mains Transformer Chassis Parts List

Reference	Description	12NC Code
2.	Transformer TRZ4.	
CR01, 02.	Diode Bridge MDA 980-2.	
5.	Fuse Holder 23 321 E.	
6.	Fuse Holder 231 347.	
F01.	Fuse D8/TD 2.	
F02.	Fuse D1/6.3.	
F03.	Fuse D1/4.	
F04.	Fuse D1/0.8.	
F05, 06.	Fuse D1/0.2.	
11.	Mains Filter F1126/39.	
12.	Mains Cable 12.	
C01.	Capacitor 2x5000μF, 40V.	
C03.	Capacitor 4700μF, 25V.	
C02.	Capacitor 2x7500μF, 25V.	

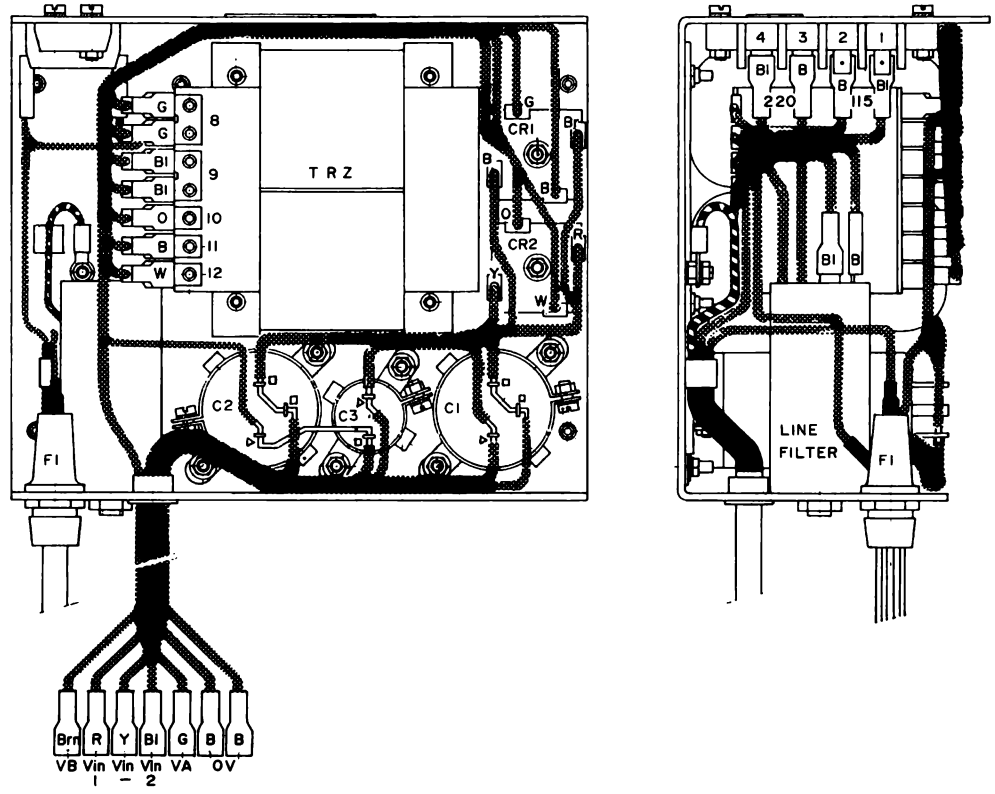


Figure 3-9 Mains Transformer Chassis Component Layout

Table 3-2 +5V Regulator Card Parts List

Reference	Description	12NC Code
1	Printed circuit	5111 100 06112
U1	Integrated circuit TDA 2640	
U2.	Integrated circuit µA 710 DC	
C1,2,3.	Capacitor 470µF, 10V, Fitco.	
C4.	Capacitor 150pF, ±2%, ceramic.	
C5,6,10.	Capacitor 1µF, 35V, CTS13.	
C7.	Capacitor 22µF, 25V, Fitco.	
C8.	Capacitor 3.9µF, ceramic.	
C9.	Capacitor 1.5nF, 400V, ±1%, CKM111.	
C11-14.	Capacitor 10µF, 25V, Fitco.	
C15.	Capacitor 2.7nF, ±10%, ceramic.	
C16.	Capacitor 1nF, ±10%, ceramic.	
C17-18.	Capacitor 100µF, 63V, Fitco.	
C19.	Capacitor 0.0033µF, 400V, ±10%, CM111.	
C20.	Capacitor 100pF, ceramic.	
C21,22.	Capacitor 10nF, ceramic.	
CR1.	Diode Transzorb ICTE5.	
CR2.	Diode BZX79 C6V2.	
CR3-12.	Diode BZX79 C4V7.	
CR4-6.	Diode BAX12.	
CR7.	Diode BZX79 C16.	
CR8.	Diode BZX79 C12.	
CR9,10.	Diode BZX75 C2V1.	
CR11.	Diode IN 750 A.	
R1-21.	Resistor 100Ω, 0.5W, ±5%.	
R2.	Resistor 316Ω, 0.125W, 1%.	
R3.	Resistor 10KΩ, 0.125W, ±1%.	
R4,7,9-11,15.	Resistor 1KΩ, 0.25W ±5%.	
R5,6,8,17,30.	Resistor 2KΩ, 0.25W, ±5%.	
R12.	Resistor 180KΩ, 0.25W, ±5%.	
R13.	Resistor 390KΩ, 0.25W, ±5%.	
R14-31.	Resistor 4.7KΩ, 0.25W, ±5%.	
R16.	Resistor 38.3KΩ, 0.125W, ±1%.	
R18.	Resistor 3.48KΩ, 0.125W, ±1%.	
R19.	Resistor 6.2KΩ, 0.25W, ±5%.	
R20.	Resistor 390Ω, 0.5W, ±5%.	
R22.	Resistor 383Ω, 0.125W, ±1%.	
R23.	Resistor 220Ω, 0.25W, ±5%.	
R24.	Resistor 2.61KΩ, 0.125W, ±1%.	
R25.	Resistor 511Ω, 0.125W, ±1%.	
R26,29.	Resistor 2.7KΩ, 0.25W, ±5%.	
R27.	Resistor 100Ω, 0.25W, ±5%.	
R28.	Resistor 680Ω, 0.25W, ±5%.	
R32.	Resistor 10KΩ, 0.25W, ±5%.	
Q3.	Transistor 2N2219.	
Q4-8.	Transistor BSX20.	
PR1.	Potentiometer 500Ω, 2600P 501.	

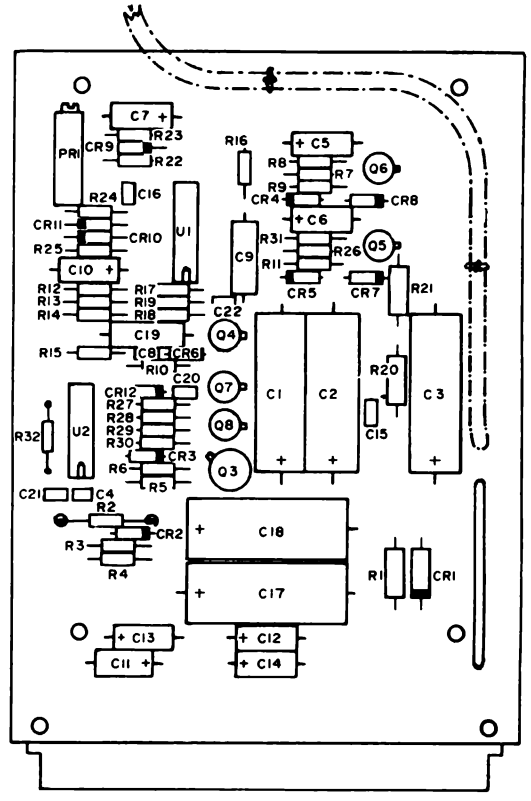


Figure 3-10 +5V Regulator Card Component Layout

Table 3-3 Memory Supply Card Parts List

Reference	Description	12NC Code
1.	Printed circuit	5111 100 06102
7.	Heat Sink.	
U1,U2.	Integrated circuit TDA 2640.	
CR1,2.	Diode BAX12.	
CR3.	Diode 1N750A.	
CR4,5.	Diode MR820.	
CR6.	Diode BZX79C11.	
CR7.	Diode BZX79 C4V7.	
CR8.	Diode BZX79 C6V2.	
CR9.	Diode ICTE5.	
CR10,11.	Diode BZX75 C2V1.	
CR12.	Diode IN823.	
Q1,2,3.	Transistor BD262.	
Q4.	Transistor BD236.	
Q5-12.	Transistor BSX20.	
Q13,14.	Transistor ZN2905.	
Q15,16.	Transistor BSX60.	
C1,2.	Capacitor 1µF, 35V	CTS13.
C3,4.	Capacitor 220µF, 10V,	Fitco.
C5,6.	Capacitor 47µF, 25V,	Fitco.
C7,8,9.	Capacitor 10µF, 25V,	Fitco.
C10,11.	Capacitor 220µF, 25V,	Fitco.
C12.	Capacitor 4.7µF, 16V,	Fitco.
C13,14.	Capacitor 1500pF, 63V,	Micropoco.
C15.	Capacitor 2.7nF,	ceramic.
C16,17,23.	Capacitor 10nF,	ceramic.
C18,20,21.	Capacitor 1nF,	ceramic.
C19.	Capacitor 100pF,	ceramic.
C22.	Capacitor 3.3µF, 25V,	Fitco.
R1,2.	Resistor 300a, 0.25W, ±5%.	
R3,4.	Resistor 330a, 0.25W, ±5%.	
R5.	Resistor 390a, 0.25W, ±5%.	
R6,7,8.	Resistor 510a, 0.25W, ±5%.	
R9.	Resistor 680a, 0.25W, ±5%.	
R10.	Resistor 910a, 0.25W, ±5%.	
R11,13,14,17.	Resistor 1Ka, 0.25W, ±5%.	
R18,19.	Resistor 2.4Ka, 0.25W, ±5%.	
R20.	Resistor 220a, 0.25W, ±5%.	
R21,22.	Resistor 1.5Ka, 0.25W, ±5%.	
R23.	Resistor 0.33a, ±5%,	Ref.224E.
R24.	Resistor 0.21a, ±5%,	Ref.224E.
R25,26,29.	Resistor 2Ka, 0.25W, ±5%.	
R27,28.	Resistor 4.7a, 0.25W, ±5%.	

Table 3-3 contd.

Reference	Description	12NC Code
R30.	Resistor 6.2Ka, 0.25W, ±5%.	
R31.	Resistor 82Ka, 0.25W, ±5%.	
R32,33.	Resistor 390Ka, 0.25W, ±5%.	
R34.	Resistor 180Ka, 0.25W, ±5%.	
R35,36.	Resistor 510a, 0.5W ±5%.	
R37.	Resistor 1Ka, 0.5W, ±5%.	
R38,43.	Resistor 383a, 0.125W, ±1%.	
R39.	Resistor 1Ka, 0.125W, ±1%.	
R40.	Resistor 1.1Ka, 0.125W, ±1%.	
R41.	Resistor 511a, 0.125W, ±1%.	
R42.	Resistor 2.61Ka, 0.125W, ±1%.	
R44,45.	Resistor 38.3Ka, 0.125W, ±1%.	
R12,15,16.	Resistor 10Ka, 0.25W, ±5%.	
L1.	Inductance SLF 2741.	
L2.	Inductance SLF 2731.	
P1.	Potentiometer 2600 P 501.	
P2,3.	Potentiometer 2600 P 102.	

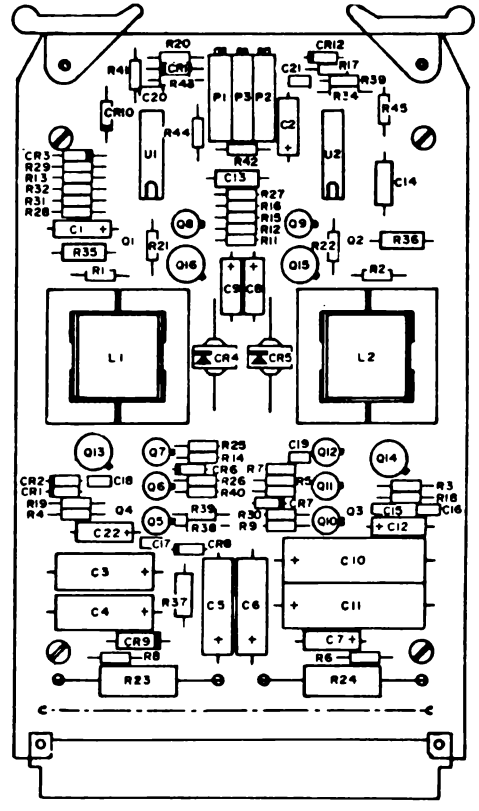


Figure 3-11 Memory Supply Card Component Layout

Table 3-4 Sequence Card Parts List

Reference	Description	12NC Code
1.	Printed circuit	5111 100 06122
U1.	Integrated circuit 74121.	
U2.	Integrated circuit 7438.	
U3.	Integrated circuit 74132.	
U4.	Integrated circuit 7402.	
U5.	Integrated circuit 9602.	
R19,20.	Resistor 4.3Kn, 0.25W, 5%.	
R27.	Resistor 5.1Kn, 0.25W, 5%.	
R1.	Resistor 3.9Kn, 0.5W, 5%.	
R2.	Resistor 6.19Kn, 0.25W, 1%.	
R3,4.	Resistor 10Kn, 0.25W, 5%.	
R6.	Resistor 10Kn, 0.25W, 1%.	
R7.	Resistor 3.83Kn, 0.25W, 1%.	
R8.	Resistor 820n, 0.25W, 5%.	
R9.	Resistor 31.6Kn, 0.125W, 1%.	
R10,11,13.	Resistor 100n, 0.25W, 5%.	
R12.	Resistor 19.6Kn, 0.125W, 1%.	
R5,14,24,26.	Resistor 1Kn, 0.25W, 5%.	
R15.	Resistor 2Kn, 0.25W, 5%.	
R16.	Resistor 240n, 0.25W, 5%.	
R17.	Resistor 270n, 0.25W, 5%.	
R21.	Resistor 316n, 0.25W, 1%.	
R29.	Resistor 9.09Kn, 0.125W, 1%.	
R23.	Resistor 4.7Kn, 0.25W, 5%.	
R25-28.	Resistor 510n, 0.25W, 5%.	
R18.	Resistor 150n, 0.5W, 5%.	
CR11.	Diode 1N 5821.	
CR1,2,4,6,8,10.	Diode BAX12.	
CR3.	Diode BZX79 C4V7.	
CR5.	Diode BZX75 C3V6.	
CR7.	Diode BZX79 C8V2.	
CR12.	Diode BYX49 300R.	
CR13.	Diode BZX79 C6V2.	
CR14.	Diode BZX75 C2V1.	
CR9.	Diode BZX79 C11.	
C3.	Capacitor 100µF, 10V.	
C1.	Capacitor 22µF, 63V, Fitco.	
C2,8,12-14.	Capacitor 10nF, ceramic.	
C5.	Capacitor 33µF, 10V, CTS13.	
C4-7.	Capacitor 1nF, ceramic.	
C6.	Capacitor 0.47µF, 40V, ±1%, CKB68.	

Table 3-4 contd.

Reference	Description	12NC Code
C9.	Capacitor 22µF, 10V, Fitco.	
C10-18.	Capacitor 10µF, 25V, Fitco.	
C15.	Capacitor 150pF, ±2%, ceramic.	
C16.	Capacitor 0.1µF, ±10%, MPR.	
C17.	Capacitor 0.01µF, ±10%, MPR.	
Q1-3,5,6,9.	Transistor BS x 20.	
Q7.	Transistor 2N 2219.	
Q4.	Transistor BS x 29.	
Q8.	Transistor BD x 78 (TO220).	
K1.	Relay MR 2ME 15005.	
K2.	Relay MR MC 15002.	
4.	Heat Sink LB 65 81 CB.	

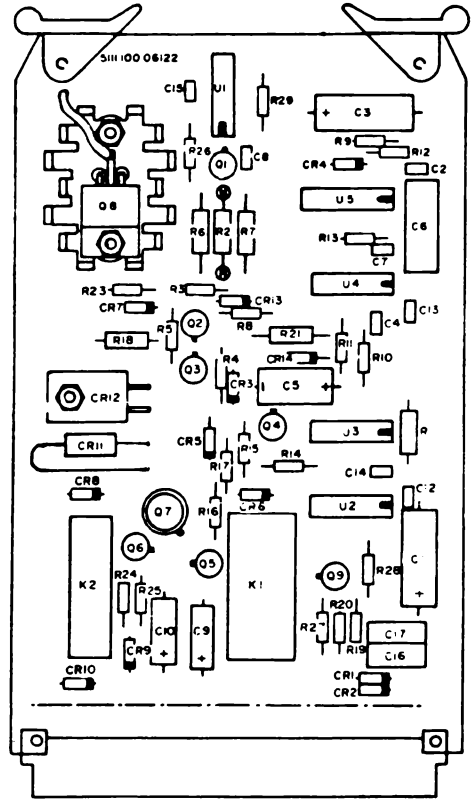


Figure 3-12 Sequence Card Component Layout

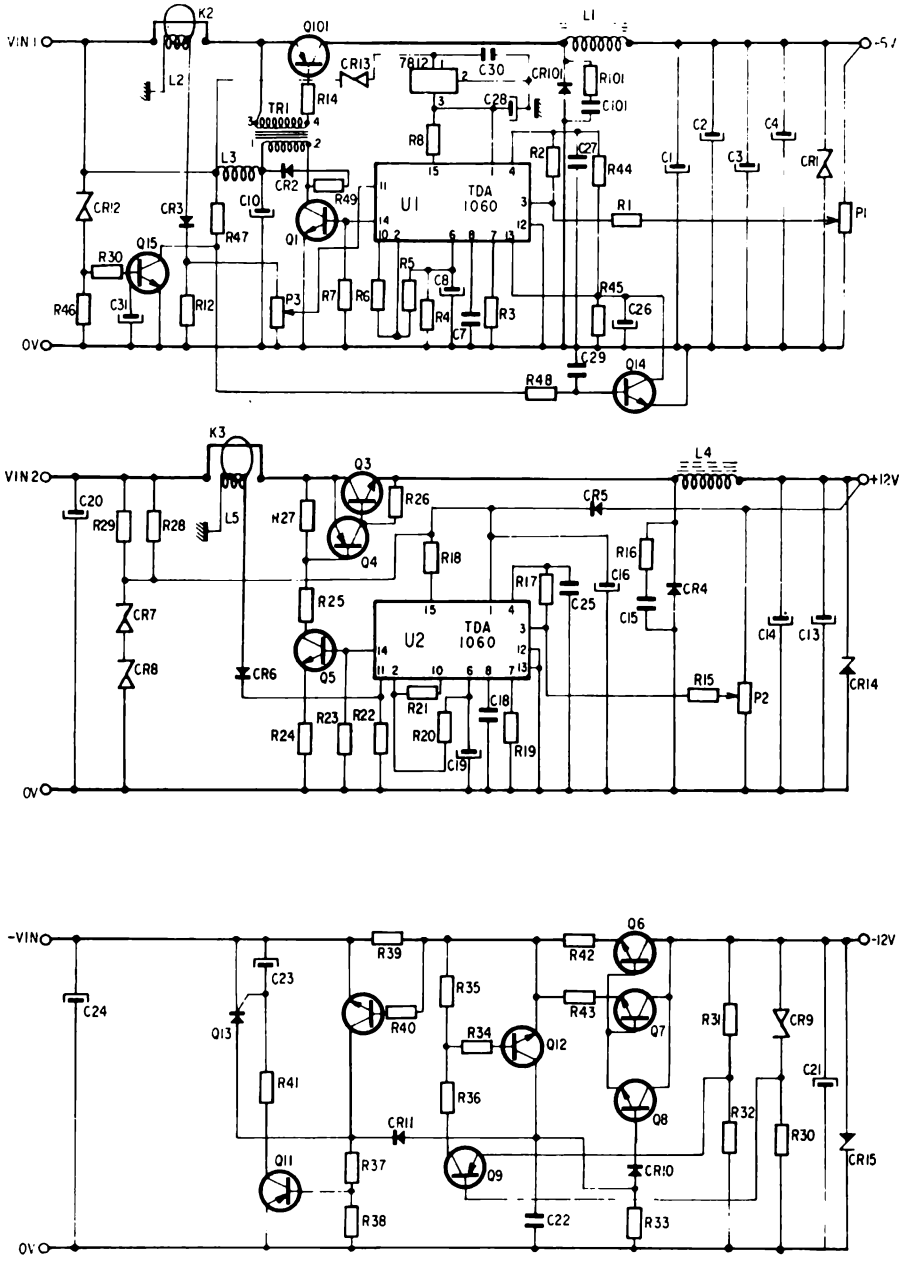


Figure 3-13 +5VR Regulator Card Schematic

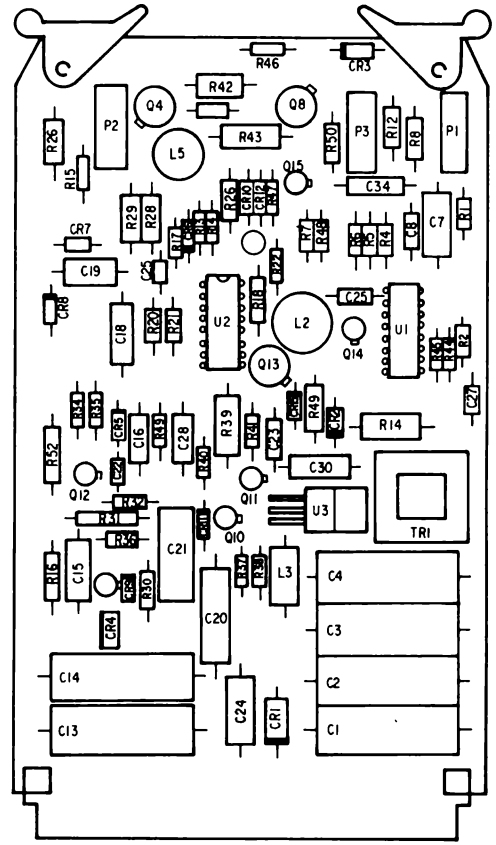


Figure 3-14 +5VR Regulator Card Component Layout

Table 3-5 +5VR Regulator Card Parts List

Reference	Description	12NC Code
	Printed circuit	5111 100 06553
U1,2.	Integrated circuit TDA1060.	
U3.	Integrated circuit 7812 (TO220).	
Q10,12,14.	Transistor BSX20.	
Q4.	Transistor 2N2905.	
Q5.	Transistor 2N2222.	
Q8.	Transistor 2N2219.	
Q9-11.	Transistor 2N2906.	
Q15.	Transistor BSX21.	
Q101.	Transistor 2N4399 (TO3).	
Q1,3,6,7.	Transistor BDX77 (TO220).	
Q13.	Thyristor 2N223.	
CR1.	Diode ICTE5.	
CR4.	Diode MR820.	
CR2,3,5,6,11.	Diode BAX12A.	
CR7.	Zener Diode BZX79C5V6.	
CR8.	Zener Diode BZX79C7V5.	
CR10.	Zener Diode BZX75C2V1.	
CR9.	Diode 1N823.	
CR12.	Diode BZX79C20.	
CR13.	Diode BZX79C5V1.	
CR14,15.	Diode ICTE12.	
CR101.	Diode 1N3900R.	
R30.	Resistor 750Ω, 0.25W, 5%.	
R31.	Resistor 383Ω, 0.25W, 1%.	
R32.	Resistor 261Ω, 0.25W, 1%.	
R34,40.	Resistor 100Ω, 0.25W, 5%.	
R35.	Resistor 680Ω, 0.25W, 5%.	
R4,36.	Resistor 2.7KΩ, 0.25W, 5%.	
R37,41.	Resistor 68KΩ, 0.25W, 5%.	
R1,38,48,50.	Resistor 10KΩ, 0.25W, 5%.	
R39.	Resistor 0.22Ω, 10%, RWM4x10.	
R42,43.	Resistor 0.1Ω, 10%, RWM4x10.	
R27.	Resistor 1KΩ 0.25W, 5%.	
R16.	Resistor 200Ω, 0.5W, 5%.	
R2,17.	Resistor 100KΩ, 0.25W, 5%.	
R18.	Resistor 510Ω, 0.5W, 5%.	
R5,20.	Resistor 2KΩ, 0.25W, 5%.	
R6,21.	Resistor 6.2KΩ, 0.25W, 5%.	
R22.	Resistor 21.5Ω, 0.125W, 1%.	
R23.	Resistor 560Ω, 0.25W, 5%.	
R24.	Resistor 300Ω, 0.25W, 5%.	
R25.	Resistor 1.5KΩ, 0.5W, 5%.	

Table 3-5 contd.

Reference	Description	12NC Code
R26	Resistor 68Ω, 0.5W, 5%.	
R28,29.	Resistor 470Ω, 0.5W, 5%.	
R45.	Resistor 27KΩ, 0.25W, 5%.	
R7.	Resistor 68Ω, 0.25W, 5%.	
R8.	Resistor 390Ω, 0.5W, 5%.	
R33.	Resistor 3KΩ, 0.25W, 5%.	
R12.	Resistor 4.7Ω, 0.5W, 5%.	
R14.	Resistor 2.2Ω, 5%, RB59.	
R19.	Resistor 21.5KΩ, 0.125W, 1%.	
R3.	Resistor 17.8KΩ, 0.125W, 1%.	
R44.	Resistor 200KΩ, 0.25W, 5%.	
R46.	Resistor 3.9KΩ, 0.25W, 5%.	
R47.	Resistor 33KΩ, 0.25W, 5%.	
R49.	Resistor 150Ω, 0.5W, 5%.	
R15.	Resistor 3KΩ, 0.25W, 5%.	
R101.	Resistor 4.7Ω, 5%, RB59.	
C1-4.	Capacitor 1000µF, 6.3V.	
C7,18.	Capacitor 3.3nF, 400V, 1%.	
C8.	Capacitor 22µF, 10V, Fitco.	
C15.	Capacitor 0.1µF, 100V, 10%, MPR.	
C13,14.	Capacitor 470µF, 25V, Fitco.	
C16,23.	Capacitor 4.7µF, 16V, Fitco.	
C19.	Capacitor 47µF, 10V, Fitco.	
C20,21,24.	Capacitor 47µF, 25V, Fitco.	
C22.	Capacitor 1.2nF, 100V, 10%, ceramic.	
C25.	Capacitor 3900pF, 100V, 10%, ceramic.	
C27,29.	Capacitor 10nF, ceramic.	
C26.	Capacitor 1.5µF, 25V, CTS13.	
C10.	Capacitor 33µF, 40V, Fitco.	
C28.	Capacitor 3.3µF, 25V, Fitco.	
C30.	Capacitor 0.22µF, 100V, MPR.	
C31.	Capacitor 4.7µF, 10V, CTS13.	
C101.	Capacitor 0.1µF, 100V, 20%, PMA.	
L2.	Inductance SLF 3761.	
L3.	Shock inductance	
L5.	Inductance SLF 3781.	
L4.	Inductance SLF 3771.	
L1.	Inductance SLF 3751.	
TR1.	Transformer TR3792.	
PR1,2,3.	Potentiometer 500Ω, 2600P 501.	

SECTION IV

CONTROL PANELS

4.1 There are two Control Panels which can be fitted to the P851 CPU: a MINI Panel with two push-buttons, a three position self-locking switch and five LEDs; and a FULL Control Panel.

4.2 MINI PANEL

The Mini Panel and its logic are mounted on a 100x160mm Eurocard which fits into the left-hand slot above the mains transformer chassis and it is fastened to the Eurorack with two screws as shown in Figure 4-1. Connections to the CPU are via the back panel connector and the GP-Bus-S printed circuit board.

4.3 Switches and Lamps

There are two momentary action push-button switches, a three position self-locking key switch, and five LEDs mounted on a panel 128x16mm; this panel is fastened to the front edge of the card by two nuts and bolts and it also holds the two captive screws which hold the panel in position in the Eurorack. The functions of the switches and lamps are:

- INT -- a momentary action push-button switch which when pushed generates a control panel interrupt.
- RUN -- the LED which lights when the CPU is running a software program.
- ON -- the LED which lights when the +5VL voltage is present on the card.
- Safety Key -- a three position self-locking switch which in the LOCK RTC position inhibits the START button and enables the Real Time Clock if this option has been selected. In the 0 position the CPU is in the normal mode, and in the TEST position the micro-diagnostic program

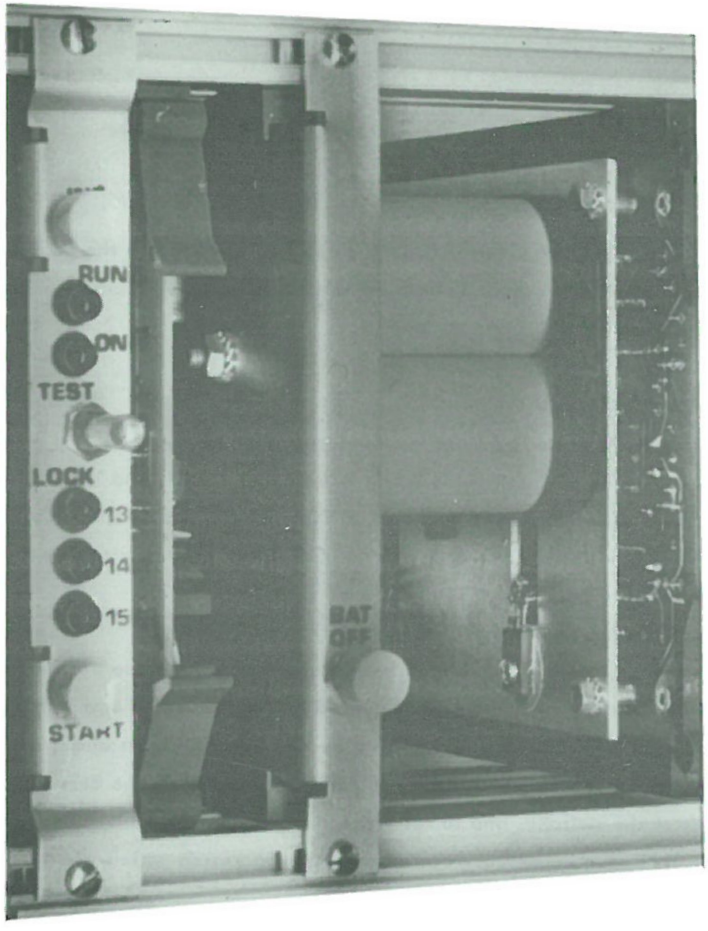


Figure 4-1 Mini Panel

starts to run until the Safety-key is returned to the 0 position.

- Test Indicators -- are the three LEDs that provide a coding address of the faulty element when the Safety-key is returned from the TEST position to the 0 position.
- START -- a momentary action push-button switch which when pushed generates the IPL signal to the CPU provided that the Safety-key is in the 0 position.

4.4 Logic Diagram

The logic diagram Figure 4-6 was derived from schematic diagram No. 5111 199 72880 dated 760519, and the following logic description is based on this diagram and EDS P851 dated 760712.

4.5 Start Logic

The start logic is initiated either by the START button or from an outside source if the Remote Load option is connected, provided the Safety-key is in the 0 position. The start signal is loaded into one of the RS flip-flops of element E2 and the resultant output signal from pin 9 is called IPL. This IPL signal is used either to activate a TTL type CPU via pin 1C09 of the card connector, or to activate one of the address lines of the PROM element A1. Provided the other address lines of the PROM do not inhibit it e.g. Safety-key in the LOCK RTC position etc., the output from the PROM (PUPN and BIO 01N to BIO 03N) is sent to the CPU to activate the IPL routine. PUPN is sent first to the CPU which replies with the signal PUPS enabling the BIO lines 01 to 03.

4.6 IPL/Program Loading

The IPL routine loads the contents of the hardware bootstrap ROM into the first 64 locations in memory. Once loaded the bootstrap starts to load the software IPL from the chosed input peripheral using data from the Data Switches D0 and D1. Each of these elements contains eight dual in line slide switches which are used to provide the Automatic Loading Data for the bootstrap; this loading data follows the standard pattern: bits 0 to 3 indicate the type of device, bits 4 to 9 contain control information for the CU, and bits 10 to 15 contain the device address. Figure 4-2 shows the data 0 and 1 positions for all sixteen switches. The switches

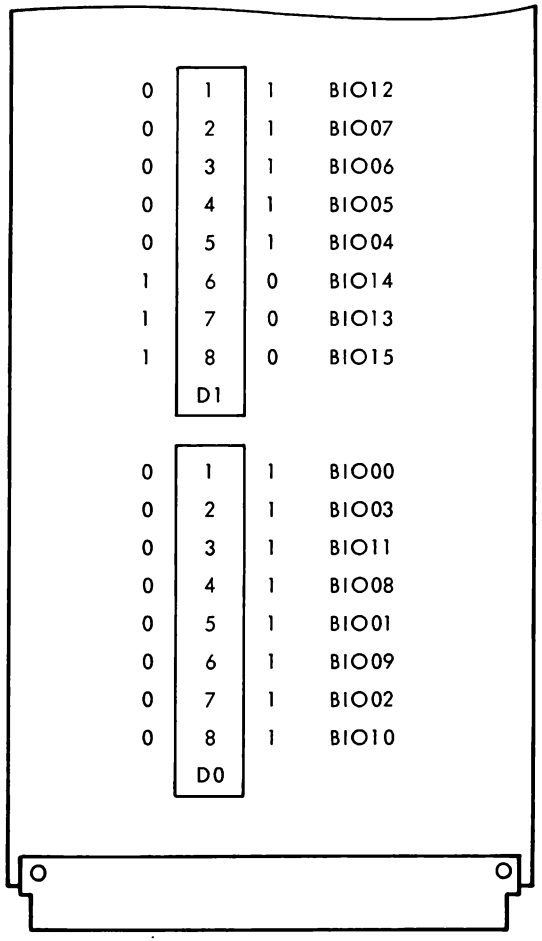


Figure 4-2 Data Switches

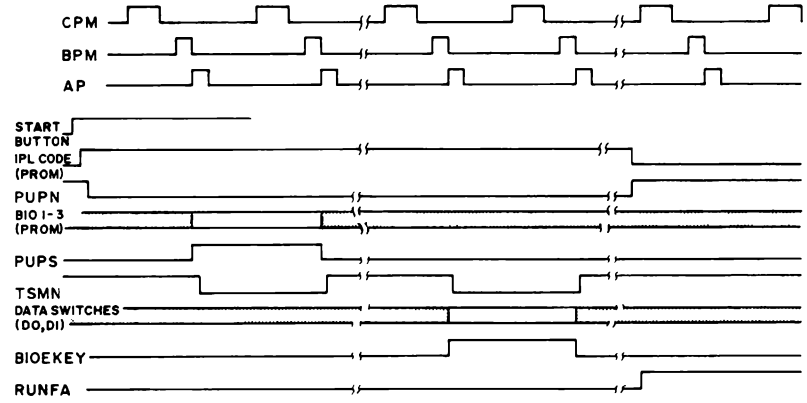


Figure 4-3 IPL Timing

will have been put into the appropriate data pattern for the selected input device at system installation time, and thereafter will not normally be changed. Data from the Data Switches is validated by the signal BIOEKEY from the CPU and it is loaded into register A15 for use by the bootstrap. Figure 4-3 shows the timing of the IPL exchange with the CPU.

4.7 Interrupt Logic

The interrupt logic memorises and controls three interrupt signals, one from the MINI Panel INT button and the other two from the power supply. One or more interrupt signals can be present at the same time because the output from the PROM has an encoded priority. After the first interrupt has been answered by PUPS, the second (or subsequent interrupt) interrupt command appears on the PROM output and PUPN remains active low. Figure 4-4 shows the timing for an interrupt.

4.8 INT. This signal is initiated by pushing the INT button on the MINI Panel. It is shaped by an RS flip-flop in element E2 and memorised by one of the

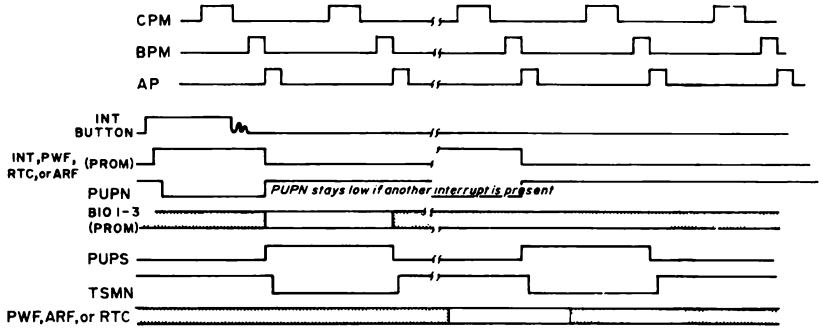


Figure 4-4 Timing for INT, PWF, ARF, or RTC

D type flip-flops in element D2. The output from D2 is stored by one of the elements of A3 whose output activates the address lines of the PROM. The PROM sends the PUPN signal to the CPU which answers with the PUPS signal. PUPS activates the BIO lines 01N to 03N and resets the flip-flops which memorised the INT signal via elements A2 and B2.

4.9 Real Time Clock (RTCN). This signal is initiated by the power supply every 20ms but will only be taken into account if the U link is in the correct position and the Safety-key is in the LOCK RTC position. The signal clocks one of the flip-flops of element D2 whose output is stored by one of the elements of A3. The output from A3 and E2 activate the address lines of the PROM which sends the PUPN signal to the CPU which answers with the PUPS signal. PUPS activates the BIO lines 01N to 03N and resets the flip-flops that memorise the RTCN signal. The RTCN is also sent to the CPU as RTCFZ1N via connector pin 1C02 where it sets the RTC flip-flop in the CPU.

4.10 Power Failure (PWFN). This signal is generated by the power supply and clocks one of the flip-flops of element D3. The output from D3 is stored by one of the flip-flops of element A3 whose output activates the address lines of the PROM which sends the PUPN signal to the CPU. The CPU replies with the PUPS signal which activates the BIO lines 01N to 03N and resets the flip-flops that memorise the power failure signal via elements A2 and B2.

4.11 Automatic Restart (ARF). This signal is initiated by the PWFN signal (at the time of a power on) provided that the Safety-key is in the LOCK RTC position and that the ARF U link is in the correct position. It clocks the other flip-flop of element D3 via invertor C2, and the output from D3 is stored by one of the flip-flops of element A3 whose output activates the address lines of the PROM after the power has been restored. When the power has been restored the PROM sends the PUPN signal to the CPU which replies with the PUPS signal to activate the BIO lines 01N to 03N and reset the flip-flops that memorise the ARF signal via elements A2 and B2.

4.12 Test

The serviceability of the CPU and its system can be tested at any time by putting the Safety-key into the TEST position. This action loads one of the RS flip-flops in element E2 whose output activates an address line of the PROM which sends the PUPN signal to the CPU. The CPU replies with the PUPS signal to activate the BIO lines 01N to 03N back to the CPU which then starts to run the micro-diagnostic routine. This routine will continue to run until the Safety-key is restored to the 0 position. To ensure that all the elements of the CPU and its system have been tested, the Safety-key should be left in the TEST position for at least 30 seconds. If a fault is detected the error code is displayed on the three LEDs DS1 to DS3 via the Bus Transceiver element E1. A description of the tests run and the error codes will be found in the Trouble-Shooting section.

4.13 PROM Contents

The dedicated addresses of the PROM and their contents are given in Table 4-1.

Table 4-1 PROM Contents

H	G	F	E	D	C	B	A	Y4	Y3	Y2	Y1	
15		2	3	04	07	06	05	09	10	11	12	
LOCKN		TESTN		PWF		RTC		PUP		BIO02		
RUNFAN			IPL		ARF		CPINT		BIO03		BIO01	
X	X	X	X	1	0	X	X	1	0	1	1	PWF
0	1	1	0	X	1	X	X	1	0	0	1	ARF
0	0	1	0	0	0	1	X	1	1	0	1	RTC
X	X	1	0	0	0	0	1	1	1	1	0	CPINT
1	1	1	1	0	X	X	X	1	0	1	0	IPL
1	X	0	X	0	0	0	1	1	1	1	1	TEST
All other addresses								0	0	0	X	

4.14 Run Mode

Once the program has been loaded, by pushing the START button, and is running, it can only be stopped by a HALT instruction (normally used at the end of the power failure routine) and the CPU switches into the Idle state. In this state, if the START button is pushed the CPU will try to load a new program. If the INT button is pushed while the program is running, the interrupt signal will be taken into account by the CPU but software will direct the operator's action; the operator cannot restart the program at its current address using the MINI panel. Anytime while the program is running the Safety-key can be switched from the 0 position to the LOCK RTC position and vice versa without affecting the program.

4.15 Power Failure/Automatic Restart

The power failure interrupt and sub-routine will be executed each time a mains failure longer than 10ms is detected and the routine ends with a HALT instruction. The Automatic Restart routine will only be carried out if the Safety-key is in the LOCK RTC position and the ARF U link is in the correct position. If the above conditions are fulfilled the automatic restart routine will restart the program using the data from the power failure routine.

4.16 Remote Loading

By using the U link CV1 it is possible to operate the CPU as part of a bigger system using either an LSI or TTL CPU as the master and the P851M as a slave, or to use one P851M as a master to control another P851M which acts as a slave. This is done via the two signals CPINT and IPL which are connected direct to the back panel connector.

4.17 U Link Data

The position of the U links is shown in Figure 4-5.

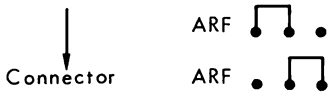
4.18 Back Panel Connector

Table 4-2 shows the pin connections of the back panel connector and their signal names. Those marked with an asterisk are connected to the CPU via a cable and two twelve pin sockets; for these signals the CPU pinning is also given. The signal names in parenthesis will be present on the connector if the CPU is connected to a TTL system, however, these signals are terminated on the MINI Panel card without being used by the CPU.

4.19 Logic and Card Details

Figure 4-6 shows the logic for the MINI Panel. Figure 4-7 shows the card and component layout and Table 4-3 lists the components.

ARF U link is located in C3 area



RTC U link located between C2 and C3 areas



Remote IPL U link located between E2 and D2 areas



Figure 4-5 U Link Data

Table 4-2 Connector Data

CPU Pin No	MINI Panel Pin Numbers					CPU Pin No
2C27	1C01	RUNFA	*	1A01	0V	2A27
2C28	02	RTCFZ1N	*	02	0V	2A28
2C29	03	PUPN	*	03	0V	2A29
2C30	04	MCLN	*	04	0V	2A30
2C31	05	PUPS	*	05	0V	2A31
2C32	06	BIOEKEY	*	06	0V	2A32
	07	RTCN		07	0V	
	08	PWFN		08	RSLN	
	09	(IPL) (IPLZ1N)		09	0V	
	10	(CPINT)		10	TSMN	
	11	(START)		11	0V	
	12	(LOADMN)		12	0V	
	13	(LOADRN)		13	0V	
	14	(READRN)		14	0V	
	15	(READMN)		15	0V	
	16	(READSTN)		16	0V	
	17	(RCP1N)		17	(RCP0N)	
	18	(RCP3N)		18	(RCP2N)	
	19	(UNLOCKN)		19	0V	
	20	(INSTN)		20	0V	
	21	(CPMCN)		21	0V	
	22	(RUNN)		22	0V	
	23	0V		23	0V	
	24	BIO08N		24	BIO00N	
	25	BIO09N		25	BIO01N	
	26	BIO10N		26	BIO02N	
	27	BIO11N		27	BIO03N	
	28	BIO12N		28	BIO04N	
	29	BIO13N		29	BIO05N	
	30	BIO14N		30	BIO06N	
	31	BIO15N		31	BIO07N	
	32	+5V		32	+5V	

* Cable links between Mini Panel and CPU Connector 2

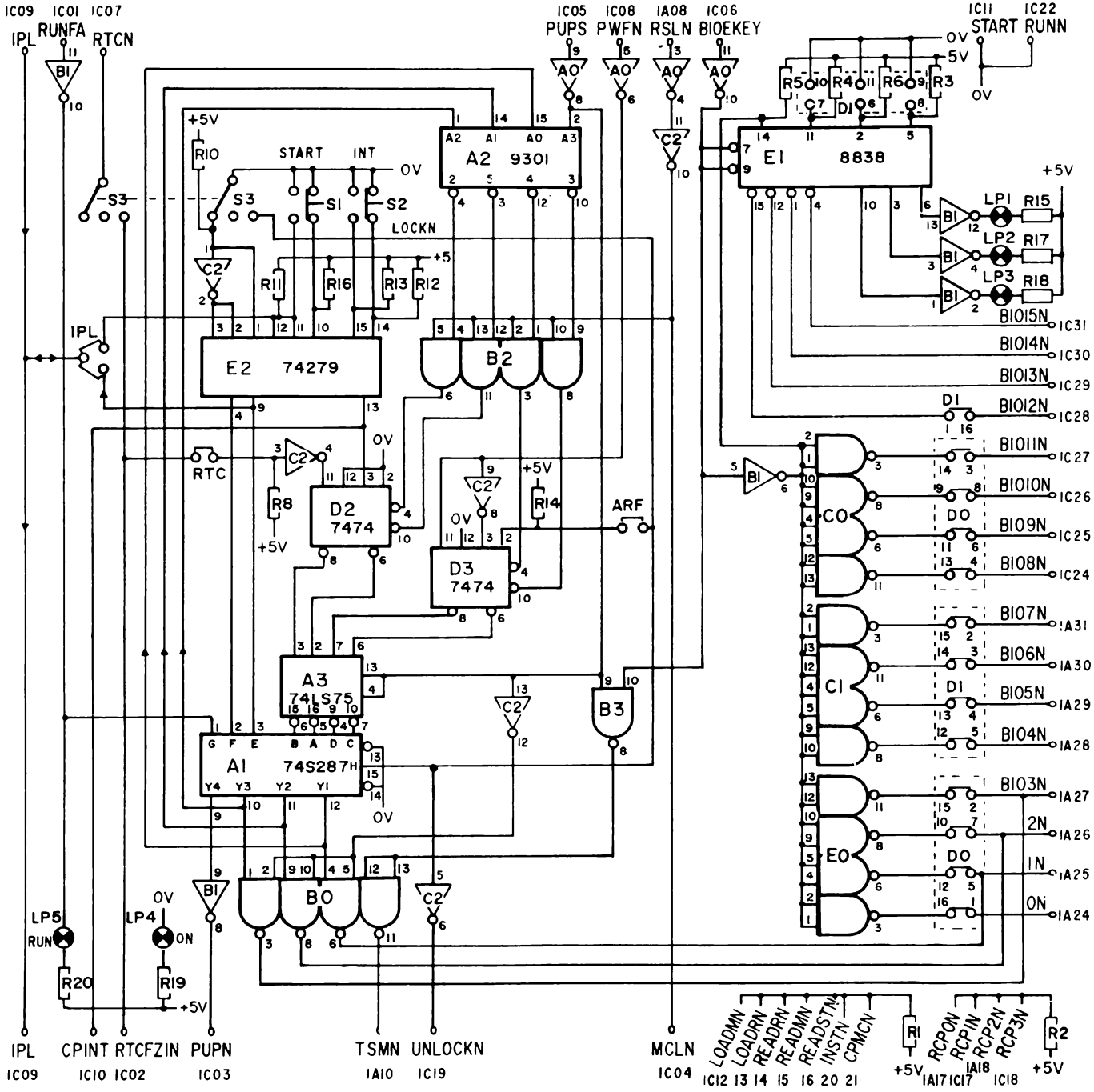


Figure 4-6 Logic Diagram

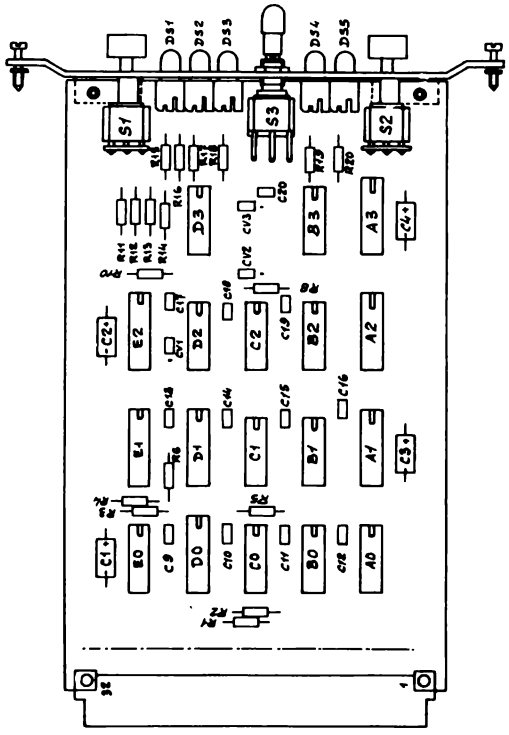


Figure 4-7 Mini Panel Card

Table 4-3 Mini Panel Components

Reference	Description	[illegible]
E1.	Integrated circuit 8838	
B0,C0,C1,E0.	Integrated circuit 7438	
E2.	Integrated circuit 74279	
B1.	Integrated circuit 7416	
A0.	Integrated circuit 74LS14	
A2.	Integrated circuit 9301	
B2.	Integrated circuit 7408	
C2.	Integrated circuit 7404	
D2,D3.	Integrated circuit 7474	
A3.	Integrated circuit 74LS75	
B3.	Integrated circuit 7400	
A1.	Integrated circuit 74S287	
D0,D1	2 x (8 switches dual in line)	
R1-6,8,10-14,16.	Resistor 1Kn, 1/4W, 5%.	
R15,17-20.	Resistor 120n, 1/4W, 5%.	
CV1,2,3.	2 Switches. 2 Push buttons. 1 lever lock switch (3 positions). 5 LED.	

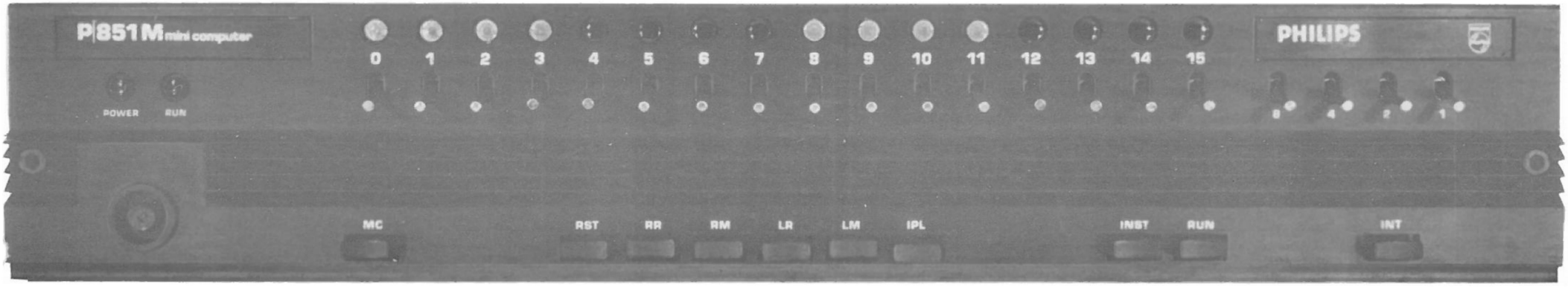


Figure 4-8 Full Panel

4.20 FULL PANEL

The Full Panel is a 2U high (88.90mm) and 19 inch wide panel which is fitted to the Eurorack and may be used for program loading and fault finding procedures. An Inter Panel card fitted with a 3 position Safety-key provides the interface between the Full Panel and the CPU and between the Power Supply and the CPU.

4.21 Control Panel Switches and Lamps

There are ten momentary action pushbutton switches that provide the command signals to the CPU, sixteen toggle switches to enable data to be sent to the CPU, four toggle switches to enable a register address to be sent to the CPU, sixteen display lamps (associated with the data switches), and a further two lamps to display the Power ON and RUN conditions of the CPU. The switches and lamps and their functions are described as follows:

- MC -- is a momentary action pushbutton switch that generates the Master Clear signal (MCLN), which is sent to the CPU and is also used by some of the logic of the Inter Panel card. This signal is also generated by the RSLN signal from the power supply during a power ON sequence.
- RST -- is a momentary action pushbutton switch that generates the Read Status signal and enables the contents of the program status word to be displayed on the DATA lamps.
- RR -- is a momentary action pushbutton switch that generates the Read Register signal and enables the contents of a register (selected by the Register Address switches) to be displayed on the DATA lamps.
- RM -- is a momentary action pushbutton switch that generates the Read

Memory signal and enables the contents of a selected memory address to be displayed on the DATA lamps. Consecutive words can be read by repeated operation of the RM button.

- LR -- is a momentary action pushbutton switch that generates the Load Register signal and enables the contents of the DATA switches to be loaded into the register selected by the Register Address switches.
- LM -- is a momentary action pushbutton switch that generates the Load Memory signal and enables the contents of the DATA switches to be loaded into a selected memory address.
- IPL -- is a momentary action pushbutton switch that generates the Initial Program Load signal and enables the contents of the Hardware Bootstrap ROM to be loaded into the first 64 locations of memory (00 to 63), which then loads the program into memory from the input device selected by the DATA switches.
- INST -- is a momentary action pushbutton switch that generates the Instruction Step signal that enables the CPU to perform one instruction of the program (as indicated by the P register) each time the button is pushed.
- RUN -- is a momentary action pushbutton switch that generates the signal that enables the CPU to start running a program.
- INT -- is a momentary action pushbutton switch that generates the control panel Interrupt signal that enables the operator to change the parameters of the program or to load additional data media while a program is being run.
- 8 4 2 1 -- are the four toggle switches that enable the operator to select a Scratch Pad register. In the downwards position they represent a binary 0 and in the upwards position they represent a binary 1.
- DATA -- are the sixteen toggle switches and their associated lamps marked 0 to 15 that enable a data word to be loaded into either a register or a memory location using the LR or LM buttons. In the downwards position they represent a binary 0 and in the upwards position they represent a binary 1 and after the LR or LM button has been pushed the data word is displayed on the lamps. When a running program stops, the lamps display the address of the next word in program (i.e. the contents of the P register).
- POWER -- This lamp is lit as soon as the +5VL has reached its working level.
- RUN -- This lamp is lit by the signal RUNFA from the CPU and indicates that a program is running.

4.22 Safety-Key

The Safety-key is a three position self locking switch mounted on the front edge of the Inter Panel card. The three positions are:

- LOCK -- In this position the Real Time Clock (RTCM) signal from the power supply and the Automatic Restart flip-flop are enabled provided that the two U links are in the correct position and all the control panel buttons are inhibited except the INT button.
- TEST -- In this position the CPU starts to run a micro-diagnostic program and will display the result on lamps 13, 14 and 15.
- Center -- In this position the CPU is in the normal RUN mode which enables all the control panel buttons but inhibits the RTCN and ARF signals.

4.23 Inter Panel Logic Description

All the functions generated by the command buttons of the control panel except the MC button are stored by flip-flops whose outputs activate the address lines of three ROMs. The encoded output command from the ROMs is sent to the CPU via BIO lines 00N to 04N. The selected register address (if required by the command) is sent to the CPU via BIO lines 05N to 08N, and when the command is executed by the CPU data from the DATA switches to the CPU or data from the CPU to the lamps are sent direct on all sixteen BIO lines.

4.24 Command Button Logic. When any of the Command buttons are pushed (except MC) the Inter Panel logic performs the same basic function for each separate button so only one button will be described here. Pushing the button on the control

panel sends a signal to the Inter Panel card that loads the D input of the appropriate flip-flop in either element A3, D3, or B2. The outputs from the flip-flops of these three elements activate the address lines of three ROMs (or PROMs) whose outputs prepare the function code which will be sent to the CPU on BIO lines 00N to 04N at the same time as the ROMs send the PUPN signal to the CPU. The CPU replies with the PUPS signal which enables the function code onto the BIO lines via gates D2 and A1, the register address (if a register function) from the control panel onto BIO lines 05N to 08N via gates E2 and F2 (gates E2 are enabled by a signal from the ROMs), and sends the TSMN signal back to the CPU. After the receipt of the TSMN signal the CPU sends the BIOEKEY signal to the Inter Panel to enable either data from the DATA switches to the CPU or data from the CPU to light the DATA lamps on the control panel. It also sends the TSMN signal back to the CPU and resets the three flip-flops that hold the signals from the power supply, and triggers the enable input pins of elements A3, D3, and B2. Figure 4-9 shows the general timing for a control panel exchange.

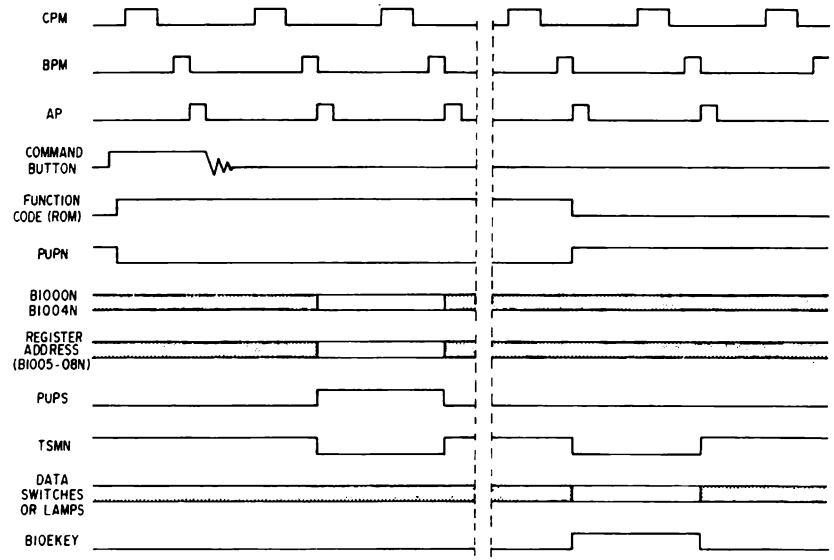


Figure 4-9 Control Panel Timing

4.25 MC Button Logic. When this button is pushed it generates the MCLN signal which is sent to the CPU via the back panel connector and is also used to reset the flip-flops on the Inter Panel card. The same signal is also generated by the RSLN signal from the power supply.

4.26 Data Logic. The data logic comprises eight Bus Transceiver elements wired back to back. The sixteen BIO lines from the CPU connect to the BUS pins of elements D0, F0, D1, and F1 via the back panel connector. The IN and OUT pins of these elements are cross connected to the IN and OUT pins of elements C0, E0, C1, and E1 whose BUS pins are connected to the Control Panel via a cable and connector. Data to or from the Control Panel is controlled by the BIOEKEY signal from the CPU. If this signal is a 0 logic level the contents of the BIO lines are displayed on the DATA lamps and if it is at logic level 1 data from the DATA keys is transferred to the CPU.

4.27 Real Time Clock (RTCN). This signal is initiated by the power supply every 20ms but will only be taken into account if the U link is in the correct position and the Safety-key is in the LOCK position. The signal clocks one of the flip-flops of element E3 whose output is stored by one of the elements of B2. The output from elements B2, D3, and A3 activate the address lines of the ROMs which send the PUPN signal to the CPU which in turn answers with the PUPS signal. This signal activates BIO lines 00N to 04N and sends the TSMN signal back to the CPU. The flip-flop is reset by an output signal from the ROMs and the BIOEKEY signal from the CPU.

4.28 Power Failure (PWFN). This signal is generated by the power supply and clocks one of the flip-flops of element F3. The output from F3 is stored by one of the elements of B2 whose output (together with the outputs from D3 and A3) activate the address lines of the ROMs which send the PUPN signal to the CPU.

The CPU replies with the PUPS signal that activates BIO lines 00N to 04N and sends the TSMN signal back to the CPU. The flip-flop is reset by the RSLN signal from the power supply.

4.29 Automatic Restart (ARF). This signal is initiated by the PWFN signal (at the time of a power failure) provided that the Safety-key is in the LOCK position and that the U link is in the correct position. It clocks the other flip-flop of element F3 whose output is stored by one of the elements of B2 whose output (together with the outputs from D3 and A3) activates the address lines of the ROMs after the power has been restored. When the power has been restored the ROMs send the PUPN signal to the CPU which replies with the PUPS signal. PUPS activates the BIO lines 00N to 04N and sends the TSMN signal back to the CPU. The flip-flop is reset by the RSLN signal from the power supply.

4.30 Test

When the CPU is in the Idle state it is possible to test the serviceability of the CPU and its system by putting the Safety-key in the TEST position. This action activates an address line of ROM C3 whose output (together with the outputs from C2 and B3) sends the PUPN signal to the CPU. The CPU replies with the PUPS signal which activates BIO lines 00N to 04N and sends the TSMN signal back to the CPU which starts to run the microdiagnostic routine. This routine will continue to run until either a fault is detected or the test is completed without error. When a fault is detected error code is displayed on the DATA lamps. A description of these error codes and the tests which can be run manually will be found in the Trouble Shooting section.

4.31 ROM Contents

The three ROMs (or PROMs) provide a priority encoding of the command, test, and interrupt signals sent to the CPU (via BIO lines 00N to 04N) as a function code. The contents of the ROMs are given in Table 4-4.

4.32 U Link Data

The position of the U links is shown in Figure 4-10.

4.33 Back Panel Connector

Table 4-5 gives the pin connections of the back panel connector and their signal names. Those marked with an asterisk are connected to the CPU via a cable and two twelve pin sockets; for these signals the CPU pinning is also given.

4.34 Control Panel Interface

Table 4-6 gives the interface connections of the cable that connects the Control Panel to the Inter Panel card.

4.35 Logic Diagrams

Figure 4-11 shows the logic diagram of the Inter Panel card and Figure 4-13 shows the logic diagram of the Control Panel.

4.36 Components and Component Layout

Figure 4-12 shows the component layout of the Inter Panel card and Table 4-7 lists the components. Figure 4-14 shows the component layout of the Control Panel and Table 4-8 lists its components.

Table 4-4 a ROM Contents

ROM 7488A or PROM 74188

ROM I (2981) address C2

	E	D	C	B	A		Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1
PINNING	14	13	12	11	10		9	7	6	5	4	3	2	1
SIGNAL	RUNFA RSTN		RTC PWF		ARF		RTCZ0 ARFZ0		INH2	PUPZ1N	BIO4	BIO3	BIO2	BIO1
0	0	0	0	0	0	RST	0	0	1	0	1	1	0	0
1	0	0	0	0	1	XX	0	0	1	1	1	1	1	1
2	0	0	0	1	0	PWF	0	1	1	0	0	0	1	1
3	0	0	0	1	1	XX	0	0	1	1	1	1	1	1
4	0	0	1	0	0	RTC	1	0	1	0	0	1	0	1
5	0	0	1	0	1	XX	0	0	1	1	1	1	1	1
6	0	0	1	1	0	PWF	0	1	1	0	0	0	1	1
7	0	0	1	1	1	XX	0	0	1	1	1	1	1	1
8	0	1	0	0	0	IDLE	0	0	0	1	1	1	1	1
9	0	1	0	0	1	ARF	0	1	1	0	0	0	0	1
10	0	1	0	1	0	PWF	0	1	1	0	0	0	1	1
11	0	1	0	1	1	XX	0	0	1	1	1	1	1	1
12	0	1	1	0	0	RTC	1	0	1	0	0	1	0	1
13	0	1	1	0	1	ARF	0	1	1	0	0	0	0	1
14	0	1	1	1	0	PWF	0	1	1	0	0	0	1	1
15	0	1	1	1	1	XX	0	0	1	1	1	1	1	1
16	1	0	0	0	0	RST	0	0	1	0	1	1	0	0
17	1	0	0	0	1	XX	0	0	1	1	1	1	1	1
18	1	0	0	1	0	PWF	0	1	1	0	0	0	1	1
19	1	0	0	1	1	XX	0	0	1	1	1	1	1	1
20	1	0	1	0	0	RTC	1	0	1	0	0	1	0	1
21	1	0	1	0	1	XX	0	0	1	1	1	1	1	1
22	1	0	1	1	0	PWF	0	1	1	0	0	0	1	1
23	1	0	1	1	1	XX	0	0	1	1	1	1	1	1
24	1	1	0	0	0	IDLE	0	0	0	1	1	1	1	1
25	1	1	0	0	1	XX	0	0	1	1	1	1	1	1
26	1	1	0	1	0	PWF	0	1	1	0	0	0	1	1
27	1	1	0	1	1	XX	0	0	1	1	1	1	1	1
28	1	1	1	0	0	RTC	1	0	1	0	0	1	0	1
29	1	1	1	0	1	XX	0	0	1	1	1	1	1	1
30	1	1	1	1	0	PWF	0	1	1	0	0	0	1	1
31	1	1	1	1	1	XX	0	0	1	1	1	1	1	1

Table 4-4 b ROM Contents

ROM 7488A or PROM 74188

ROM II (2991) address C3

	E	D	C	B	A		Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1
PINNING	14	13	12	11	10		9	7	6	5	4	3	2	1
SIGNAL	INSTN CPINT		RMN RRN		TESTN		INH3		RRCPN PUPZ1N		BIO4	BIO3	BIO2	BIO1
0	0	0	0	0	0	XX	0	1	1	1	1	1	1	1
1	0	0	0	0	1	XX	0	1	1	1	1	1	1	1
2	0	0	0	1	0	XX	0	1	1	1	1	1	1	1
3	0	0	0	1	1	XX	0	1	1	1	1	1	1	1
4	0	0	1	0	0	XX	0	1	1	1	1	1	1	1
5	0	0	1	0	1	XX	0	1	1	1	1	1	1	1
6	0	0	1	1	0	XX	0	1	1	1	1	1	1	1
7	0	0	1	1	1	INST	0	1	1	0	1	0	0	1
8	0	1	0	0	0	XX	0	1	1	1	1	1	1	1
9	0	1	0	0	1	XX	0	1	1	1	1	1	1	1
10	0	1	0	1	0	XX	0	1	1	1	1	1	1	1
11	0	1	0	1	1	XX	0	1	1	1	1	1	1	1
12	0	1	1	0	0	XX	0	1	1	1	1	1	1	1
13	0	1	1	0	1	XX	0	1	1	1	1	1	1	1
14	0	1	1	1	0	XX	0	1	1	1	1	1	1	1
15	0	1	1	1	1	XX	0	1	1	1	1	1	1	1
16	1	0	0	0	0	XX	0	1	1	1	1	1	1	1
17	1	0	0	0	1	XX	0	1	1	1	1	1	1	1
18	1	0	0	1	0	XX	0	1	1	1	1	1	1	1
19	1	0	0	1	1	RM	0	1	1	0	1	1	1	0
20	1	0	1	0	0	XX	0	1	1	1	1	1	1	1
21	1	0	1	0	1	RR	0	1	0	0	0	1	0	0
22	1	0	1	1	0	TEST	0	1	1	0	0	1	1	1
23	1	0	1	1	1	IDLE	0	0	1	1	1	1	1	1
24	1	1	0	0	0	XX	0	1	1	1	1	1	1	1
25	1	1	0	0	1	XX	0	1	1	1	1	1	1	1
26	1	1	0	1	0	XX	0	1	1	1	1	1	1	1
27	1	1	0	1	1	XX	0	1	1	1	1	1	1	1
28	1	1	1	0	0	XX	0	1	1	1	1	1	1	1
29	1	1	1	0	1	XX	0	1	1	1	1	1	1	1
30	1	1	1	1	0	XX	0	1	1	1	1	1	1	1
31	1	1	1	1	1	CPINT	1	1	1	0	0	1	1	0

Table 4-4 c ROM Contents

ROM 7488A or PROM 74188
ROM III (2971) address B3

	E	D	C	B	A		Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1
PINNING	14	13	12	11	10		9	7	6	5	4	3	2	1
SIGNAL	RUN	IPL	LMN	LRN	RUNFA		RTCZ0	RTCZ0	RRCPN	PUPZ1N	BIO4	BIO3	BIO2	BIO1
0	0	0	0	0	0	XX	0	0	1	1	0	0	0	0
1	0	0	0	0	1	XX	0	0	1	1	0	0	0	0
2	0	0	0	1	0	XX	0	0	1	1	0	0	0	0
3	0	0	0	1	1	XX	0	0	1	1	0	0	0	0
4	0	0	1	0	0	XX	0	0	1	1	0	0	0	0
5	0	0	1	0	1	XX	0	0	1	1	0	0	0	0
6	0	0	1	1	0	XX	0	0	1	1	0	0	0	0
7	0	0	1	1	1	XX	0	0	1	1	0	0	0	0
8	0	1	0	0	0	XX	0	0	1	1	0	0	0	0
9	0	1	0	0	1	XX	0	0	1	1	0	0	0	0
10	0	1	0	1	0	XX	0	0	1	1	0	0	0	0
11	0	1	0	1	1	XX	0	0	1	1	0	0	0	0
12	0	1	1	0	0	XX	0	0	1	1	0	0	0	0
13	0	1	1	0	1	XX	0	0	1	1	0	0	0	0
14	0	1	1	1	0	IPL	0	0	1	0	0	0	1	0
15	0	1	1	1	1	XX	0	0	1	1	0	0	0	0
16	1	0	0	0	0	LR	0	0	0	0	1	0	0	0
17	1	0	0	0	1	LR	0	0	0	0	1	0	0	0
18	1	0	0	1	0	LM	0	0	1	0	1	0	1	0
19	1	0	0	1	1	LM	0	0	1	0	1	0	1	0
20	1	0	1	0	0	LR	0	0	0	0	1	0	0	0
21	1	0	1	0	1	LR	0	0	0	0	1	0	0	0
22	1	0	1	1	0	RUN	0	0	1	0	1	1	0	1
23	1	0	1	1	1	RUN	0	0	1	0	1	1	0	1
24	1	1	0	0	0	LR	0	0	0	0	1	0	0	0
25	1	1	0	0	1	LR	0	0	0	0	1	0	0	0
26	1	1	0	1	0	LM	0	0	1	0	1	0	1	0
27	1	1	0	1	1	LM	0	0	1	0	1	0	1	0
28	1	1	1	0	0	LR	0	0	0	0	1	0	0	0
29	1	1	1	0	1	LR	0	0	0	0	1	0	0	0
30	1	1	1	1	0	XX	0	0	1	1	0	0	0	0
31	1	1	1	1	1	XX	0	0	1	1	0	0	0	0

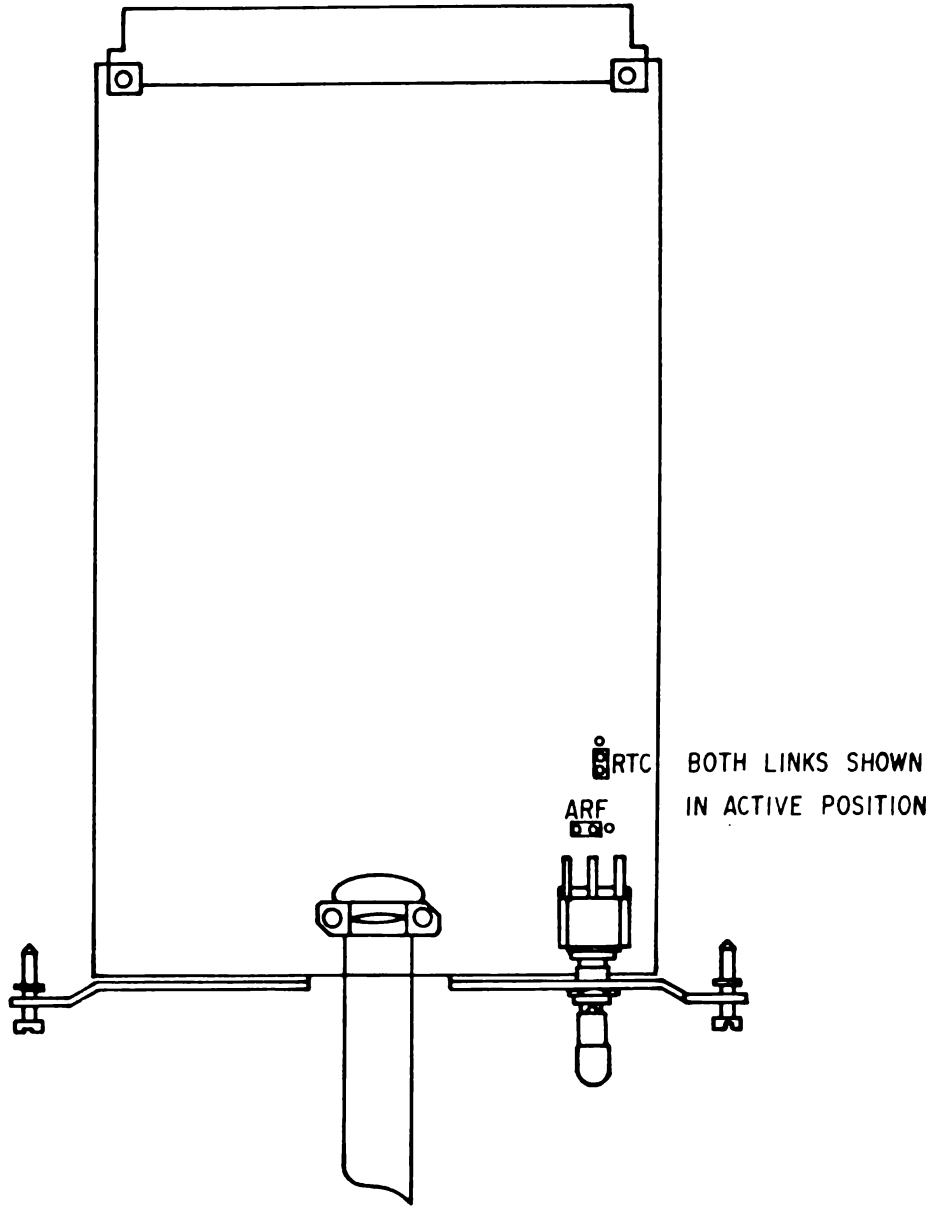


Figure 4-10 U Link Data

Table 4-5 Back Panel Connector Data

CPU Pin No	Inter Panel Pin Numbers					CPU Pin No
2C27	1C01	RUNFA	*	1A01	0V	2A27
2C28	1C02	RTCFZ1N	*	1A02	0V	2A28
2C29	1C03	PUPN	*	1A03	0V	2A29
2C30	1C04	MCLN	*	1A04	0V	2A30
2C31	1C05	PUPS	*	1A05	0V	2A31
2C32	1C06	BIOEKEY	*	1A06	0V	2A32
	1C07	RTCN		1A07	0V	
	1C08	PWFN		1A08	RSLN	
	1C09			1A09	0V	
	1C10			1A10	TSMN	
	1C11			1A11	0V	
	1C12			1A12	0V	
	1C13			1A13	0V	
	1C14			1A14	0V	
	1C15			1A15	0V	
	1C16			1A16	0V	
	1C17			1A17		
	1C18			1A18		
	1C19			1A19	0V	
	1C20			1A20	0V	
	1C21			1A21	0V	
	1C22			1A22	0V	
	1C23	0V		1A23	0V	
	1C24	BIO08N		1A24	BIO00N	
	1C25	BIO09N		1A25	BIO01N	
	1C26	BIO10N		1A26	BIO02N	
	1C27	BIO11N		1A27	BIO03N	
	1C28	BIO12N		1A28	BIO04N	
	1C29	BIO13N		1A29	BIO05N	
	1C30	BIO14N		1A30	BIO06N	
	1C31	BIO15N		1A31	BIO07N	
	1C32	+5V		1A32	+5V	

* Cable link between Inter Panel Card and CPU Connector 2

Table 4-6 Control Panel Interface Data

Signal	Address on the Card (solder connections)	Address on the Control Panel Connector
BIO15N	4	A 1
BIO14N	7	A 2
BIO13N	9	A 3
BIO12N	8	A 4
BIO11N	22	A 5
BIO10N	16	A 6
BIO09N	25	A 7
BIO08N	19	A 8
BIO07N	2	A 9
BIO06N	3	A10
BIO05N	12	A11
BIO04N	13	A12
BIO03N	23	A13
BIO02N	17	A14
BIO01N	24	A15
BIO00N	18	A16
RRN	14	A18
0V	34.35	A19
CPMCN	26	B 1
BIOEKEY	32	B 2
IPL	28	B 3
RUNN	29	B 5
CPINT	20	B 7
RCPON	11	B 8
RUNFA	33	B 9
LRN	30	B10
RCP1N	10	B11
RSTN	27	B12
RCP2N	6	B13
RCP3N	5	B14
RMN	21	B15
INSTN	15	B16
LMN	31	B17
LOCKN	1	B18
+5V	36.37	B19

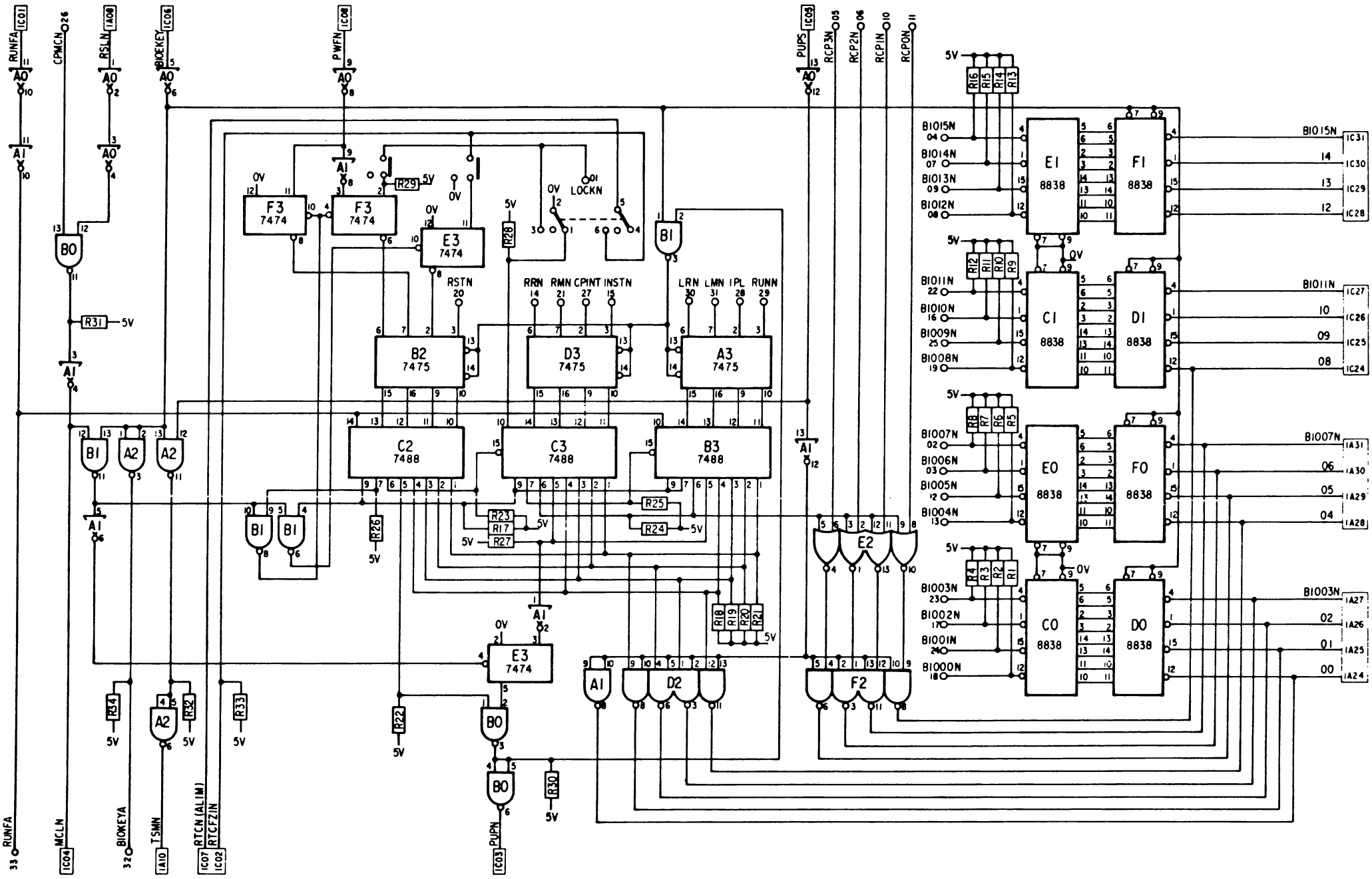


Figure 4-11 Logic Diagram of Inter Panel

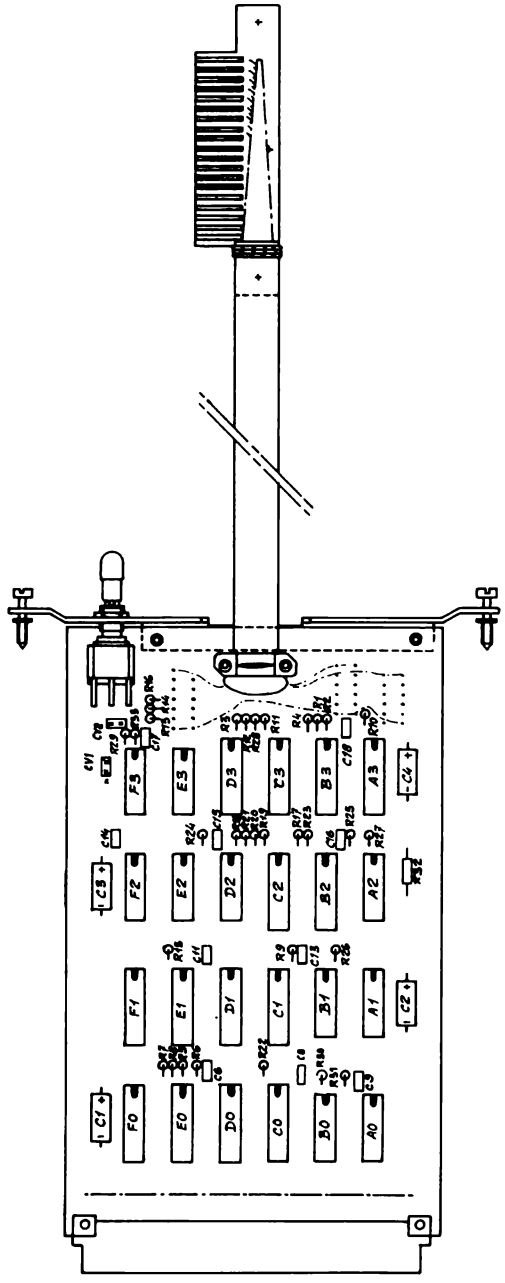


Figure 4-12 Layout of Inter Panel Card

Table 4-7 Component List of Inter Panel

Reference	Description	12NC Code
C0,C1,D0,D1,E0,E1,F0,F1.	Integrated circuit 8838	
A2,D2,F2.	Integrated circuit 7438	
E3,F3.	Integrated circuit 74LS74	
A3, B2, D3.	Integrated circuit 74LS75	
A0.	Integrated circuit 7414	
E2.	Integrated circuit 74LS02	
B1.	Integrated circuit 74LS00	
B3,C2,C3.	Integrated circuit 7488	
B0.	Integrated circuit 74LS26	
A1.	Integrated circuit 74LS04	
R1-16.	Resistor 1.2Kn, 1/4W, 5%.	
R17-34.	Resistor 1Kn, 1/4W, 5%.	
CV1,CV2.	2 Switches. Self locking switch with 3 positions. 1 Cable of 37 wires with 2 x 19 point connector.	

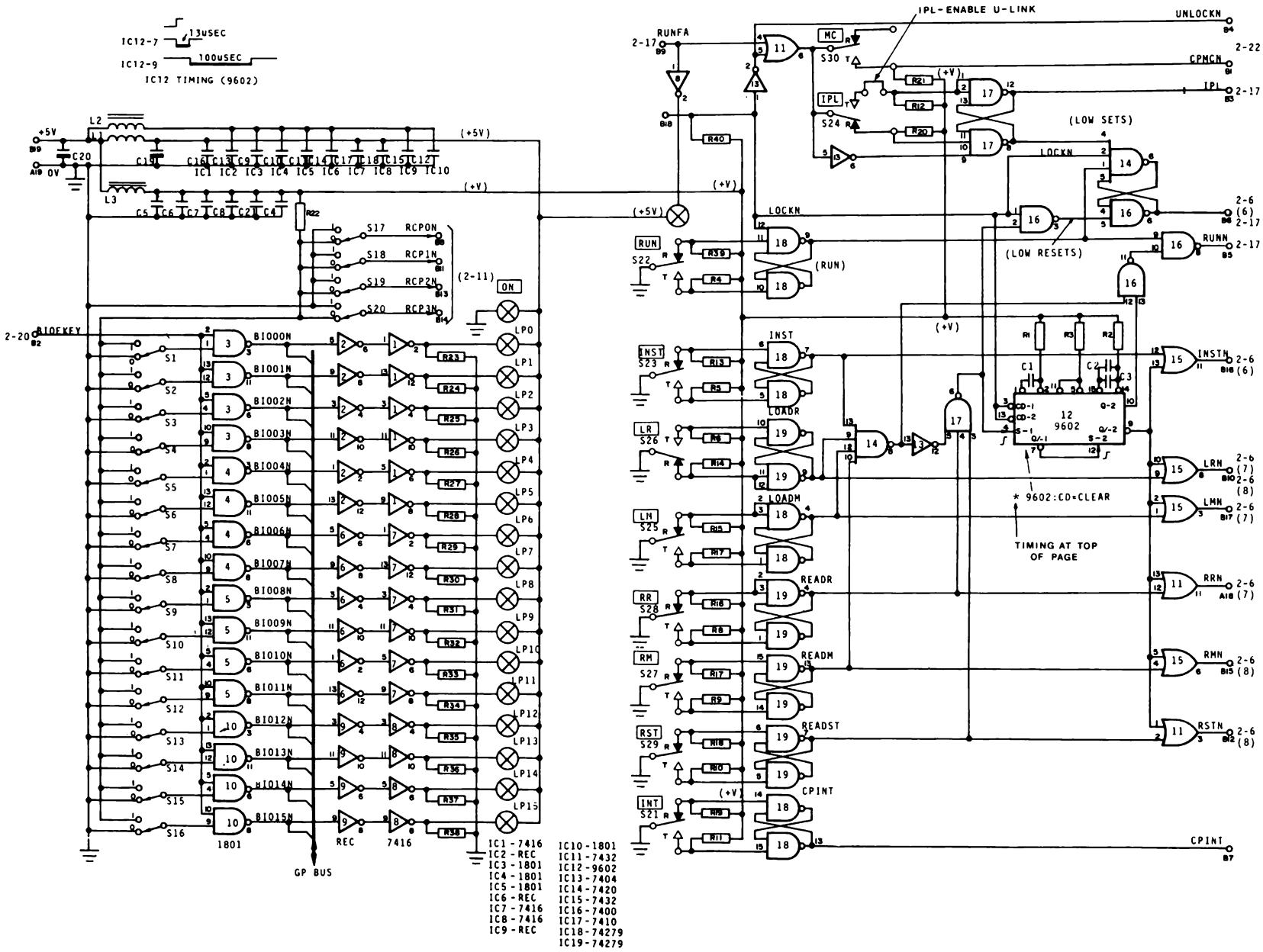
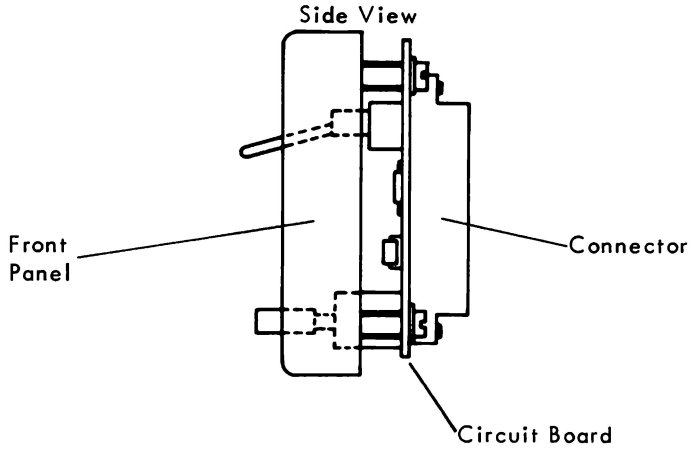


Figure 4-13 Logic Diagram of Control Panel

Table 4-8 Control Panel Parts List



Reference	Description	12NC Code
	Control Panel	5111 199 81450
	Front Cover	79700
	Printed Circuit, equiped	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, 47µF, 20V,	CTS13
C1	Capacitor, 4700pF, 100V, ±10	cer plat
C2 thru C18, C21	Capacitor, 3900pF, 100V, ±10	cer plat
L1, L2, L3	Inductor	
R1	Resistor, 10Kn, 0.250W, ±5	
R2	Resistor, 46.4Kn, 0.125W, ±1	
R23 thru R38	Resistor, 330n 0.250W, ±5	
R3 thru R22, R39, R40	Resistor, 1Kn, 0.250W, ±5	
	Toggle Switch 7101 LYCG	
	Push Button Switch 2RT - TFB	
	Lamp 6V 30mA ref. 2306	
	Jumper Block F088	

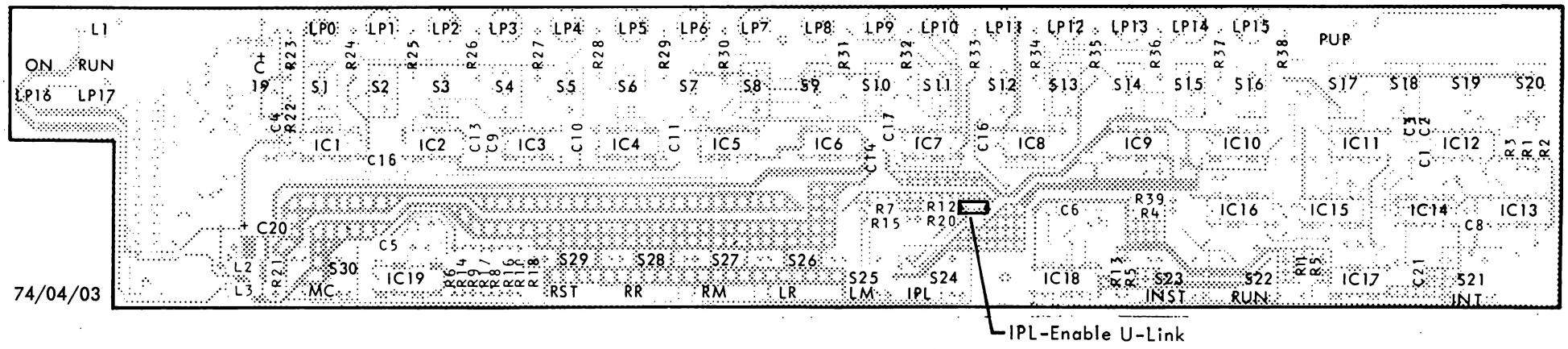


Figure 4-14 Control Panel Layout

SECTION V

CPU

5.1 INTRODUCTION

The Central Processor Unit (CPU) is contained on a 6 layer, 160 x 233.4mm Eurocard. The card is designed to plug directly into the General Purpose Bus S and to execute the P851M instruction set. When considering the operation of the CPU it must be remembered that the Bus is used for input, output and internal transfers. For input or output transfers the Bus is always referred to as the I/O Bus. CPU controls all data transfers using the Master/Slave principle and also decides to which Master the Bus is to be allocated. CPU is controlled by two types of instructions: Software instructions and Microinstructions. Software instructions are Fetched from an external Memory and decoded by the Planet LSI (Large Scale Integrated circuit) device. Microinstructions are contained in ten ROMs (Read Only Memories). Normally an instruction from Memory is decoded to form the start address of a microinstruction routine. The microinstruction routine then assumes control of CPU functions until another Software instruction is addressed. There are 16 interrupts, including 4 internal, which operate over 63 Memory levels. All data transfers with CPU are normally via the interrupt routine under microinstruction control. For arithmetic operations microinstruction data is output from the ROMs to Spalu which specifies the type of arithmetic function. Principle hardware features available with CPU are:

- 16 Scratch Pad registers, 14 of which are accessed by software.
- Power Failure/Automatic Restart
- Real Time Clock interrupt every 20mS
- 16 interrupt levels including 4 internal
- Bus Control
- Interface to either Full or Mini Control Panels

- Initial Program Loading
- Multiply, divide, double shift, add, subtract
- Timeout
- CPU cycle time of 640nS

5.2 Terms Used in the Text

There are three transfer signals: Transfer to Control Unit TMPN, Transfer to External Register TMEN, Transfer to Memory TMRN. To prevent repetition of these signal names TMSN is used, meaning Transfer Master to Slave. When an external device requests the Bus it could be either IOP Input Output Processor or the DMA CU Direct Memory Access Control Unit. Only the term IOP is used to mean either IOP or DMACU. Signal names that appear in the text are the same as those used in the Logic Drawings. Letter N following a signal name means that the signal is active low logic level 0. If no N follows the signal name then it is active high logic level 1. CPU is controlled by two types of instructions.

- Microinstructions which are contained in Read Only Memories housed on the CPU card. These will be referred to as microinstructions.
- Macroinstructions which are part of the program stored in Memory. These will be referred to as Software instructions.

Some signal names and mnemonics are listed at the end of this section and their meanings are given. When the text refers to a paricular microinstruction routine and a more explicit description is required then refer to the back of this section for the routines. When cross references are necessary the paragraph heading is used so refer to the index for the paragraph number.

5.3 Interface Signals (Figure 5-1)

All CPU Interface Signals are described here except for those associated with the GP Bus which are explained in the Interface Section.

- MCLN -- Signal from the Power Supply at switch-on or from the Control Panel Master Clear button. CPU uses it to enable GP Bus signal CLEARN.
- BIOEKEY-- When BIOEKEY is 0 the contents of the Bus are displayed at the Control Panel. When 1 the Control Panel Data Keys transfer a data pattern to the CPU via the Bus Lines.
- PUPS -- In response to PUPN from the Control Panel CPU returns PUPS to the Control Panel to enable a function code to be sent back to CPU.
- RUNFA -- When a program is running signal RUNFA is used to illuminate the RUN lamp at the Control Panel.
- RTCFZ1N -- Sent from the Power Supply, this signal is routed via the Control Panel to be the Real Time Clock input giving an input every 20mS.
- CPZ0N -- Decoded from an RIT instruction, this signal is sent to SCUZ to reset the Break Detect F/F.
- CPFN- Output from the Control Panel Interrupt Flip-Flop and linked to one of the 16 interrupt lines.
- PFFN -- Output from the Power Fail Flip-Flop and linked to one of the 16 interrupt lines.
- PIFN -- Output from the Program Interrupt Flip-Flop and linked to one of the 16 interrupt lines.
- RTCAN -- Output from the Real Time Clock Flip-Flop and linked to one of the 16 interrupt lines.

5.4 Major Units of the CPU (Figure 5-1)

The following Logic Units are housed on the CPU card. For the Spalu and Planet (Large Scale Integrated Circuits) the registers contained in the devices are also described.

- Bootstrap ROM -- Initial Program Loading facility enabling the Bootstrap to be loaded into Memory. Initiation is from the Control Panel.
- RAR -- Rom Address Register addresses the 10 Control Roms via 10 address lines.
- CPU Clock -- CPU Clock supplies the clock pulses which control CPU operations. There are 7 clock pulses used by CPU, AP, BPM, BPN, CPM, CPN, CP, QA.
- BMTT (Bidirectional MOS/TTL Translators) -- The BMTT devices connect the CPU Internal Bus (+10V MOS) to the Bidirectional Input/Output Lines (+5V TTL).

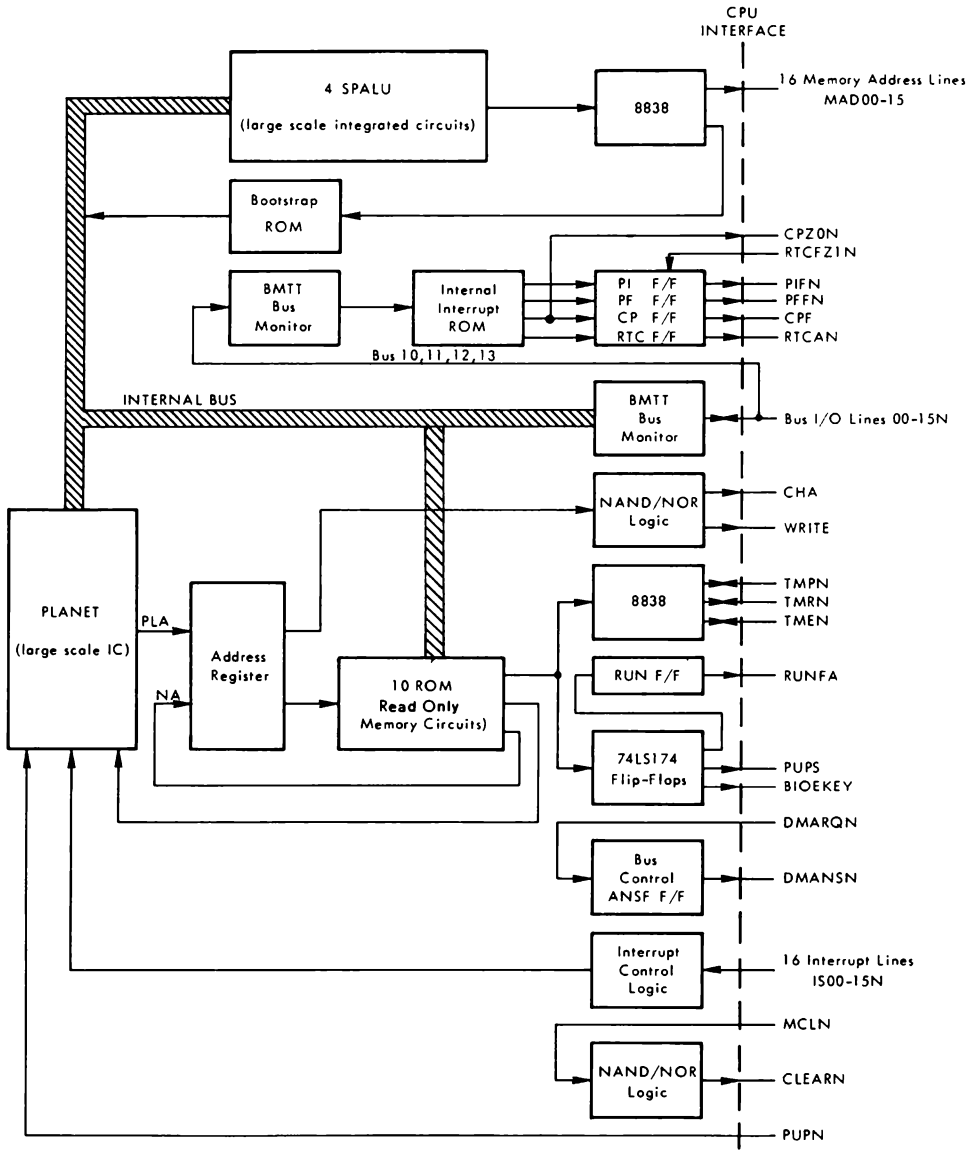


Figure 5-1 Interface Signals and Major Units

- ANSF F/F -- This F/F controls the time at which the Bus may be transferred between CPU and another Master and also data on the BIO and MAD lines.
- Interrupt Control Logic -- Four internal interrupts are latched at the CPU and in conjunction with 12 external interrupts form the interrupt lines. These interrupts are normally tested every CPU cycle except during the execution of certain microinstruction routines when they are inhibited.
- Spalu -- Scratch Pad and Arithmetic Logic Unit is a 4-bit integrated circuit. CPU has 4 Spalu circuits to make up the 16-bit word. There are 16 general registers stacked in a RAM and 4 working registers. The Scratch Pad registers are latched inside the device but registers AX and RX are controlled externally.

  Scratch Pad is the term given to registers A1-A15. Access is either from Control Panel or software, but overall control is by microinstruction A1-A14 are addressed from instruction format during CPU operation. The CPU holds one or both operands of an instruction or a result. For Memory transfers they can be used as addressing and indexing registers. A15 can be addressed from instruction format although it is principally used as the stackpointer by the interrupt system.

  ALU is the term given to the Arithmetic Logic Unit. This enables the subtraction, addition, or combination of either P register Scratch Pad and A register or Scratch Pad, A register and AX register.
- Planet -- Planet decodes the Bus lines to set up the Rom Address Register which in turn addresses the Read Only Memories. CPU also uses Planet to contain the Program Status and other CPU operating conditions. The principal registers are:

  Register K. Containing 16-bits it holds either the complete instruction or the most significant word of a double length instruction. The Program Status Word is a combination of 4 registers PLR, CR, FU, ENB.

  PLR. The Program Level Register is 6-bits long and is used to hold the priority level of the program being executed.

  CR. The Condition Register is 2-bits long and is used to hold the response or state of a result after the execution of certain instructions.

  ENB. The Enable flip-flop is one bit and is used to either put the CPU

in the Enable or Inhibit Interrupt State.

FU. The User flip-flop is one bit and is used to either put CPU in the System or User Mode.

Other Planet registers are:

IR. The Interrupt flip-flop is one bit and sets after a new interrupt level has been accepted by CPU.

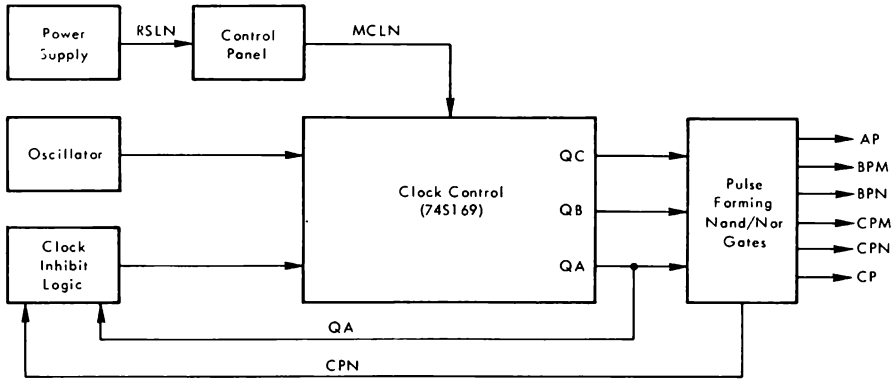
MODE. The Mode flip-flops are 2-bits long and are used to decide how the Bus lines will be decoded.

RUN. The Run flip-flop is one bit and is set when CPU is executing the program.

5.5 CPU CLOCK (Figure 5-23c)

The CPU Clock is a synchronous counter which generates all of the CPU timing pulses. The following block diagram shows the clock with associated logic and timing signals. Input to the clock is from an oscillator which supplies the basic 80nS input frequency. Timing pulses and their function at the CPU are as follows:

- CP -- clocks the ROM Address Register to set up the next ROM Address.
- CPN -- clocks the ROM data field, tests 4 internal interrupts, enables interface signals TMRN, TMPN, TMEN, and inhibits the clock for a CPU transfer.
- CPM -- clocks the operation execution in Spalu, and clocks the RS Field into Planet.
- BPN -- tests the 16 interrupt lines.
- BPM -- clocks the microinstruction into Spalu.
- AP -- clocks the ROM micro-control signals.



A timing diagram showing the relationship of these timing pulses is shown in Figure 5-23c.

5.6 Clock Starts at Switch-On

At switch-on the Master Clear signal MCLN from the Control Panel is low and the Clock is inhibited. When MCLN goes high the Clock starts on the next rising edge of the oscillator input signal.

5.7 BUS CONTROL (Figure 5-23c)

The P851 system employs the General Purpose Bus S. This Bus can be used by the CPU to perform three types of transfer: input transfers, output transfers, and internal CPU transfers. When Input or Output transfers are specified then the Bus is referred to as the I/O Bus. For Internal transfers it is referred to as the Internal Bus. Note that Input and Output transfers cannot take place at the same time, and during I/O transfers the clock is stopped and the Internal Bus is in a waiting state. At the beginning of all I/O transfers the following circuits are initiated:

- The ANSF F/F is set or reset depending on the type of I/O transfer.
- Timeout Logic is initiated. This is a safety feature in the event of an end of transfer signal not being received by CPU. In this case Timeout generates its own TSMN signal (TOUT).

All Internal Bus transfers are under microinstruction control to transfer data between CPU registers. Internal transfers do not use the ANSF F/F or Timeout Control Logic and the CPU Clock is not stopped. All data transfers use the Master/Slave principle. CPU is always Master and cannot be a Slave. Master/Slave Principle is described in the Interface Section under the paragraph heading Data Transfers.

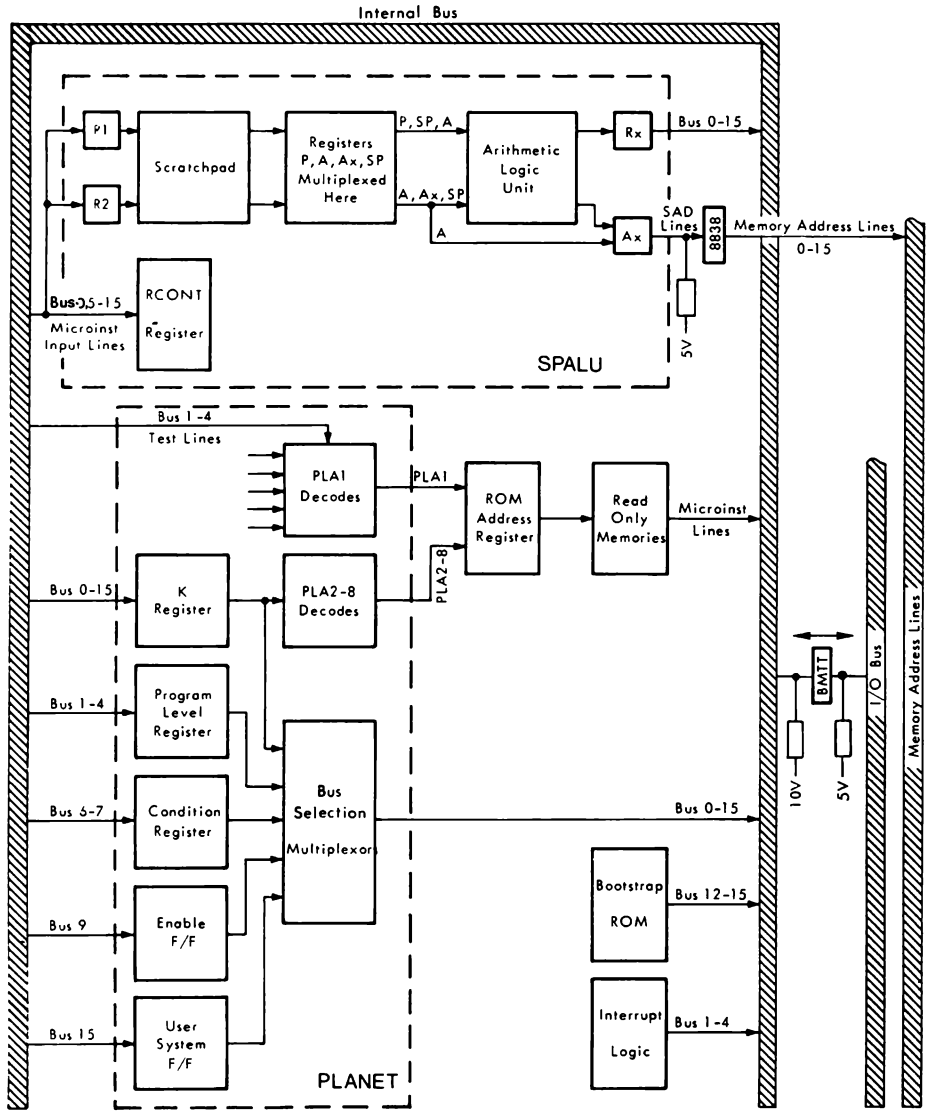


Figure 5-2 Relationship of Internal and I/O Bus

5.8 Relationship of the Internal and I/O Bus (Figure 5-2)

The Internal Bus interconnects CPU registers for internal transfers and operates at a 10-volt level. The I/O Bus interconnects the CPU and other devices for I/O transfers and operates at a 5-volt level. When data is on the Bus it could be for either internal or I/O transfer. For an I/O transfer the BMTT logic, linking the internal and I/O Bus, must be enabled in one of two ways:

- For IOP transfers the ANSF F/F Control Logic sends signal ANSB which disables the BMTT.
- For CPU transfers the BMTT Logic is enabled on ROM Address Line RADA5 which also gives the direction of transfer.

For Internal Bus transfers the BMTT logic is disabled and the internal transfer is enabled by the appropriate microinstruction control signal.

5.9 Internal Bus Transfers

The Internal Bus is shown with the registers including Planet and Spalu which are associated with the Bus. There are different ways of transferring data on the Internal Bus:

- Microinstruction data from the Control ROMs to Spalu uses Bus line 0, and 5-15 under control of signal DATA.
- Microinstruction Test Bits from the Control ROMs to Planet uses Bus lines 1-4.
- A new interrupt level from the Interrupt Logic to be compared with the current interrupt level in Planet uses Bus 1-4 under control of signal INTVAL.
- Microinstruction data from the Bootstrap ROM to Spalu uses Bus lines 12-15 under control of signal IPSN.

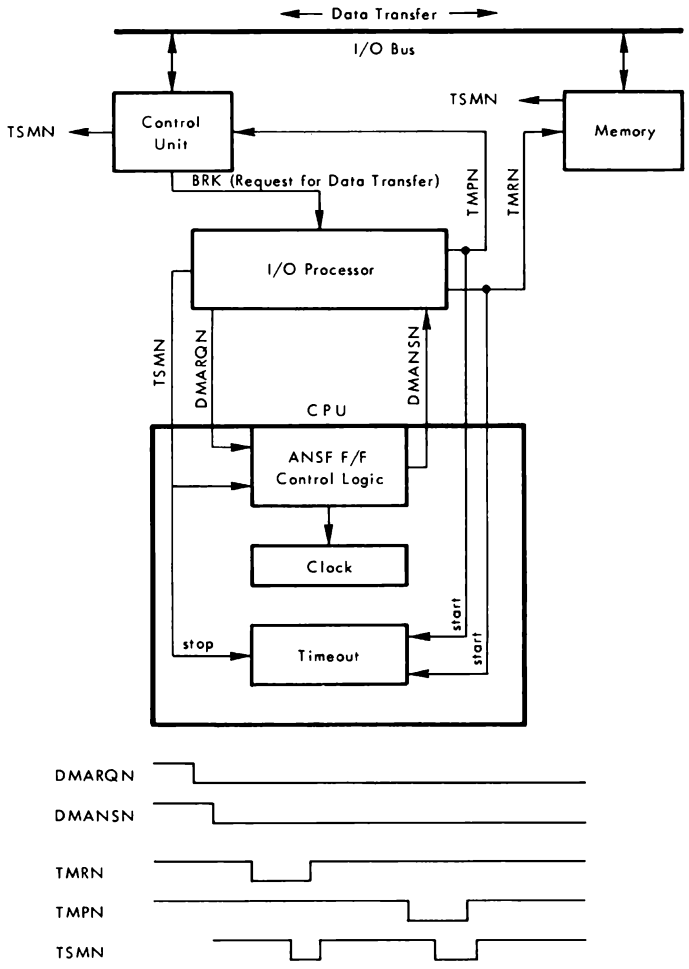
Note that these enabling signals DATA, INTVAL, IPSN are all microinstruction signals initiated at time AP.

5.10 I/O Bus Transfers

The P851 system uses 2 channels for data transfer, Program Channel and IOP Channel. These transfers cannot be carried out simultaneously; IOP must request the Bus from CPU before it will allow a data transfer to take place. Both of these data transfers have their own timing signals.

5.11 Allocating the Bus for IOP Transfers

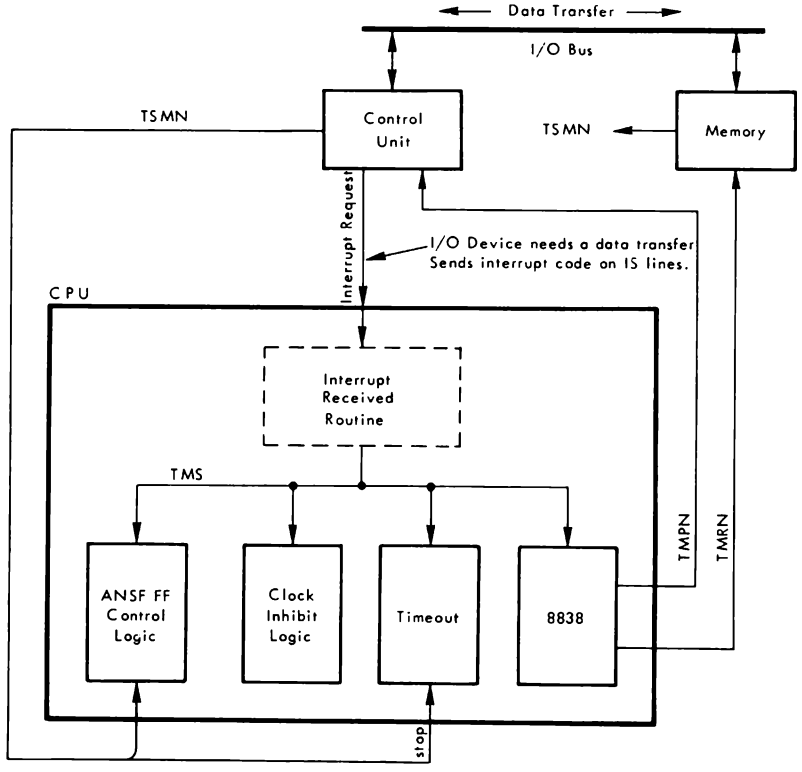
The IOP uses timing signals DMARQN and DMANSN. This type of request may be used for taking the Bus and transferring blocks of data between a Control Unit and Memory.



CU requests a data transfer by sending signal BRK to IOP → IOP requests permission to use Bus by sending signal DMARQN. If Bus is not busy DMARQN sets ANSF F/F in CPU → ANSF F/F sends DMANSN to IOP giving permission to use the Bus → Bus is now allocated so ANSF F/F is set to prevent other CPU I/Os. ANSF F/F stops the Clock, IOP generates TMRN, TMPN to enable the data transfers and also to initiate Timeout → Data transfer takes place on I/O Bus → Transfer is complete so IOP sets DMARQN high which resets ANSF F/F, starts CPU Clock and resets Timeout.

5.12 Allocating the Bus for Program Channel Transfers

For Program Channel TMSN initiates and TSMN ends the transfer. For the relationship between the data transfer signals see Interface Section, Exchange Timing.



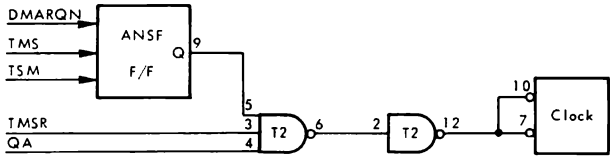
An interrupt is received from the Control Unit and the Interrupt Received Routine is executed → Microinstruction signal TMS initiates Timeout, stops the clock and inhibits the ANSF F/F so that any other transfer requests are ignored → Data transfer is initiated by either TMRN or TMPN → TSMN is sent from I/O Device to CPU indicating end of transfer. ANSF F/F is reset, clock is started and Timeout is reset.

5.13 CLOCK INHIBIT LOGIC (Figure 5-23c)

The Clock Inhibit Logic operates in two different ways depending on whether the IOP or Program Channel is initiating the transfer.

5.14 IOP Transfers

The Clock Inhibit Logic and control signals necessary for starting and stopping the clock during an IOP transfer are shown in the diagram.



To switch off the clock the first Nand-Gate has three inputs which must all be equal to 1.

- Input to pin 5 DMARQN= 0 sets the ANSF F/F output 9=1 to enable the first input.
- Input to pin 3 ROM control signal TMSR=1 is generated at CP time.
- Input to pin 4 the next positive-going QA clock pulse.

With all three inputs = 1 the first Nand-Gate opens and applies an inhibit to pin 2 of the second Nand-Gate which switches off the clock. To switch on the clock, the IOP transfer is complete and DMARQN=1. The ANSF F/F resets which applies logic level 0 to input 5 of the first Nand-Gate. The output pin 6 goes high enabling the second Nand-Gate which switches on the clock.

5.15 Prevent CPU Taking the Bus

During IOP transfers the clock is stopped. This prevents the addressing of any new microinstructions. Therefore signal TMS cannot enable a transfer.

5.16 Program Channel Transfers (Figure 5-3)

The Clock inhibit Logic and control signals necessary for starting and stopping the clock during a Program Channel transfer are shown in the diagram included within this paragraph. To switch off the clock the first Nand-Gate has three

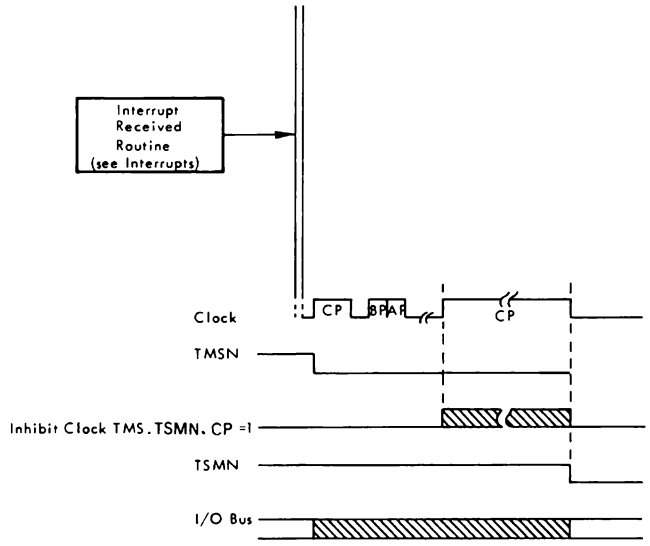
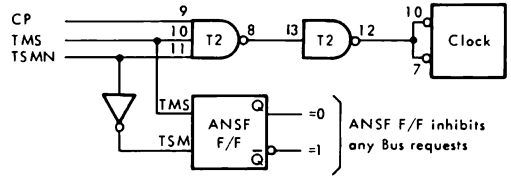


Figure 5-3 Clock is Inhibited during Program Channel Transfer

inputs which must all equal 1.

- Input to pin 11, TSMN=1. TSMN is always equal to 1 except when a data transfer has just ended.
- Input to pin 10, TMS=1, a microinstruction signal enabled at time AP.
- Input to pin 9, CP clock pulse.

With all three inputs equal to 1 the first Nand-Gate opens and applies an inhibit to pin 3 of the second Nand-Gate which switches off the clock. To switch on the clock, the data transfer is completed and TSMN=0 which applies logic level 0 to pin 11 of the first Nand-Gate. The output pin 8 goes high enabling the second Nand-Gate which switches on the clock.

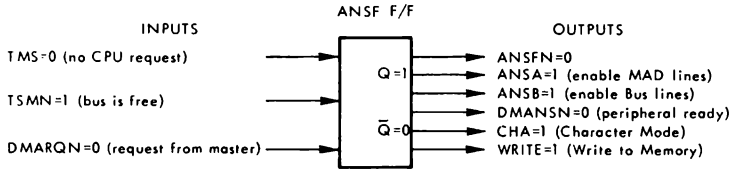


5.17 Prevent I/O Processor Taking the Bus

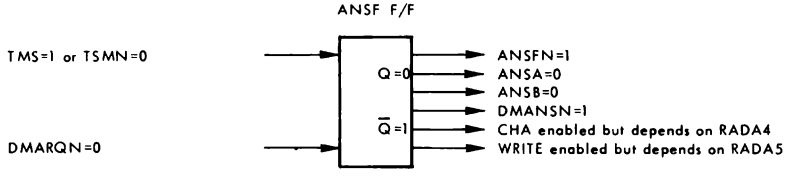
During Program Channel transfers two inhibits are applied to the ANSF F/F pin 12 to hold it in the reset state. TMS holds the ANSF F/F reset during the transfer. TSM is derived from the "end of transfer" interface signal TSMN. TSM holds the ANSF F/F reset for a short time after the CPU transfer is complete.

5.18 ANSF F/F CONTROL LOGIC (Figure 5-23c)

CPU uses the ANSF F/F logic to control the time which the Bus may be transferred between CPU and another master. DMARQN is the request from IOP, and DMANSN the reply from CPU. Assuming the Bus is free and no CPU requests are waiting, the following conditions will exist when DMARQN goes low.



DMARQN goes low but either Bus is busy or there is a CPU request. The following conditions will exist.



5.19 TIMEOUT (Figure 5-23c)

The Timeout logic comprises two monostable multivibrators housed in one logic device 9602. The purpose of Timeout is simply that in the event of an incorrect Memory location being addressed or the TSMN signal not being received by CPU, then CPU can generate an internal TSMN known as TOUT. The way in which the Timeout Logic is used depends on whether the I/O transfer is with IOP or with CPU.

5.20 Timeout for a CPU Transfer (Figure 5-4)

At time CP the ROMs are addressed and two microinstruction signals are generated, TMSN and TMS. TMSN is gated with CPN to output one of the TMSN interface signals to an I/O device. This output is also used to trigger the Timeout 6.4 microsecond monostable. The monostable remains triggered and under normal circumstances is reset by the TSMN interface signal at the end of each transfer. However in the event that no TSMN is received after 6.4 microseconds, the first monostable sets and triggers the 200 nanosecond monostable. 200 nanoseconds later TOUT is generated and used by the CPU in two ways.

- TOUT removes the Clock inhibit starting the CPU Clock.
- As microinstruction signal TMS is still active TOUT opens the Nand-gate and generates signal VN. VN is input to the Planet Condition register.

Note that both monostables are reset by the internal TSMN.

5.21 Timeout for IOP Transfer

Timeout for IOP transfers are the same as CPU except for the initiation and also TOUT is used in a different way. Initiation is by TMPN or TMRN on the bi-directional lines. When Timeout occurs TOUT simulates the end of the transfer signal, TSMN. Note that the VN error signal is not generated.

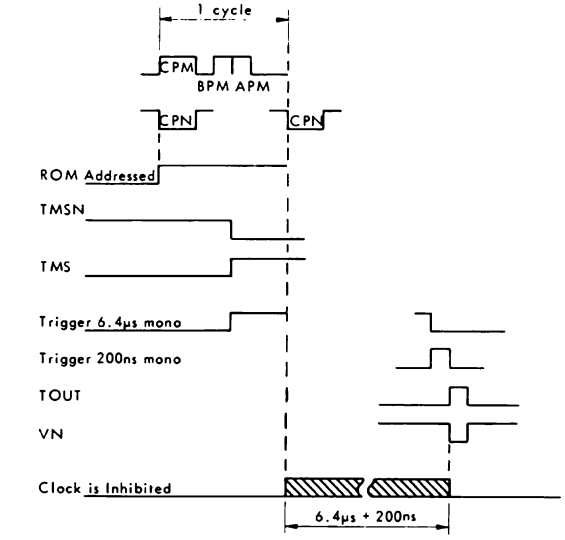


Figure 5-4 No Data Transfer so Timeout Occurs

5.22 INTERRUPTS (Figure 5-23d)

The interrupt system permits switching from one program to another of higher priority. Most of the routine is under microinstruction control though software is used to restore the CPU registers when returning to an interrupted program. There are 16 interrupts, 12 external and 4 internal.

LEVEL

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
↓
61
62
63 CPU Idle

Note that internal interrupts can be connected in any order by means of the wiring on CPU connector No 2. Also both internal and external interrupts can be allocated to any of the 16 lower levels depending on system requirements.

At power on and at each Master Clear the following conditions occur:

- CPU goes to Enable Interrupt Mode.
- Current program level becomes 63.
- All internal and external interrupts are reset.

5.23 Interrupt Received Routine (Figure 5-5)

Microinstruction Signal SCEI=0 enables interrupts. When BPN clock pulse equals 0 the Interrupt Lines IS 0-15 are tested for an interrupt request. When no interrupts are present the IS Lines are at logic level 1. When an interrupt is received then one of the IS Lines will go low and set up a code at the Priority Encoder IEC Lines.

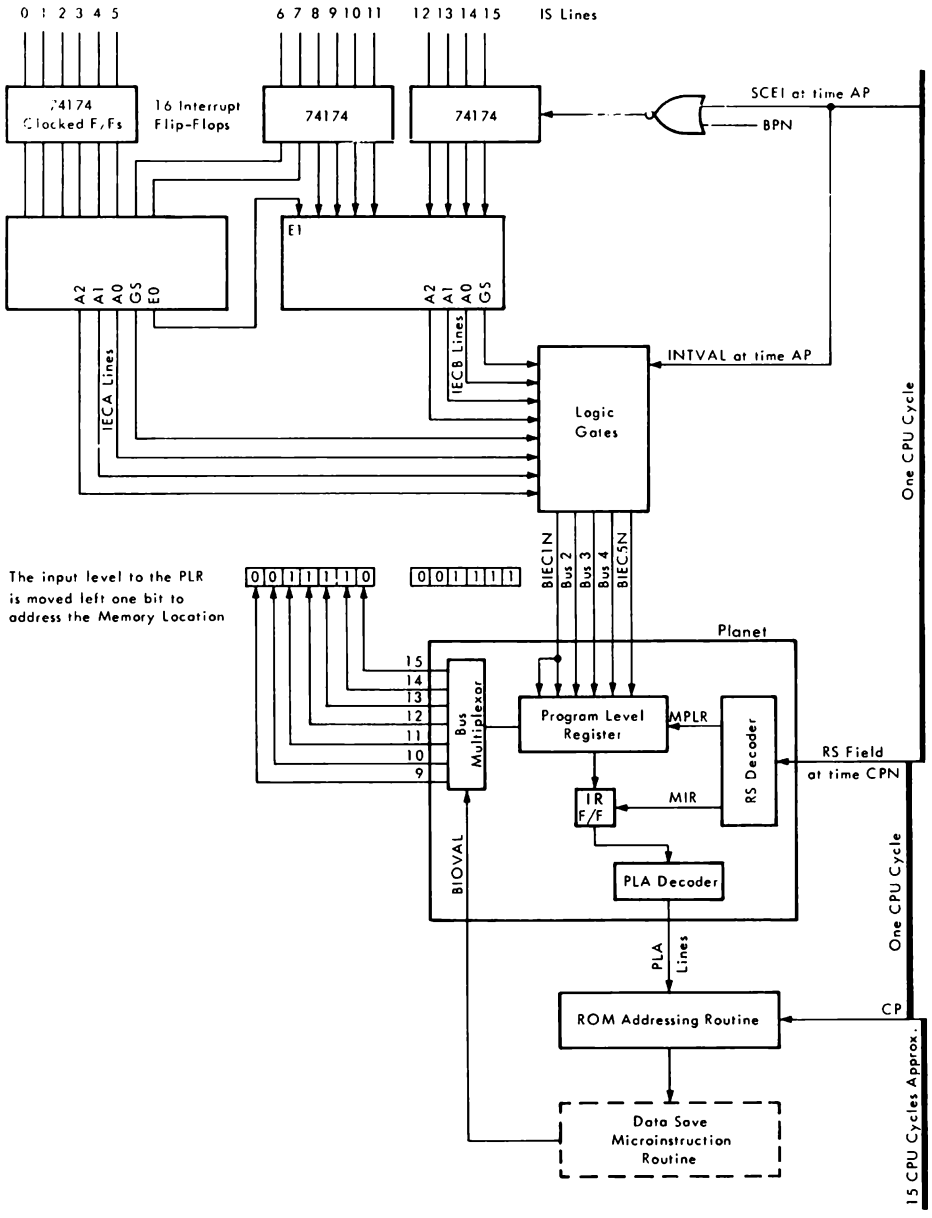


Figure 5-5 Interrupt Received Routine

The decodes for the IEC Lines are shown at the rear of this manual under Logic Element 74148. The Priority Encoders look at the IS Lines in two groups: IS 0-7 and IS 8-15. If a low appears on one of the lines 0-7 then an output code is on the IECA Lines and E0 output is high holding the IECB Lines high. If no interrupts are present on 0-7 the IECA Lines are high, E0 is low which enables a code to be set up by 8-15 on the IECB Lines. Microinstruction signal INTVAL goes high to enable the IEC Lines to be decoded by the Logic Gates. The Truth Table below shows the relationship between the Interrupt Lines and the input to the Program Level Register. The new interrupt level is compared with the level being serviced. If the new interrupt is not at a higher level then the program continues. If it is higher then the Interrupt Flip-Flop will set. At this point the interrupt is valid. PLA Decodes set up the start address of the Data Save Microinstruction Routine.

IS Lines	BIEC0	BUS 1	BUS 2	BUS 3	BUS 4	BIEC5
15	0	0	1	1	1	1
14	0	0	1	1	1	0
13	0	0	1	1	0	1
12	0	0	1	1	0	0
11	0	0	1	0	1	1
10	0	0	1	0	1	0
9	0	0	1	0	0	1
8	0	0	1	0	0	0
7	0	0	0	1	1	1
6	0	0	0	1	1	0
5	0	0	0	1	0	1
4	0	0	0	1	0	0
3	0	0	0	0	1	1
2	0	0	0	0	1	0
1	0	0	0	0	0	1
0	0	0	0	0	0	0

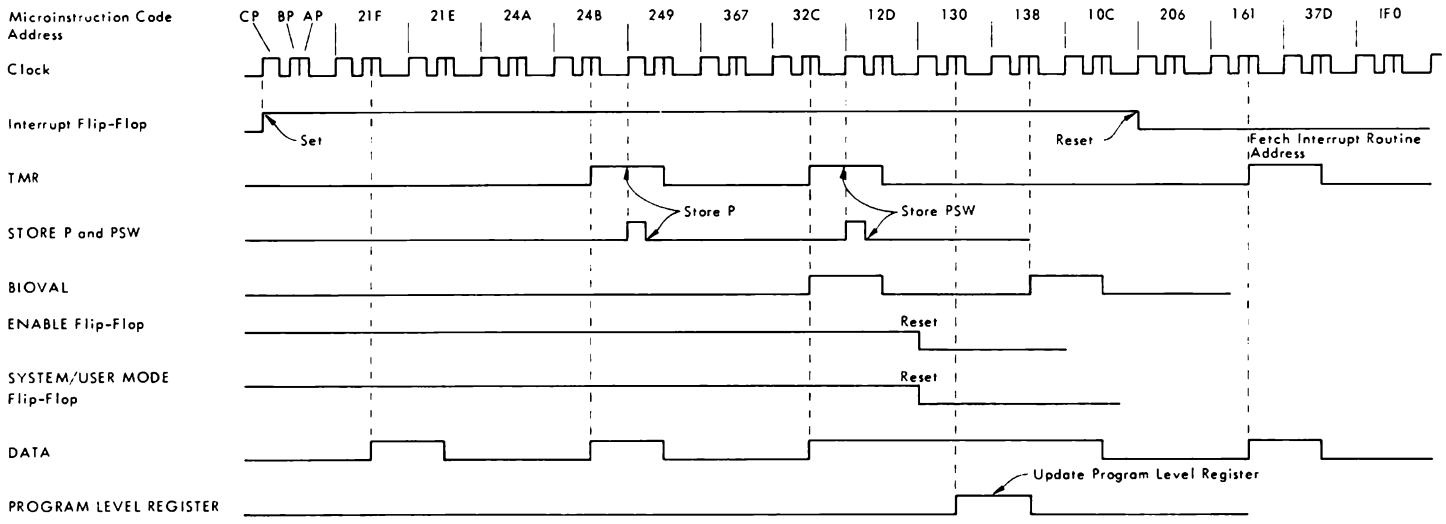


Figure 5-6 Data Save Microinstruction Routine

5.24 Service Interrupt

While an interrupt is being serviced CPU keeps a continual check to see if any higher interrupts are present. This is achieved by generating microcontrol signals SCEI and INTVAL. These signals can enable or inhibit the testing of an interrupt but this will always depend on the type of microinstruction being executed. During the execution of a program the current program level is recorded in the PL register and the Interrupt F/F is reset.

5.25 Return to the Interrupted Routine

When an interrupt program is complete a return must be made to the routine that was interrupted. This is achieved under software control as follows:

- At the end of a routine an instruction must be executed to reset the Interrupt F/F at the input to the Priority Encoders. If an Internal Interrupt then instruction RIT will reset the F/F. If an external interrupt any of the I/O instructions (INR, OTR, SST, CIO, CIO halt) will reset it.
- The contents of the Program Counter and Program Status Word of the routine are written back into the Spalu P register and Planet PSW Registers. Note that when making a return under Monitor control the programs can switch from one to another independantly of the interrupt routine. So in this case P and PSW would not contain interrupted data.

5.26 Internal Interrupts

A full description of the internal interrupts is given in this section under the heading, Internal Interrupts.

5.27 PLANET (Figures 5-7 and 5-23b)

The Planet device is a Large Scale Integrated Circuit. It provides the logic for microprogram addresses, program status word and control of the interrupt routine.

5.28 Program Status Word

PSW is the term given to four of the Planet Registers. It is essential to record the condition of these registers if for example a return to a program has to be made as in the case of the interrupt routine. The PSW Registers are PLR, ENB, FU, CR.

- PLR -- The Program Level Register is a 6-bit register which specifies the level of the program currently being serviced.
- ENB -- The Interrupt Enable/Inhibit Flip-Flop is enabled or inhibited by an input from Bus 9.
- FU -- The System/User Mode Flip-Flop is set to either System or User Mode by an input from Bus 15. CPU must be in System Mode when executing Priviledged Instructions.
- CR -- The Condition Register is used by software to determine a course of action and is used in different ways. For instance during an arithmetic operation the CR has the following meaning:

  CR=00 result = 0
  CR=01 result is less than 0
  CR=10 result is greater than 0
  CR=11 result is an overflow

  However when executing a Control Instruction the CR has the following meaning:

  CR=00 command accepted
  CR=01 command not accepted
  CR=11 device address unknown

For this description consider that Planet has three main functions: executing the RS field of the addressed instruction, setting up the PLA Lines for the ROM address, and bus selection under control of signal BIOVAL.

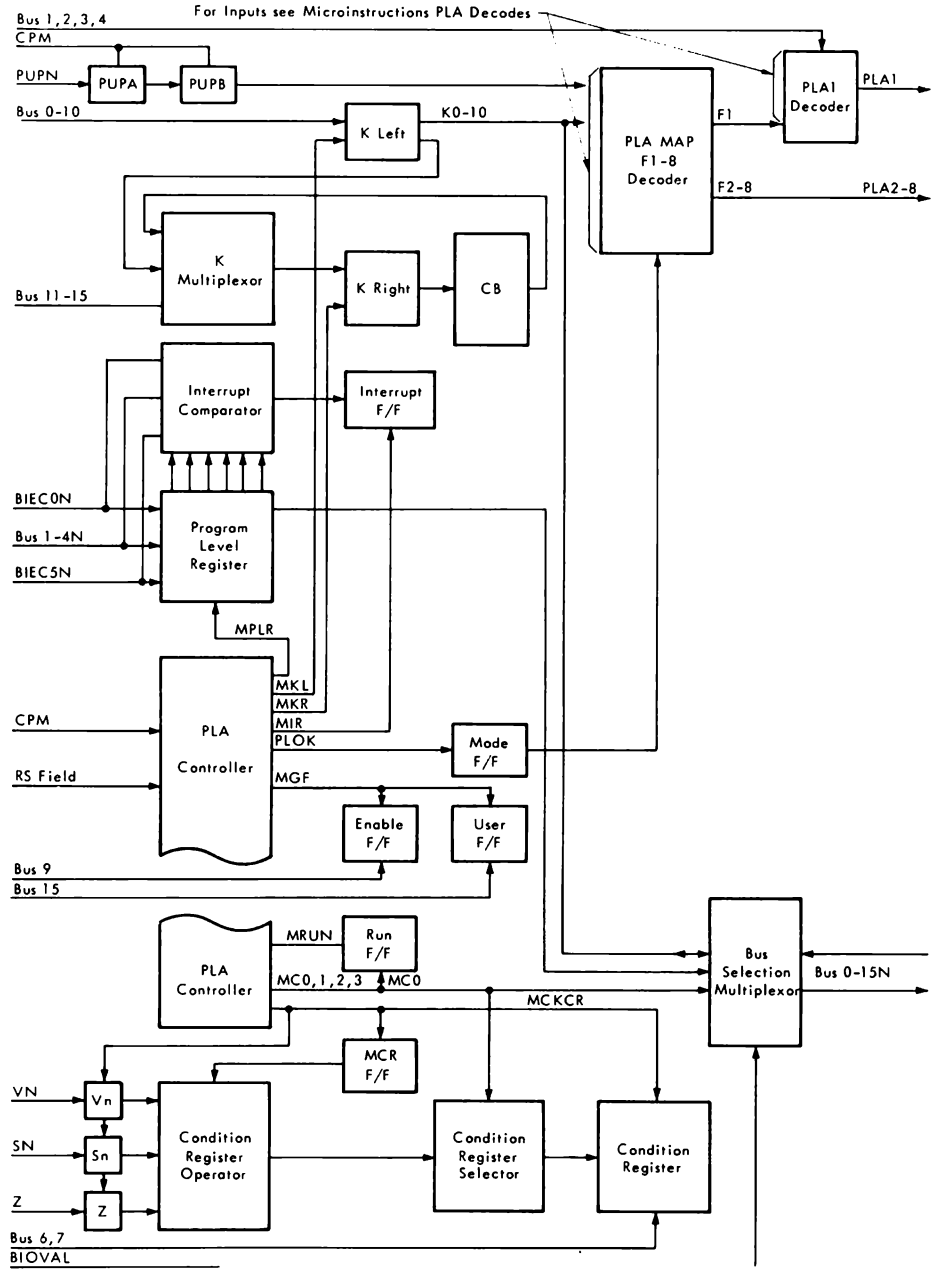


Figure 5-7 Planet Block Diagram

5.29 Planet Decodes the RS Field

The ROM RS field is memorised by Planet at CPN time and decoded by the PLA Controller to give 12 internal signals for use in Planet. These signals are made up with 8 microcommands to load Planet Registers and 4 control lines.

5.30 Control Lines

MC0,1,2,3 are used in two different ways:

- to switch the selectors before the next microinstruction is addressed at CP time.
- as data inputs to the RUN F/F, MODE F/F, and the MCR F/Fs for the next CP.

5.31 Microcommands

- MCKR -- loads the Condition Register in one of three ways: with Bus 6 or 7 of the software control instruction; with VN, SN or Z; with 11 to set CR to decimal 3 which will indicate an overflow condition.

  The CR Selector is controlled by MC1, MC3 which are decodes of the RS field.

MC1	MC3	CR0 Input	CR1 Input
0	0	1	1
0	1	Bus 6	Bus 7
1	0	CRD 0	CRD 1

  When MC1, MC3 are 1and 0 respectively then the Condition Register specifies two consecutive cycles: 1st cycle -- Spalu output flags V specifies whether the operation was Arithmetic, Logic or Compare, is latched on CP (the CR mode is controlled by MC0 and MC2, latched in the MCR0, MCR2 flip-flops); 2nd cycle -- Condition Register is loaded with the outputs of the CR Operator which combines VSZ, MCR0, MCR2 to produce CRD0, CRD1 as in the accompanying table.

Condition Register

	CRD0			CRD1		
	Logic	Arithmetic	Compare	Logic	Arithmetic	Compare
	MC0 MC2	MC0 MC2	MC0 MC2	MC0 MC2	MC0 MC2	MC0 MC2
V S Z	0 0	0 1	1 1	0 0	0 1	1 1
0 0 0	0	0	0	1	1	1
0 0 1	0	0	0	0	0	0
0 1 1	1	1	1	0	0	0
0 1 0	1	1	1	0	0	0
1 1 0	1	1	0	0	1	1
1 1 1	1	1	1	0	1	1
1 0 1	0	1	1	0	1	1
1 0 0	0	1	1	1	1	0

- MPLR -- loads the Program Level Register (PLR). When active, this signal loads the 6-bit PLR via the Bus lines 1-4 and BIEC0 and 5N lines with the new program level to be tested. BIEC5N, the least significant bit, is directly connected from the encodes. Bits 1-4 are sent via the Bus. BIEC0N, the most significant bit, is the same value as Bus 1 as they are linked at the device input.
- MGF -- when active loads two flip-flops: Enable F/F is loaded with Bus 9 of the instruction field; User F/F is loaded with Bus 15.
- MIR -- when active loads the Interrupt F/F with the output of the Interrupt Comparator. IR will set if the Interrupt Enable F/F is set and if there is a higher priority level waiting at the PLR. At the end of the current instruction the PLA Decodes produce the Interrupt Routine address. If the Interrupt F/F is set PLA Decodes produce the microinstruction address which specifies a Fetch from Memory.
- MKL -- when active the left part of K Register (KL) is loaded with Bus 0-10.
- MKR -- when active the right part of K Register (KR) can be loaded in four different ways dependant on the KMUX Selector: with Bus 11-15,

with KL Register bits 5-8, with CB Register bits 0-5, with 11111. As CB is loaded with the contents of KR minus 1 KR can act as a down counter. The KMUX Selector is controlled by MC0, MC1, MC2 as follows:

MC0	MC1	MC2	KR Input
0	1	1	Bus 11 to 15
1	0	1	K5 to K8
1	1	0	CB
1	1	1	1111

- PLOK -- when active the MD0, MD1 flip-flops are loaded with MC1, MC2 control lines. These two flip-flops indicate how the PLA ONE inputs must be decoded.

PLOK	MD0	MD1	Operation
0	0	0	Execute Mode for shifts T8, T1.0R2N Preparation routines for shifts T1.0R2, T2,T3,T4,T5, T6,T7.
1	0	1	Execute Mode for shifts T1.0R2, T2, T3, T4, T5, T6, T7.
2	1	0	Scanning of interrupts and control panel.
3	1	1	Execution of control panel routines.

5.32 Setting up the PLA Lines

The PLA Lines are 8 outputs PLA 1-8 from Planet. Output PLA1, the most significant bit, is decoded separately from the PLA2-8 outputs. The PLA1 value is determined by ROM bits 37, 38, 39 known as the Test Field. A table showing how the values of PLA1 are determined is given in this section under the heading Microinstructions, PLA1 Decodes. A table showing how the PLA2-8 decodes are determined is given under the heading PLA2-8 Decodes.

5.33 Bus Selection

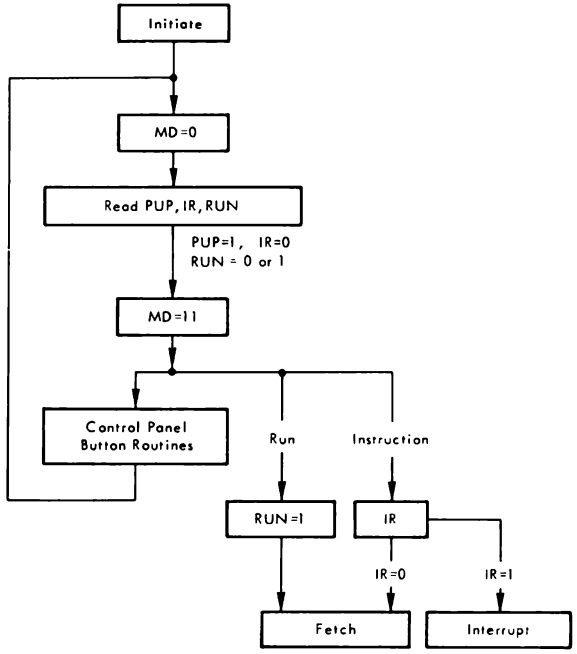
Planet may send or receive information via the Bus depending on the selectors MC0, MC1, MC2, MC3 and microinstruction signal BIOVAL.

Bus Lines	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a	1	0	0	0	0	K12	K13	K14	K11	0	0	K11	K12	K13	K14	K15
b	1	0	0	0	0	K11	K12	K13	K14	0	0	R0	R1	R2	R3	0
c	PLR0	PLR1	PLR2	PLR3	PLR4	PLR5	CR0	CR1	0	ENB	0	0	0	0	0	FU
d	0	0	0	0	0	0	0	0	0	PLR0	PLR1	PLR2	PLR3	PLR4	PLR5	0

	BIOVAL	MC0	MC1	MC2	MC3	Operation
	0	X	X	X	X	None
a	1	1	0	0	0	N on the R1 Field Shift count on BUS11 to BUS15 or 2N on BUS11 to BUS15
b	1	0	1	0	0	R2 on the R1 Field R2 plus one on R2 Field
c	1	0	0	1	0	Program Status Word
d	1	0	0	0	1	Interrupt Address

5.34 Planet Checks for Fetch Interrupt or Panel Routines

During each CPU cycle Planet continually checks its registers to determine the appropriate action to take. The following Flow Chart shows how Planet specifies either a Fetch or Interrupt routine.



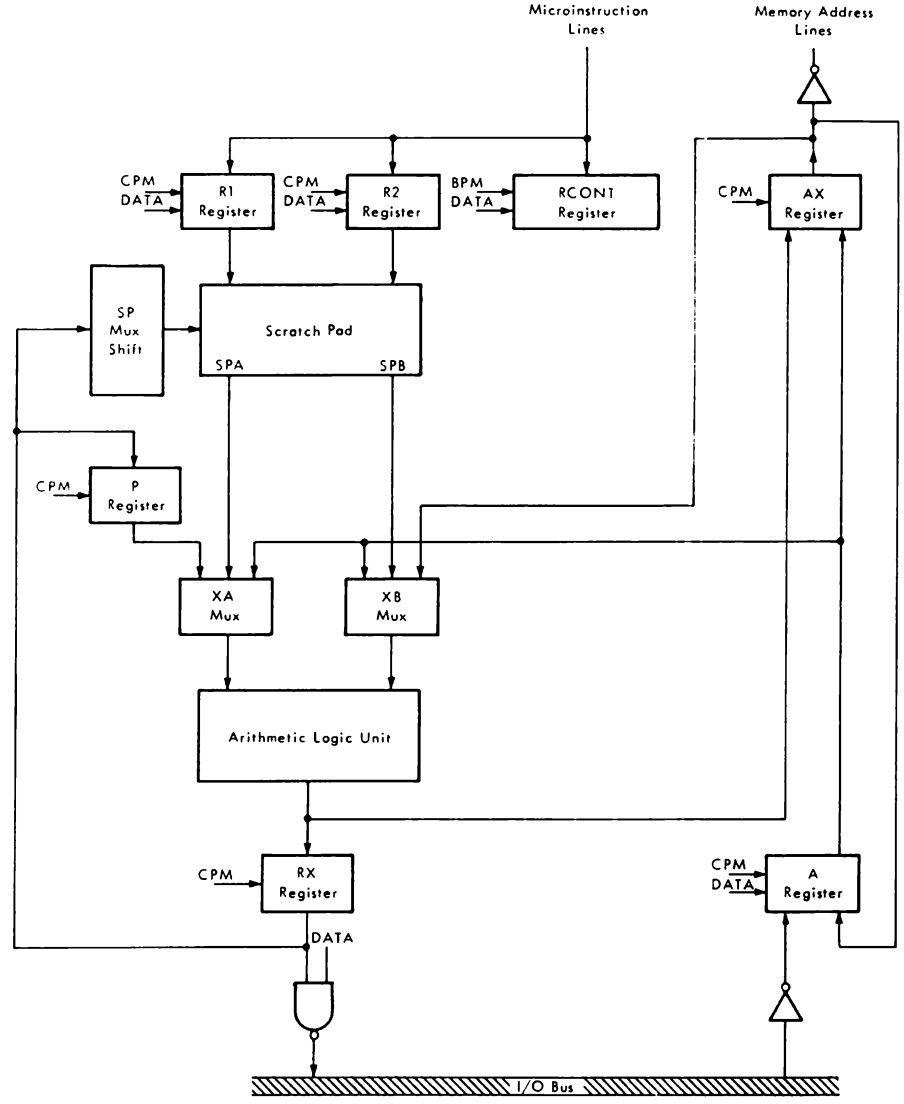


Figure 5-8 Spalu Block Diagram

5.35 SPALU (Figures 5-8 and 5-23a)

CPU employs four Spalu Large Scale Integrated circuits. Each Spalu circuit comprises 4-bits of each register. The connection of the four Spalu devices which gives 16-bit operation is shown in Figures 5-9 and 5-10. In this configuration Spalu has a maximum cycle time of 600 nanoseconds. These four devices will now only be referred to as Spalu in this description. Spalu contains the following groups of registers:

- Scratch Pad registers. These are general purpose registers known as Scratch Pad and numbered A0-A15. A15 is generally used as the Stack Pointer.
- Working registers. These are the four registers used in arithmetic operations. They are: one of the Scratch Pad registers, and registers A, RX, AX.
- Registers that are externally timed by the CPU Clock. They are RCONT, R1, R2, A, RX, P and AX.
- Operators that are only controlled by the microinstruction data loaded into the RCONT Register. These are the multiplexors and Arithmetic Logic Unit.

5.36 Spalu Operation

Although timing of registers is by the CPU Clock or the Microinstruction field overall control is by a microinstruction signal called DATA. It decides whether or not a new microinstruction is to be sent from the Control ROMs to RCONT or whether Spalu is to continue executing the previous microinstruction. DATA achieves this by controlling Spalu's registers as follows:

DATA = 0	DATA = 1
Enable RCONT to be loaded from Internal Bus	Inhibit RCONT
Inhibit data from RX to Bus	Enable data from RX to Bus
Inhibit I/O or Internal Bus to R1, R2	Enable I/O or Int. Bus to R1, R2

Note:

- During the CPU cycle that RCONT is being loaded the other registers are inhibited. During another cycle when RCONT is inhibited the other registers can be loaded.
- DATA does not affect the P register which is the Program Counter or AX register which outputs to the Memory Address Lines. They are always clocked at CP time irrespective of signal DATA.
- R1 and R2 registers can be loaded from either a software instruction via the I/O Bus or from the microinstruction from the Internal Bus.

5.37 Arithmetic Logic Unit (ALU)

The ALU has two inputs XA and XB. Each one is supplied by one of three input selectors which choose the correct operands. The XA Inputs function as follows:

- The Program Counter referred to as the P Register is 15 bits long and although it normally contains the instruction count it may sometimes contain a partial result as in the IPL sequence.
- Scratch Pad SPA outputs the contents of whichever 16 bit register is specified by R1 from the instruction field.
- A Register 16 bits long generally contains an operand from Memory or a partial result.

The XB Inputs function as follows:

- Scratch Pad SPB outputs the contents of whichever register is specified by R2 from the instruction field.
- A Register 16 bits long generally contains the current Memory address or a partial result.

ALU has one output which sends the result of an operation to either AX or RX Register. These contents can now be stored in Scratch Pad P Register sent to A Register or via the Bus to Memory.

5.38 Scratch Pad (SP)

The Scratch Pad has two selectors: one at the Scratch Pad input R1 and R2, and another at the SPMUX SHIFT input. These inputs allow a double length shift to be executed. For example Scratch Pad is loaded with RX shifted one bit left or right,

or A is loaded with AX shifted one bit left or right. Note that all registers are Latch flip-flops. Therefore each operation on a loop must have a source and a destination register. Destination is never the same as the source register. Registers are clocked at CP time only if specified by the microinstruction. When the 12 bit microinstruction word is received by Spalu the contents are tested in three different groups.

- Bits 0-3 -- These four bits decide which of the ALU inputs either XA or XB will be used.
- Bits 4-7 -- These four bits specify the type of operation to be carried out on the source inputs.
- Bits 8-11 -- These four bits prepare the destination register to receive the result of the operation.

5.39 Shift Operations

For shift operations data flow is as shown in Figure 5-9. Shift operations utilise the Scratch Pad, RX, A and AX registers. A shift operation is specified by the microinstruction when Spalu receives the following formats:

	0	1	2	3	4	5	6	7	8	9	10	11
Shift Left					1	0	0	1	1	0	0	1
Shift Right					1	0	1	0	1	0	0	1

To be defined

The type of operation and the destination registers do not change so bits 4-11 are always the same. Only bits 0-3 change to give the explicit type of shift: circular, logical etc. For the microinstruction codes and the types of operation for shifting left and right see Microinstructions, Addressing ROM Bits 24-39.

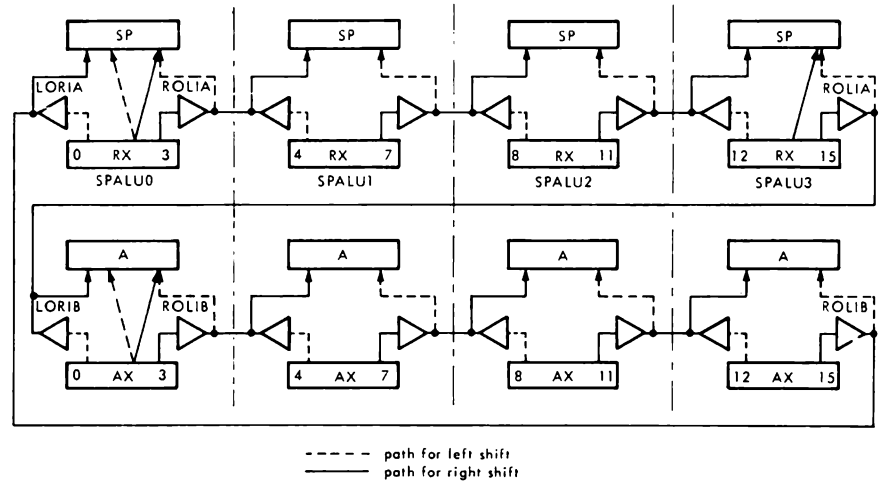


Figure 5-9 Spalu Shift Connections for 16 Bit Configuration

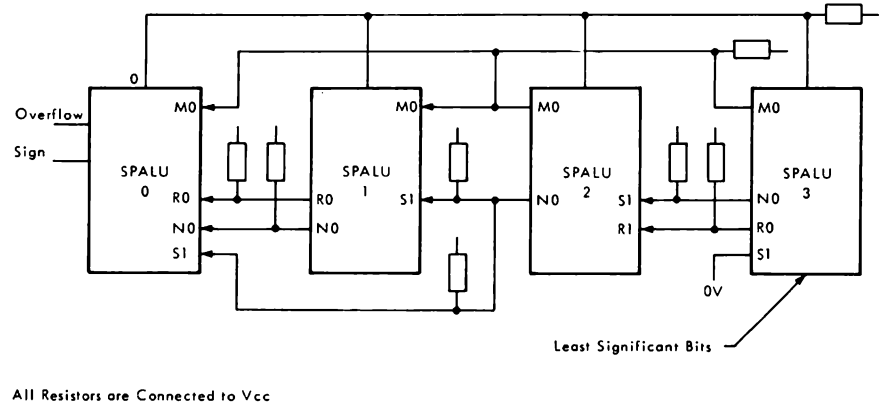


Figure 5-10 Spalu Carry Interconnections for 16 Bit Configuration

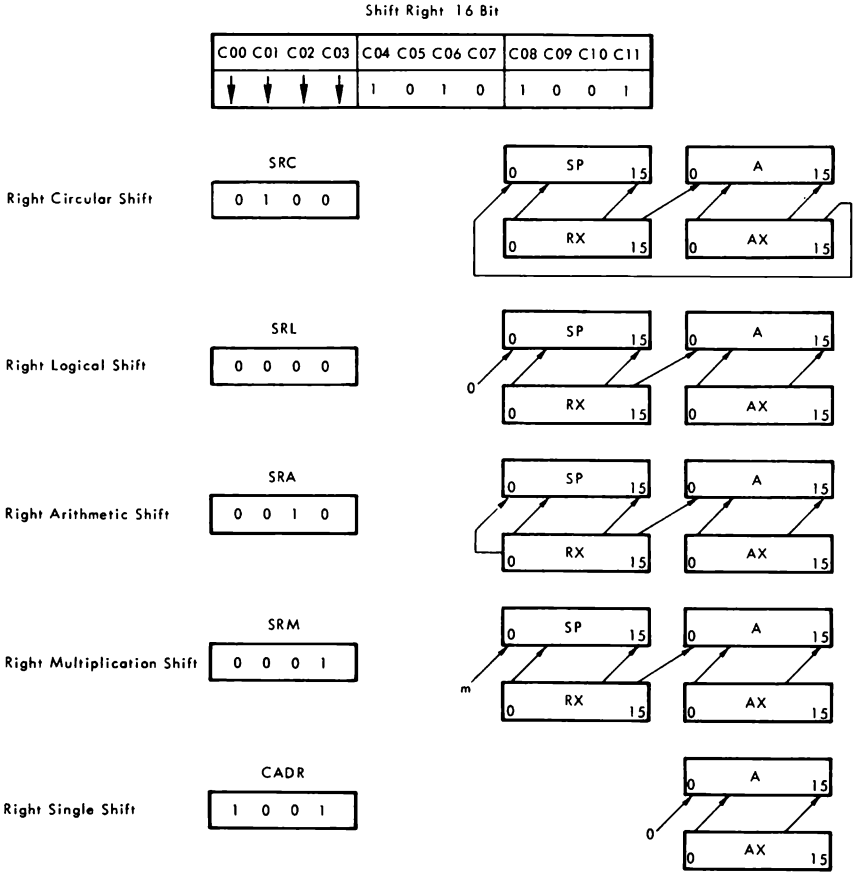


Figure 5-11 Right Shifts

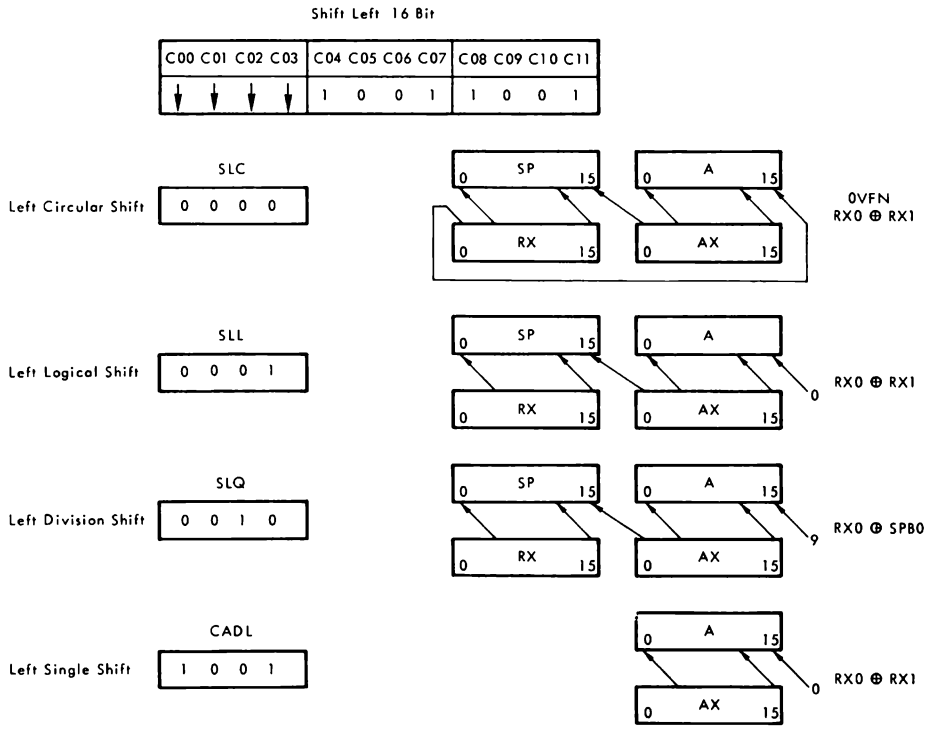


Figure 5-12 Left Shifts

5.40 SWITCH-ON (Figure 5-13)

Power is switched on and signal RSLN goes low and is input to the Control Panel Logic which in turn enables signal MCLN to CPU. CPU uses MCLN to: inhibit the Clock, reset the ROM Address Register to 000, reset the ROM Control Signal flip-flops, and send signal CLEARN to all devices associated with the Bus. Note that as soon as the MCLN pulse goes high it enables the Clock. The ROM Address Register is set to 000 and so the Master Clear Microinstruction routine is addressed. At the end of this routine the Idle Microinstruction Routine is executed. CPU loops in the Idle Routine until the IPL button is depressed and CPU executes the IPL Microinstruction Routine. CPU is now under control of the program loaded by IPL.

5.41 Master Clear at Control Panel

The Master Clear Button is depressed at the Control Panel and the Control Panel Logic sends signal MCLN in exactly the same way as the Switch-On procedure.

5.42 CONTROL PANEL DATA TO CPU

All exchanges between Control Panel and CPU use the Master/Slave principle and CPU is always Master. Depressing one of the following Control Panel buttons will initiate an I/O exchange: INT, RUN, INST, IPL, LM, LR, RM, RR, RST. A description of the function of these switches is given in the Control Panel Section. A data exchange also takes place if either a Power Failure or Automatic Restart interrupt occurs (see Internal Interrupts). Switches 0-15 can be used to set up a memory address or to represent a data word for input to a register. Lamps 0-15 can be used to display the contents of a register or memory location. Switches 1,2,4,8 are used to address one of the Scratch Pad registers A0-15.

5.43 Control Panel to CPU Handshaking Routine (Figure 5-14)

For an Input/Output Exchange CPU executes a Handshaking Routine. It is shown in the following flow chart (Figure 5-14) and operates as follows. The Read Only Memories of the Control Panel receive an input which is decoded to send signal PUPN to CPU. PUPN is used by CPU to set up the address code

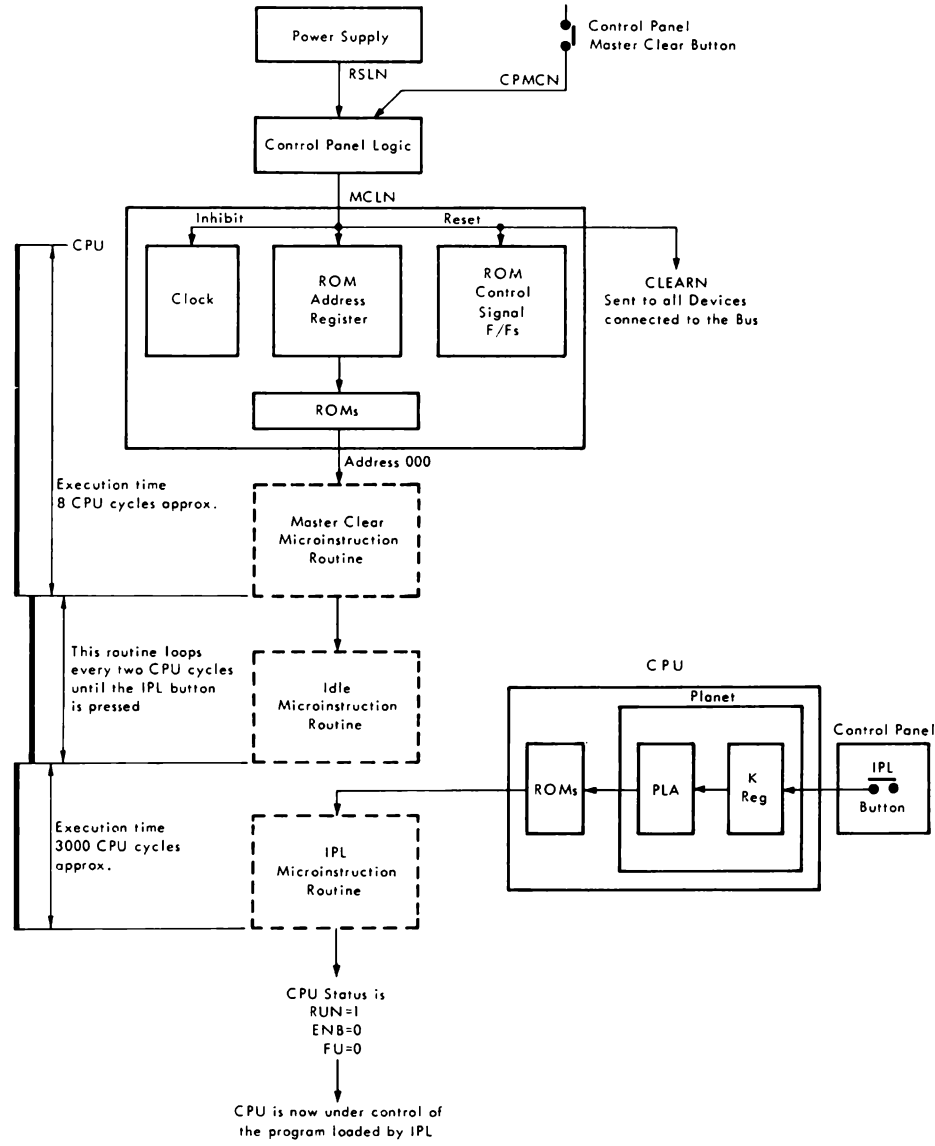
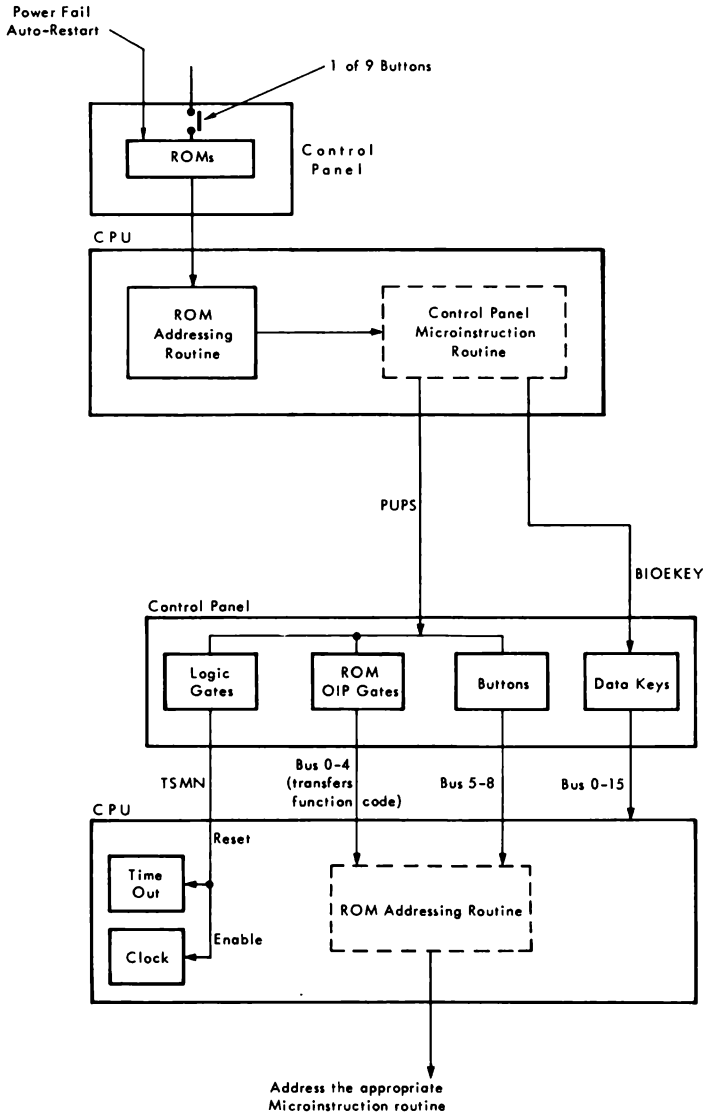


Figure 5-13 Switch-On and Execute IPL Instruction Routine



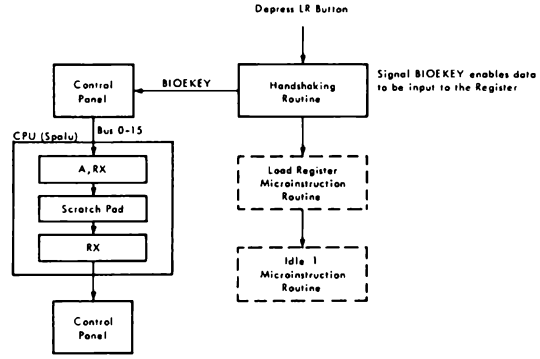
Note: The Real Time Clock Interrupt is hard wired and so does not use this Routine

Figure 5-14 Full Control Panel to CPU Handshaking Routine

for the PUP Microinstruction Routine. Microinstruction signal PUPS enables the I/O transfer and the code from the Control Panel enables CPU to address the appropriate Button Microinstruction Routine. An outline of each Button Routine follows. Detailed information for each of the Microinstruction routines is given at the end of this section.

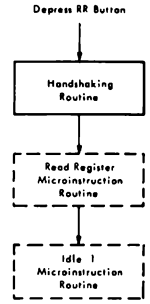
5.44 Load Register Button Routine

A Scratch Pad Register A1-A15 is selected and a 16-bit word is set up at the Control Panel switches. This word has now to be input to the Register.



5.45 Read Register Button Routine

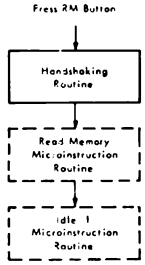
A Scratch Pad Register A1-A15 is selected and the contents are to be displayed on the Control Panel lamps.



5.46 Read Memory / Button Routine

Two stages are involved in this routine.

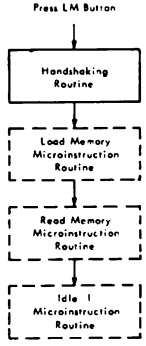
- The memory address to be read must be loaded into the Program Counter. This is achieved by setting up the address on the 16 data switches and also selecting Scratch Pad Register A0. The LR Button is pressed and the Load Register Button Routine is executed. Note that whenever Register A0 is addressed hardware automatically loads the contents into the Program Counter.
- CPU is executing the Idle 1 Microinstruction Routine. The Program Counter contains the memory address to be read.



5.47 Load Memory / Button Routine

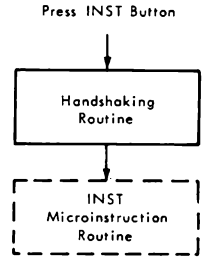
Two stages are involved.

- Identical to the first stage of Read Memory routine.
- CPU is in Idle 1 Microinstruction Routine and the Program Counter contains the memory address to be loaded.



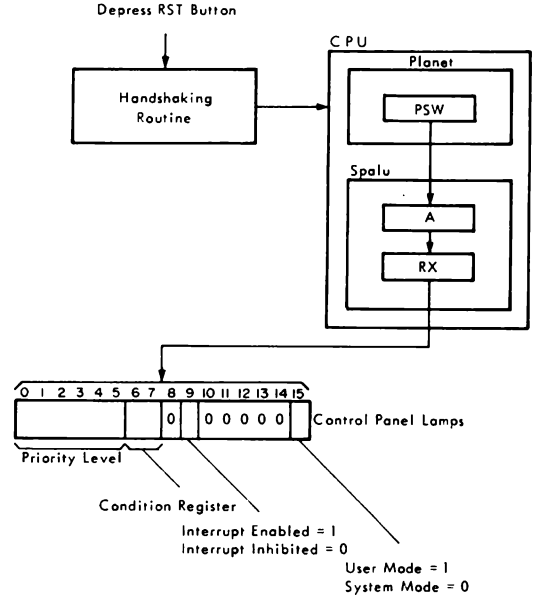
5.48 Instruction Step / Button Routine

Pressing this button causes the Program Counter to address the next instruction in program. Contents of the Program Counter are displayed at the Control Panel lamps.



5.49 Read Program Status / Button Routine

Depressing this key reads the Program Status Word stored in Planet.

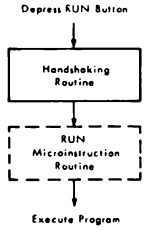


5.50 Initial Program Load/Button Routine

See paragraph entitled Initial Program Loading.

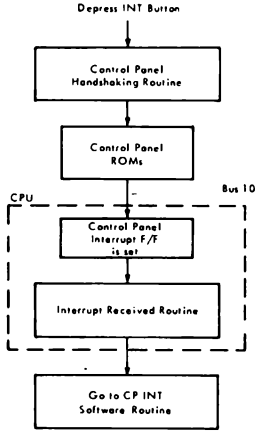
5.51 Program Run/Button Routine

Depressing the RUN button starts CPU executing the program.



5.52 Control Panel Interrupt/Button Routine

Depressing the INT button causes CPU to execute the Handshaking Routine which sets the CP Interrupt Flip-Flop in CPU.



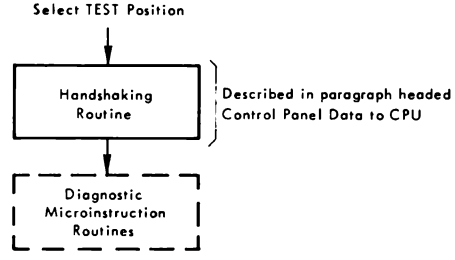
5.53 MINI-PANEL DATA TO CPU

All exchanges between Mini-Panel and CPU use the Master/Slave principle and CPU is always Master. When the following conditions occur a data transfer from Mini-Panel to CPU takes place:

- Power Failure or Automatic Restart (see Internal Interrupt).
- Depressing the INT or START button. INT is the Mini-Panel interrupt (see Control Panel Interrupt/Button Routine). START sets up the code to start the IPL Routine (see Initial Program Loading).

5.54 Test CPU/Button Routine

When the TEST Position is selected CPU executes the Diagnostic Microroutines. A description of these routines is given in the section Microdiagnostics and Troubleshooting.

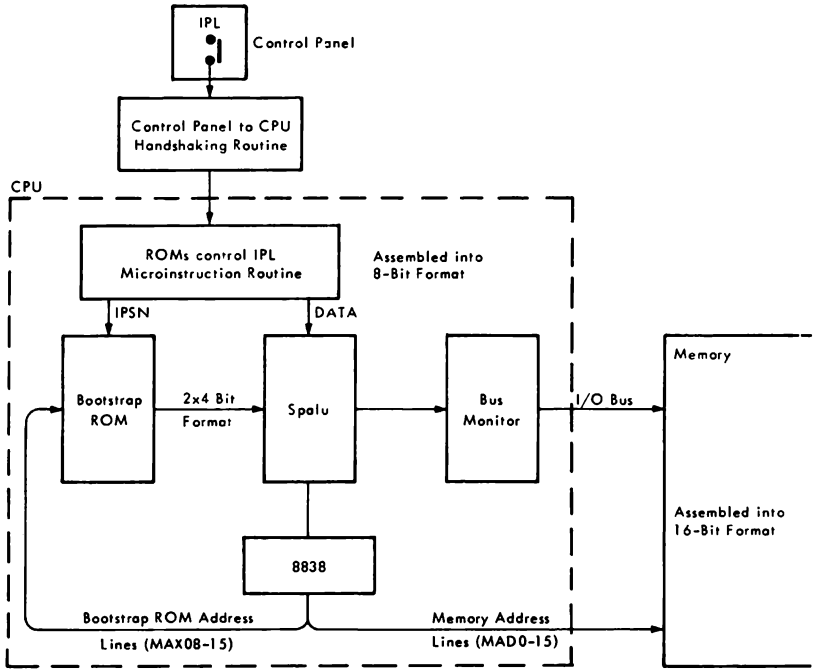


5.55 INITIAL PROGRAM LOADING (Figure 5-15)

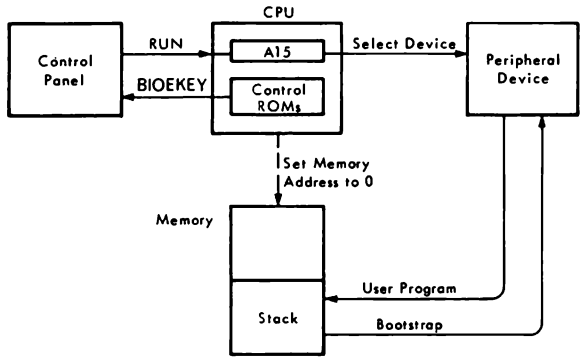
The Initial Program Loader enables the automatic loading of a User or System Program from an external source into Memory. There are two types of program loaders available with the P851:

- Sequential and ROM bootstrap generally used for small configurations may be loaded from the following devices: Mag. Tape (IOP), cassette (PC or IOP), PTR (PC), ASR33 (PC), ROM.
- Sequential and disc bootstrap dedicated for larger systems may be loaded from the following devices: Discs (IOP), Mag. Tape (IOP), cassette (PC or IOP), PTR (PC), Floppy (PC or IOP).

BOOTSTRAP TO MEMORY



USER PROGRAM TO MEMORY



Note: Memory Address set to 0 -- User Program overwrites Bootstrap

Figure 5-15 Initial Program Loading

Table 5-1 Sequential and Disc Bootstrap

```
00000                       IDENT   BEBOOT
00001                   *
00002                   *
00003                   *     MEANING OF THE KEYS DISPLAYED ON THE CONTROL PANNEL
00004                   *     ===================================================
00005                   *
00006                   *
00007                   *  BIT  0  =  0 : CHARACTER EXCHANGE ON PROGRAMMED CHANNEL.
00008                   *             1 : WORD
00009                   *
00010                   *        1  =  0 : IPL FROM A NON-DISK DEVICE.
00011                   *              1 : IPL FROM X1215 OR FLOPPY DISK.
00012                   *
00013                   *        2  =  0 : FIXED HEADS DISK.(FLOPPY IS LIKE FIXED HEADS DISK:NO SEEK)
00014                   *              1 : MOVING HEADS DISK.
00015                   *
00016                   *        3  =  0 : DEVICE ON I/O PROCESSOR (MX).
00017                   *              1 : DEVICE ON PROGRAMMED CHANNEL (CP).
00018                   *
00019                   *  4 TO 7   =  BOU LINES (USE TO QUALIFY AN INSTRUCTION I/O).
00020                   *
00021                   *        8  =  0 : SINGLE DEVICE CONTROLLER.
00022                   *              1 : MULTIPLE DEVICE CONTROLLER.
00023                   *
00024                   *        9  =  0 : NON X1215-DISK USED.
00025                   *              1 : X1215 OR FLOPPY DISK USED.
00026                   *
00027                   *10 TO 15  =  DEVICE ADDRESS.
00028                   *
00029                   * WARNING : NOTE THAT THE BIT 0 AND 3 ARE COMPLEMENTARY.
00030                   * -------
00031                          EJECT
00032                   *
00033                   *
00034                   *
00035                   *
00036                   *     BOOTSTRAP FUNCTION DESCRIPTION 77/08/24
00037                   *     =======================================
00038                   *
00039                   *     THIS NEW BOOTSTRAP ALLOW THE IPL LOADING PROCEDURE
00040                   *     INDIFFERENTLY FROM PROGRAMMED CHANNEL OR I/O PROCESSOR
00041                   *     FOR ANY DEVICES EVEN FLOPPY.
00042                   *
00043                   *     IF THE CONCERNED DEVICE IS NOT READY WHEN PRESSING IPL
00044                   *     BUTTON, BEBOOT LOOPS UNTIL THE DEVICE BECOMES READY.
00045                   *     LOOP MASK :  - NOT OPERABLE.
00046                   *     ---------    - THROUGHPUT ERROR.
00047                   *                  - DATA FAULT.
00048                   *                  - DEVICE HAS BECOME READY.
00049                   *
00050                   *     BEBOOT LOADS ONE RECORD ONTO LOCATION /80 THEN START AT /84.
00051                   *
00052                   *     THE RECORD IS THE SECTOR NUMBER 1 ON DISK, OR 254 CHARACTERS
00053                   *     FROM THE INPUT DEVICE (LEADING NULL CHAR. IS IGNORED) OR SECTOR
00054                   *     NUMBER 0 IF FLOPPY DEVICE USED.
00055                   *
00056                          EJECT
00057                   *
00058                   *
00059                   *
00060                   *
00061                   *                USED  REGISTERS
00062                   *                ===============
00063                   *
00064                   *     A1      =     BOU LINES.
00065                   *     A2      =     A(INR INST).
00066                   *     A3      =     A(CIO INST).
00067                   *     A4      =     A(SST INST).
00068                   *     A5      =     IF MX : FIRST WORD TO BE SENT TO EXTERNAL REGISTER.
00069                   *                   IF CP : CHARACTER COUNT.
00070                   *     A6      =     IF MX : SECOND WORD FOR THE MULTIPLEX.
00071                   *                   IF CP : A (NEXT CHAR. TO BE LOADED).
00072                   *     A7      =     )
00073                   *     A8      =     )  WORK REGISTERS.
00074                   *     A9      =     )
00075                   *     A10 TO A14 :  NOT USED.
00076                   *     A15     =     CONTAINS THE KEYS VALUE FROM THE CONTROL PANNEL.
00077                   *
00078                   *
00079                   * WARNING : A1 MUST NOT BE DESTROYED IF BOOT IS CALLED AGAIN.
00080                   * -------     A3 MUST BE RESTORED
00081                   *            A5 CHAR. COUNT INITIALIZED TO 254 AND DECREMENTED.
00082                   *            A6 ON CP, INITIALIZED TO /80 AND INCREMENTED.
00083                   *
00084                   *
00085                   *
00086                          EJECT
00087                          AORG    0
00088                   *
00089                   * BASE REGISTERS INITIALIZATION.
00090                   *
00091  0000  0200  F    BOOT    LIM     A2,INR          A2 = A(INR INST).
00092  0002  0300  F            LIM     A3,CIO          A3 = A(CIO INST).
00093  0004  0400  F            LIM     A4,SST          A4 = A(SST INST).
00094                   *
00095                   * GET DEVICE ADDRESS TO INITIALIZE THE I/O COMMANDS.
00096                   *
00097  0006  861E               LDM     A6,A15          A6 = KEYS VALUE.
00098  0008  243F               ANK     A6,/3F          KEEP JUST THE DEVICE ADDRESS.
```

Table 5-1 contd.

```
00099  000A  AE29              ORRS    A6,A2       INIT INR INST WITH D.A
00100  000C  AE2B              ORRS    A6,A3       INIT CIO INST.....
00101  000E  AE41              ORS     A6,HIO      INIT HIO.....
       0010  0000  F
00102                  *
00103                  * GET CONTROLLER TYPE AND ADDRESS.
00104                  *
00105  0012  871E              LDR     A7,A15      MULTI OR SINGLE
00106  0014  3FC8              SLC     A7,8        DEVICE CONTROLLER ?
00107  0016  5600  F           RF(6)   INIT20      IF SINGLE --% BRANCH.
00108  0018  260F              ANK     A6,/F       KEEP THE DEVICE CONT. ADDRESS (CA).
00109                  *
00110                  * INITIALIZE THE WER INSTRUCTIONS AND SST ONE.
00111                  *
00112  001A  AE31      INIT20  ORRS    A6,A4       INIT SST INST WITH D.A OR C.A
00113  001C  3E41              SLL     A6,1        COMPUTE EXTERNAL REGISTER ADDRESS.
00114  001E  AE41              ORS     A6,WER1     UPDATE WER INSTRUCTION
       0020  0000  F
00115  0022  AE41              ORS     A6,WER2     WITH THIS EXTERNAL REG.
       0024  0000  F
00116  0026  811C              LDR     A1,A7       A1 = BOU LINES.
00117  0028  0550              LDK     A5,/50      BLOCK LENGTH TO BE LOADED.
00118  002A  0680              LDK     A6,/80      IN LOCATION /80.....
00119                  *
00120                  * CHECK DEVICE TYPE.
00121                  *
00122  002C  3FE7              SRC     A7,7        IS THE DEVICE A DISK OR NOT ?
00123  002E  5600  F           RF(6)   WER1        NO.
00124  0030  3FL1              SLC     A7,1        IS IT A FIXED HEADS DISK ?
00125  0032  5600  F           RF(6)   NOSEEK      YES (NO SEEK TO ZERO).
00126  0034  0103              LDK     A1,3        EXECUTE A SEEK TO ZERO OPERATION
00127  0036  41C0      CIO     CIO     A1,1,0      ON MOVING HEADS DISK.
00128                          EJECT
00129                  *
00130                  * PREPARE THE NEXT LOGICAL I/O FOR A DISK.
00131                  *
00132  0038  811E      NOSEEK  LDR     A1,A15      COMPUTE THE
00133  003A  3966              SRL     A1,6        SECTOR
00134  003C  213C              ANK     A1,/3C      NUMBER.
00135  003E  8520              LDKL    A5,/80CD    FIRST WORD ON MX FOR A DISK.
       0040  80CD
00136                  *
00137                  * EXECUTE THE LOGICAL I/O FOR ANY DEVICE ON MX OR CP INDIFFERENTLY.
00138                  *
00139  0042  7500      WER1    WER     A5,0        FOR ANY DEVICE (DISC OR NOT) EXECUTE
00140  0044  7601      WER2    WER     A6,1        THE WER WHATEVER THE CHANNEL IS.
00141  0046  F031              EXR*    A4          EXECUTE A SST.
00142  0048  F02D              EXR*    A3                  CIO.....
00143  004A  5C06              RB(4)   *-4         REFUSED % TRY AGAIN.
00144                  *
00145                  * CHECK CHANNEL WANTED AND TAKE CORRESPONDING ACTION.
00146                  *
00147  004C  8194              LDR     A9,A5               A9 = NBR OF CHAR TO READ..
00148  004E  871E      MX::CP  LDR     A7,A15              IS THE DEVICE CONNECTED
00149  0050  3F43              SLL     A7,3        ON MULTIPLEX CHANNEL ?
00150  0052  5600  F           RF(6)   SST         YES. SEND A SST.
00151                  *
00152                  * INPUT ONE CHARACTER FROM THE DEVICE.
00153                  *
00154  0054  4F00      INR     INR     A7,0,0      READ ONE CHAR. IN A7.
00155  0056  5400  F           RF(4)   SST         REFUSED % SEND A SST TO KNOW WHY.
00156  0058  E994              CWR     A9,A5       LEADING CHAR. ?
00157  005A  5400  F           RF(4)   INR10       NO.
00158  005C  871C              LDR     A7,A7       YES. NO CHECK WETHER
00159  005E  580C              RB(0)   INR         WORD OR CHARACTER EXCHANGE.
00160  0060  87YE      INR10   LDR     A15,A15     IS IT IN CHARACTER MODE ?
00161  0062  5600  F           RF(6)   STORE       YES.
00162  0064  8739              STR     A7,A6       STORE THE WORD JUST READ ON CP.
00163  0066  1601              ADK     A6,1                USE SAME ROUTINE (WORD OR CHAR. ON CP)
00164                  *
00165                  * STORE CHARACTER JUST READ AND CHECK END OF TRANSFER.
00166                  *
00167  0068  E739      STORE   SCR     A7,A6       STORE ONE CHAR. TO /80 + (A6).
00168  006A  1601              ADK     A6,1        NEXT LOCATION TO BE ERASED.
00169  006C  1D01              SUK     A5,1        IS THERE ANOTHER CHAR. TO READ ?
00170  006E  5C1C              RB(4)   INR.        YES. LOOP ON INPUT.
00171                  *
00172                  * END OF TRANSFER - EXECUTE A CIO HALT.
00173                  *
00174  0070  4180      HIO     CIO     A1,0,0      CIO HALT TO STOP THE EXCHANGE.
00175                          EJECT
00176                  *
00177                  * EXECUTE A SST AND ANALYZE THE STATUS TO LOOP F.E : A READY IS SET.
00178                  *
00179  0072  4FC0      SST     SST     A7,0        SEND STATUS.
00180  0074  5C28              RB(4)   MX::CP              REFUSED % THIS ALLOW TO HANDLE THE
00181                  *                                   BOTH CASES (WAIT ON CP OR MX).
00182  0076  A720              ANKL    A7,/4007    TEST : BIT 1 = DEVICE HAS BECOME READY.
       0078  4007
00183                  *                                   13 = DATA FAULT.
00184                  *                                   14 = THROUGHPUT ERROR.
00185                  *                                   15 = NOT OPERABLE.
00186  007A  0C00              AB(4)   /00         ONE OF THESE CONDITIONS THEN
00187                  *                           START BOOTSTRAP AGAIN FROM THE
00188                  *                           VERY BEGINNING.
00189  007C  0F84              AB      /84         NORMAL END BRANCH TO IPL ROOT.
00190                          END     BOOT
```

Table 5-2 Sequential and ROM Bootstrap

```
00000                          IDENT   BOOT51
00001                  *
00002                  *
00003                  *       DISPLAY THE KEYS AS FOLLOWS:
00004                  *
00005                  *       BITS    MEANING
00006                  *       0=1     IPL LOADED FROM ASR ; FORMAT 4*4
00007                  *       1=1     START PROGRAM IN ROM
00008                  *       2=X     NOT SIGNIFICANT
00009                  *       3=1     PROGRAMMED CHANNEL
00010                  *       3=0     I/O PROCESSOR
00011                  *       4 TO 7  BOU LINES ( 4 RIGHTMOST BITS )
00012                  *       8=1     MULTI DEVICE CONTROLLER
00013                  *       8=0     SINGLE DEVICE CONTROLLER
00014                  *       9=X     NOT SIGNIFICANT
00015                  *       10 TO 15 DEVICE ADDRESS
00016                  *
00017                  *
00018                  *
00019                  *       DESCRIPTION
00020                  *
00021                  *       BOOT51 LOADS ONE RECORD ONTO LOCATION /80 THEN START AT /84
00022                  *       IF A PROGRAM STORED IN ROM IS TO BE STARTED,THE BOOT51
00023                  *       FETCHES FOR THE FIRST ROM LOCATION WHICH HOLDS THE
00024                  *       STARTING ADDRESS OF THE PROGRAM  AND  THE BRANCH IS
00025                  *       PERFORMED AT THIS ADDRESS.
00026                  *       FOR STARTING THE PROGRAM IN ROM  ONLY KEY BIT=1 IS SIGNIFICANT
00027                  *
00028                  *
00029                  *
00030                  *
00031                  *       USED REGISTERS :
00032                  *
00033                  *       A1 BOU LINES,NOT TO BE DESTROYED IF BOOT IS CALLED AGAIN
00034                  *       A2 ADDR OF INR INSTRUCTION
00035                  *       A3 ADDR OF CIO INSTRUCTION(WHICH IS DESTROYED AND NEEDS TO BE
00036                  *                  RESTORED IF BOOT IS CALLED AGAIN)
00037                  *       A4 ADDR OF SST INSTRUCTION
00038                  *       A5 MULTIPLEX : CONTENTS OF 1ST WORD TO BE SENT TO EXT REGISTER
00039                  *          IO BUS : CHARACTER COUNT,INITIALIZED AT 254 AND DECREMENTED
00040                  *       A6 MULTIPLEX : CONTENTS OF 2ND WORD TO BE SENT TO EXT REGISTER
00041                  *                  (LOADING ADDR)
00042                  *          IO BUS : ADDR OF NEXT CHAR TO BE LOADED,INIT AT /80 AND
00043                  *                   INCREMENTED
00044                  *       A7,A8 WORK REGISTERS
00045                  *       A9 WORK REGISTER
00046                  *       A10 TO A14 NOT USED
00047                  *       A15 CONTAINS THE KEYS*VALUE
00048                  *
00049                  *
00050                  *
00051                  *
00052                  *
00053                  *
00054                  *
00055                          EJECT
00056                          AORG    0
00057                  *
00058                  *
00059                  BOOT    EQU     *
00060                  *       INITIALIZE REGISTERS
00061  0000  0200  F           LDK     A2,INR      ADDR OF INR INSTRUCTION
00062  0002  0300  F           LDK     A3,CIO      ADDR OF CIO INSTRUCTION
00063  0004  0400  F           LDK     A4,SST      ADDR OF SST INSTRUCTION
00064                  *                           EXTRACT DEVICE ADDR AND INIT I/O COMMANDS
00065  0006  861E              LDR     A6,A15
00066  0008  263F              ANK     A6,/3F      DEVICE ADDR
00067  000A  9629              ADRS    A6,A2       INITIALIZE I/O INSTRUCTIONS
00068  000C  962D              ADRS    A6,A3
00069  000E  9641              ADS     A6,HIO
       0010  0000  F
00070                  *                           EXTRACT CONTROLLER ADDR AND INIT WER INST
00071  0012  871E              LDR     A7,A15
00072  0014  3FC8              SLC     A7,8        MULTI OR SINGLE DEVICE CONTROLLER
00073  0016  5600  F           RF(6)   INIT20      SINGLE ONE
00074  0018  260F              ANK     A6,/F       MULTIPLE ONE
00075                  INIT20  EQU     *           INITIALIZE MULTIPLEX DBLE WORD:
00076  001A  9631              ADRS    A6,A4       SST INSTRUCTION
00077  001C  3E41              SLL     A6,1
00078  001E  9641              ADS     A6,WER1     SET UP WER INSTRUCTIONS
       0020  0000  F
00079  0022  9641              ADS     A6,WER2
       0024  0000  F
00080                  *                           LOAD A1 WITH BOU CONTENTS
00081  0026  811C              LDR     A1,A7
00082  0028  0550              LDK     A5,80       MULTIPLEX DOUBLEWORD: LOAD 80 CHAR INTO
00083                  *                           LOCATION /80
00084  002A  0680              LDK     A6,/80
00085                  *                           CHECK IF  ROM
00086  002C  3FE7              SRC     A7,7
00087  002E  5600  F           RF(6)   NOROM       NO
00088                  *
00089                  * ROM   BOOT:  FETCHES FOR THE FIRST ROM LOCATION
00090                  *
00091                  LOOP    EQU     *
00092  0030  1602              ADK     A6,2
00093  0032  8238              LDR*    A2,A6       TAKE CONTENTS OF CURRENT LOCATION
00094  0034  5806              RB(0)   LOOP                ROM CONTENTS=0 NOT SIGNIFICANT VALUE
```

Table 5-2 contd.

```
00095  0036  9039              INR     A6                      TRY TO MODIFY THE CURRENT MEMORY LOC.CO
00096  0038  EA38              CWR*    A7,A6           TEST IF STORE WAS SUCCESSFUL
00097  003A  8816              ABR(0)    A6              NO,THE LOCATION CONTENTS WAS NOT CHANGED
00098  003C  8239              STR     A7,A6                   RESTORE PREVIOUS MEMORY CONTENTS
00099  003E  5F10              RB      LOOP                    GO TO TEST NEXT MEMORY LOCATION
00100                    *                               IT IS ROM LOCATION
00101  0040  41C0        CIO     CIO     A1,1,0          CIO SEEK ZERO
00102                    NOROM   EQU     *
00103                    *                               EXECUTE WER,WHATEVER THE CHANNEL IS
00104                    WER1    EQU     *
00105  0042  7500              WER     A5,0
00106                    WER2    EQU     *
00107  0044  7601              WER     A6,1
00108  0046  F031              EXR*    A4              SST
00109  0048  F020              EXR*    A3
00110  004A  5C06              RB(4)   *-4
00111  004C  871E              LDR     A7,A15
00112  004E  3F43              SLL     A7,3
00113  0050  5600  F           RF(6)   TST             MULTIPLEX
00114                    *
00115                    *                               TO BUS
00116                    *
00117  0052  8194              LDR     A9,A5           IF A9=A5 IGNORE LEADING CHAR.
00118                    INR     EQU     *
00119  0054  4F60              INR     A7,0,0          READ ONE CHAR
00120  0056  5C04              RB(4)   *-2
00121  0058  E994              CWR     A7,A5           LEADING CHAR?
00122  005A  5400  F           RF(4)   INR10           NO
00123  005C  27FF              ANK     A7,/FF          CHECK IF NULL
00124  005E  580C              RB(0)   INR             YES,IGNORE
00125                    *                               NO,CHECK IF ***
00126                    *
00127                    INR10   EQU     *
00128  0060  879E              LDR     A15,A15         ***
00129  0062  5600  F           RF(6)   STORE           NO B+B
00130  0064  3F44              SLL     A7,4            YES,***
00131  0066  809C              LDR     A8,A7           SAVE LEFT BITS
00132  0068  F029              EXR*    A2              READ NEXT CHARACTER
00133  006A  5C04              RB(4)   *-2
00134  006C  270F              ANK     A7,/F           GET 4 RIGHT MOST BITS
00135  006E  9702              ADR     A7,A8
00136                    STORE   EQU     *
00137  0070  E739              SCR     A7,A4           STORE CHAR
00138  0072  1601              ADK     A6,1            NEXT CHAR ADDR
00139  0074  1401              SUK     A5,1            COUNT DONE ?
00140  0076  5924              RB(1)   INP             NO
00141                    *                               YES
00142                    *
00143  0078  4180              MIO     CIO     A1,0,0
00144                    *
00145                    STATUS  EQU     *
00146  007A  4FC0              SST     SST     A7,0
00147  007C  5C04              RB(4)   *-2
00148  007E  0F84              AB      /84
00149                    *
00150                    *
00151                    *
```

SYMBOL TABLE

```
BOOT    0000  A   INR     0054  A   CIO     0040  A   SST     007A  A
MIO     0078  A   INIT20  001A  A   WER1    0042  A   WER2    0044  A
NOROM   0042  A   LOOP    0030  A   INR10   0060  A   STORE   0070  A
STATUS  007A  A
```

This routine uses the following logic:

- ROMs -- contain the IPL microinstruction routine which has overall control and generates the timing signals IPSN and DATA. Bootstrap ROM contains the Bootstrap routine in the format 256x4 bits.
- Spalu -- sends two consecutive addresses on the Memory Address Lines to read 2x4 bit format from the Bootstrap ROM.

5.56 CPU OPERATING MODES (Figures 5-16 and 5-17)

CPU operates in two modes, System and User. Priviledged instructions may only be executed in System Mode. Any attempt to execute a privileged instruction in User Mode will result in a Trap.

5.57 Privileged Instructions

The P851 set of instructions are executed either under Monitor Control in System Mode or User Program Control in User Mode. Those instructions which must be executed under Monitor Control are called Privileged Instructions and are as follows:

- Instructions which modify the CPU state HLT, INH, RIT.
- I/O instructions CIO, OTR, INR, SST, TST.
- External register instructions WER, RER.
- Instructions which modify the contents of the Stack Pointer LDR, STR, MLR, MSR, RTN in format T3B and Shift and Normalise with R2=15.

5.58 System and User Modes

CPU has two modes of operation, System and User. Whenever a Privileged Instruction needs to be executed CPU must first be switched to System Mode. Data from the execution of an LKM instruction is input to Planet via the I/O Bus and the LKM microinstruction routine is addressed. Simultaneously the Program Interrupt flip-flop is set. During the microinstruction routine interrupts are enabled with SCEI and ITV and the Interrupt code is sent to Planet on the BIEC lines. At this point CPU is in System Mode. The interrupt code has been received so the Data Save microinstruction routine is addressed saving PSW in the Memory Stack. At the end of the routine the LKM software routine is addressed. After the Monitor routine is complete a return must be made to User Mode. This is achieved by

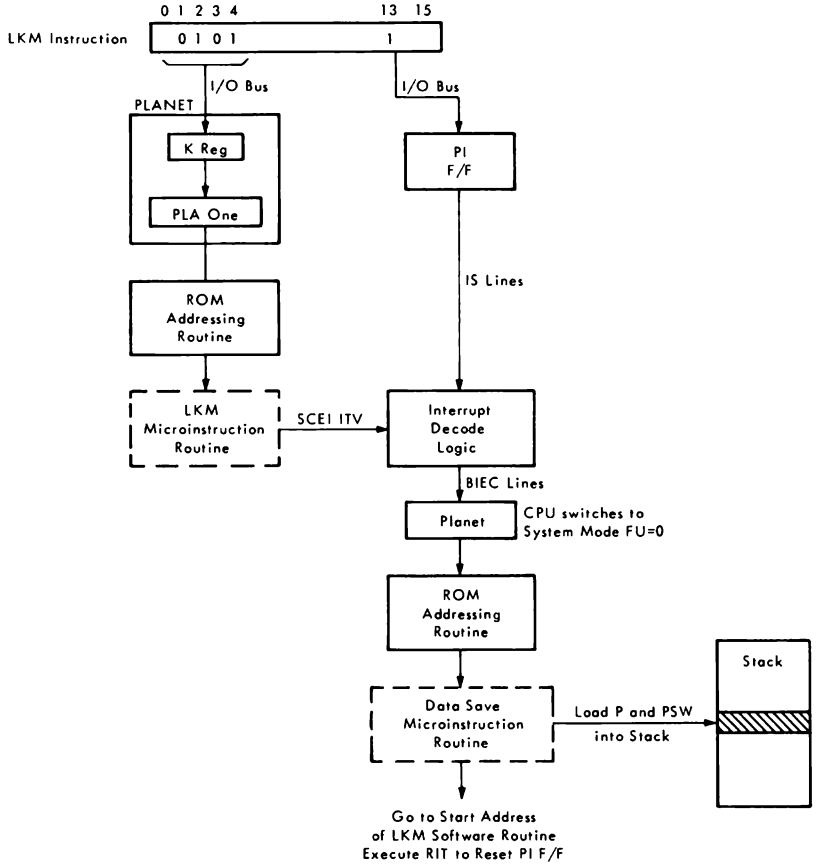
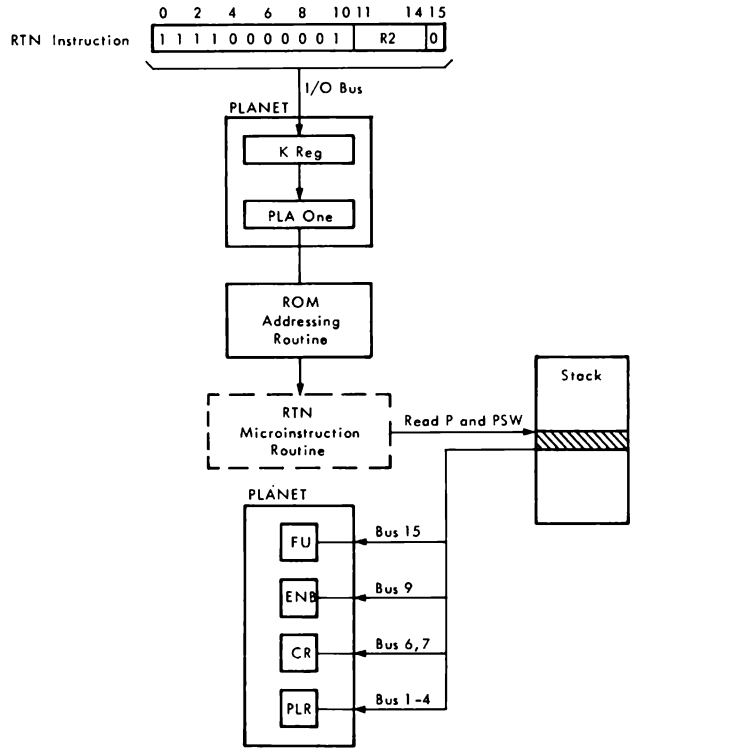


Figure 5-16 Execute LKM -- CPU goes to System Mode



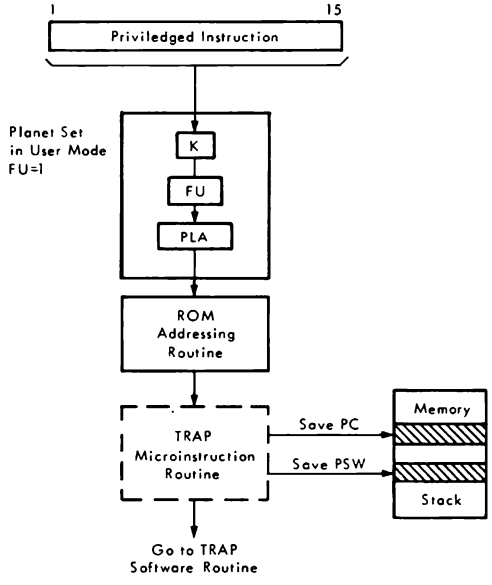
Note that on a return to User Mode bit 15 in the PSW must be logic 1 or CPU will not switch over

Figure 5-17 Execute RTN -- CPU returns to User Mode

executing an RTN software instruction. The RTN sets up the Planet PLA Register and addresses the RTN microinstruction routine. During this routine the PSW which was stored in the Stack is addressed and input to Planet via the I/O Bus. PSW data restores the PSW registers of Planet. CPU is now back in User Mode and the Program can continue.

5.59 Executing a Trap Routine

Any attempt to execute a Privileged Instruction or to address Scratch-Pad Register A15 without first switching to System Mode causes a Trap. When a software instruction is on the Bus, Planet always checks flip-flop FU. If CPU is in User Mode and Planet recognises an invalid instruction, the TRAP microinstruction routine is addressed. This routine saves PC and PSW. At the end of the routine the TRAP software routine is addressed which will probably specify an ABORT, dump the contents of Memory, and halt CPU.



5.60 INTERNAL INTERRUPTS (Figure 2-23d)

There are four internal interrupts available in the CPU.

- PFF (Power Failure / Automatic Restart) -- Set and reset via the I/O Bus.
- CPF (Control Panel Interrupt) -- Set and reset via the I/O Bus (see Control Instructions).
- PIF (Program Interrupt) -- Set and reset via the I/O Bus (see CPU Operating Modes).
- RTC (Real Time Clock) -- Set by Power Supply and reset via the I/O Bus.

Each of these interrupts is latched in a flip-flop type 74279 situated on the CPU board. The output of each F/F is used to form an input to the Interrupt Lines. All F/Fs are reset by executing a software RIT instruction.

5.61 POWER FAILURE / AUTOMATIC RESTART (Figures 5-18 and 5-19)

If a mains power failure occurs it will not be registered as a power failure unless the duration is more than 10mS. If it does exceed 10mS then the Power Supply Control Logic sets signal PWFN low. PWFN is input to the Control Panel to latch the Power Fail flip-flop PWF. If the Automatic Restart facility is fitted then the ARF flip-flop is latched but not used at this stage. Both of these routines use the control logic of the Power Supply, Control Panel, CPU, and also include the execution of microinstruction and software routines. A block diagram in flow chart form, Figure 5-18, shows the series of events.

5.62 Power Failure Routine

A power failure has occurred and the Power Supply sends signal PWFN which sets the PWF flip-flop at the Control Panel and in turn addresses the Control Panel ROMs. The Control Panel to CPU hanshaking routine is executed which sets Bus 11 = 1, which in turn sets the PFF flip-flop at CPU. PFF flip-flop sets a high priority interrupt on the IS Lines which initiates the Interrupt Received Routine. At the end of this routine the Power Failure software routine is addressed which contains an RIT software instruction. This instruction resets the PFF flip-flop at CPU. Shut-down procedure is now complete.

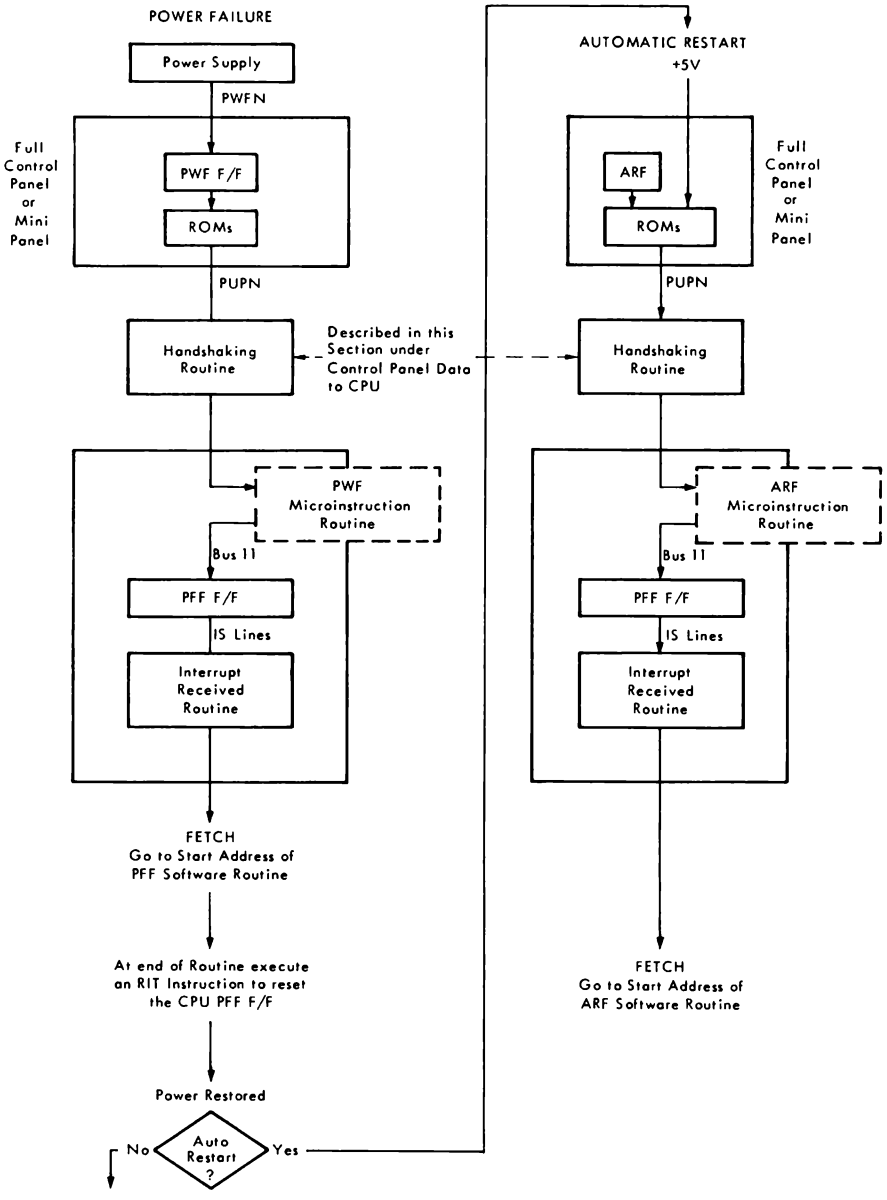
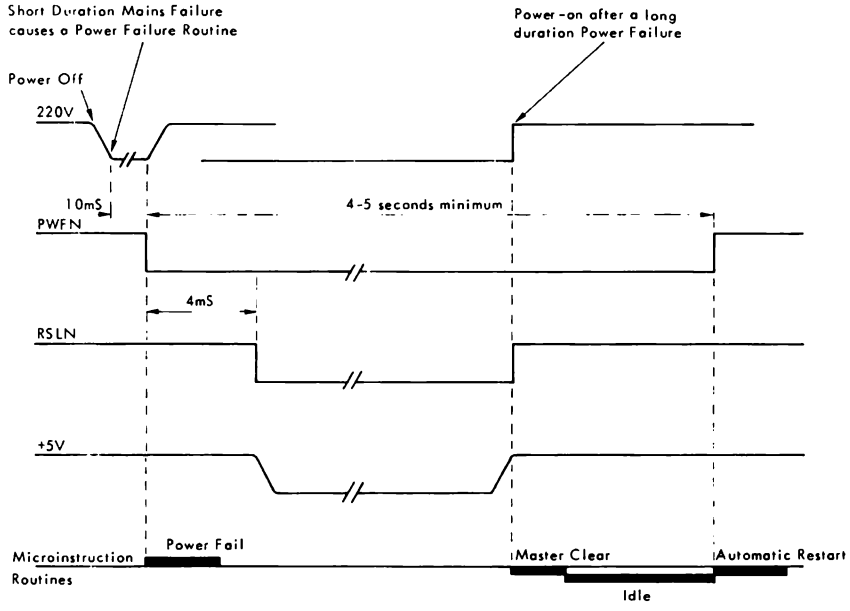


Figure 5-18 Power Failure and Automatic Restart



Note that the microinstruction routines are shown relative to the conditions that cause their initiation and are not intended to be representative of execution times.

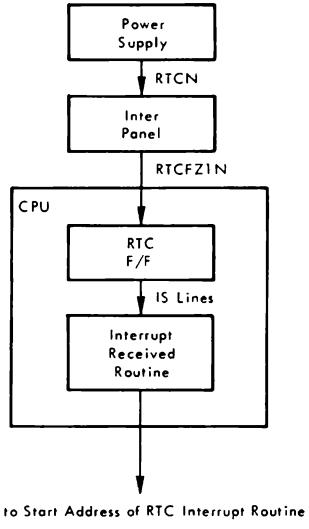
Figure 5-19 Timing -- Power Failure / Automatic Restart

5.63 Automatic Restart Routine

When power is restored +5 volts goes high, the Control Panel ROMs are addressed, and the Control Panel to CPU Handshaking Routine is executed. This sets up the start address of the Automatic Restart ARF microinstruction routine which in turn sets the PFF flip-flop at CPU. After execution of this routine, system operation can continue from where it was interrupted when power failure occurred.

5.64 REAL TIME CLOCK (RTC)

This is a 1μS pulse initiated by the Power Supply every 20mS and input to the Inter Panel to generate signal RTCFZ1N. RTCFZ1N sets the RTC F/F at the CPU and the output of this F/F is input to the Interrupt Lines. The Interrupt Received Routine is executed.



5.65 INSTRUCTION HANDLING

When a software instruction appears on the I/O Bus it defines to the CPU that a certain action must be taken. This action is defined in the instruction format. The CPU must also decide from this instruction whether or not alternative action is necessary. For example, if an instruction is entered which is not valid then CPU ignores the contents of the instruction and takes alternative action. The following description splits the P851 set of instructions into appropriate groups and gives a brief description of how the CPU reacts to each group of instructions.

5.66 Load Instructions

To perform an operation on the contents of a Memory location or a register its contents must first be placed in one of Spalu's Scratch Pad registers. The register to be used is specified by the instruction format. There are three load instructions LDR, LD, LDK.

- LDR enables a 16-bit word to be loaded from any register into the specified register.
- LD enables a 16-bit word to be loaded from anywhere in Memory to the specified register.
- LDK loads a constant into the specified register.

5.67 Store Instructions

These instruct CPU to store the contents of a register. Storage is either in a memory location or register specified by the instruction format.

5.68 Arithmetic Instructions

These instruct CPU to perform add or subtract operations using Spalu registers. The instruction format contains an operand and also specifies the Spalu register whose contents are needed for the arithmetic operation.

5.69 Logical Instructions

Logical instructions tell CPU to perform AND or OR operations. The instruction format may specify a memory location or a register for the first operand. The second operand is specified as one of Spalu's registers. The two operands are ANDed or ORed by CPU.

5.70 Character Handling Instructions

These tell CPU to either exchange or compare the contents of a specified register. In this way an 8-bit character can be placed in the Spalu register specified by the instruction format.

5.71 Branch Instructions

Branch instructions cause the contents of the Program Counter to be updated and go to the address of another routine. This is achieved by Planet using its Condition Register to compare the Branch Instruction with the condition of the previous instruction. If certain conditions are fulfilled then CPU causes a Branch.

5.72 Shift Instructions

These enable CPU to rotate the contents of one of Spalu's registers A1-A7. The instruction format specifies the register and the type of shift operation.

5.73 Control Instructions

These instructions control CPU as follows.

- ENB sets the Enable flip-flop in Planet which enables interrupts.
- INH resets the Enable flip-flop which inhibits interrupts.
- RIT resets one of the four internal interrupt flip-flops of CPU.
- HLT resets the Enable flip-flop in Planet and CPU goes to the Idle Micro-instruction Routine.
- LKM sets the CPU's Program Interrupt flip-flop which changes the CPU operating mode from User to System.

Note that INH and ENB instructions would only normally be used in Stand Alone programs. A breakdown of these instructions is given under the heading Control Instructions -- Detailed Description.

5.74 Input Output Instructions

These instructions control all data transfers between CPU and peripherals. For input output routines, control instruction LKM links to Monitor and therefore all data transfers are under Monitor control.

5.75 P851 List of Instructions

Table 5-3 groups the P851 instruction set and indicates possible CPU action that could result after the execution of a software instruction. Meanings of the abbreviations are as follows.

- Number 1 indicates that CPU may be in System or User Mode. If A15 or T3B is shown this indicates that if register A15 is addressed or if the instruction format is T3B type then that instruction must be executed in System Mode. Any attempt to execute one of these instructions in User Mode will cause a Trap.
- Letter T is used to indicate that a Trap Condition may occur due to certain conditions. These conditions are different for each instruction and are defined in the Software Manual, instruction set.
- The letters NC are used to indicate that when the instruction is executed the Condition Register remains unchanged.
- The symbol * indicates that the instruction causes a microinstruction routine to be addressed.

For more detailed information of the instruction set (e.g. execution times) refer to the Software Manuals.

Table 5-3 P851 List of Instructions

		System	User	Trap	Condition Register	Micro Routine
LOAD AND STORE						
LDK	Load Constant	A15				*
LDR	Load Register / Register	A15	T3B			*
LD	Load Register	A15				*
STR	Store Register / Register	A15	T3B		NC	
ST	Store Register	A15			NC	*
MLK	Multiple Load Constant	A15				*
MLR	Multiple Load / Register	A15	T3B			
ML	Multiple Load	A15				*
MSR	Multiple Store / Register	A15	T3B		NC	
MS	Multiple Store	A15			NC	*
ARITHMETIC						
ADK	Addition Constant	A15				*
ADR	Addition Register / Register	A15				*
AD	Addition	A15				*
SUK	Subtract Constant	A15				*
SUR	Subtract Register / Register	A15				*
SU	Subtract	A15				*
MUK	Multiply with constant	1	1			
MUR	Multiply Register / Register	1	1			*
MU	Multiply	1	1			*
DVK	Divide with Constant	1	1			
DVR	Divide Register / Register	1	1			*
DV	Divide	1	1			*
DAK	Double Add with Constant	1	1			*
DAR	Double Add Register / Register	1	1			*
DA	Double Add	1	1			*
DSK	Double Subtract with Constant	1	1			*
DSR	Double Subtract Register / Register	1	1			*
DS	Double Subtract	1	1			*
C2R	Two's Complement	1	1			
C2	Two's Complement	1	1			*
IMR	Increment Memory / Register	1	1			
IM	Increment Memory	1	1			*
NGR	Negate Register	1	1			*
LOGIC						
ANK	Logical AND with Constant	A15		T		*
ANR	Logical AND Register / Register	A15		T		
AN	Logical AND	A15		T		*
ORK	Logical OR with Constant	A15				*
ORR	Logical OR Register / Register	A15				*
OR	Logical OR	A15		T		*
XRK	Exclusive OR with Constant	A15		T		*
XRR	Exclusive OR Register / Register	A15				*
XR	Exclusive OR	A15				*
TM	Test Mask	A15				*
TNM	Test Not Mask	A15				*
CWK	Compare Word with Constant	A15				
CWR	Compare Word Register / Register	A15				*
CW	Compare Word	A15				*
C1R	One's Complement Register / Register	A15				*
C1	One's Complement	A15		T		*
CMR	Clear Memory / Register				NC	
CM	Clear Memory				NC	*

Table 5-3 contd.

		System	User	Trap	Condition Register	Micro Routine
CHARACTER HANDLING						
LCK	Load Character	A15		T	NC	
LCR	Load Character / Register	A15		T	NC	
LC	Load Character	A15		T	NC	*
SCR	Store Character / Register	A15		T	NC	
SC	Store Character	A15		T	NC	*
CCK	Compare Character	A15		T		
CCR	Compare Character / Register	A15		T		
CC	Compare Character	A15		T		*
ECR	Exchange Character Register / Register	A15		T	NC	*
BRANCH						
AB	Absolute Conditional Branch	1	1		NC	*
ABR	Absolute Conditional Branch to Register	1	1		NC	*
ABI	Absolute Branch	1	1		NC	
RF	Relative Forward Conditional Branch	1	1		NC	*
RB	Relative Backward Conditional Branch	1	1		NC	*
CF	Call Function	A15			NC	*
CFR	Call Function / Register	A15			NC	*
CFI	Call Function	A15			NC	
RTN	Return	A15				*
EXK	Execute Constant	1	1	T	NC	
EXR	Execute / Register	1	1		NC	*
EX	Execute	1	1		NC	*
SHIFT						
SLA	Single left arithmetic shift	1	1			*
SRA	Single right arithmetic shift	1	1			*
SLL	Single left logical shift	1	1			*
SRL	Single right logical shift	1	1			*
SLC	Single left circular shift	1	1			*
SRC	Single right circular shift	1	1			*
SLN	Single left and normalize shift	A15			NC	*
SRN	Single right and normalize shift	A15			NC	*
DLA	Double left arithmetic shift	1	1			*
DRA	Double right arithmetic shift	1	1			*
DLL	Double left logical shift	1	1			*
DRL	Double right logical shift	1	1			*
DLC	Double left circular shift	1	1			*
DRC	Double right circular shift	1	1			*
DLN	Double left and normalize shift	A15			NC	*
DRN	Double right and normalize shift	A15			NC	*
CONTROL						
ENB	Enable interrupt	1	1		NC	*
HLT	Halt	1			NC	*
RIT	Reset internal interrupt	1			NC	*
INH	Inhibit interrupt	1			NC	*
LKM	Link to monitor	1	1		NC	*
INPUT/OUTPUT						
CIO	Control I/O	1				*
INR	Input to Register	1		T		*
SST	Sense Status	1				*
OTR	Output from Register	1		T		*
TST	Test Status	1		T		*
WER	Write External Register	1		T	NC	*
RER	Read External Register	1		T		*

5.76 CONTROL INSTRUCTIONS -- DETAILED DESCRIPTION

CPU reads all Control Instructions via the I/O Bus. The following paragraphs show how the instruction affects control over CPU registers.

5.77 Reset Internal Interrupt

There are four internal interrupts each one being latched in a 74279 flip-flop at the CPU. This sets an interrupt code on the IS lines which remains until the appropriate Interrupt flip-flop is reset by executing an RIT software instruction. When the RIT appears on the I/O Bus one of the following four bits will be set to zero.

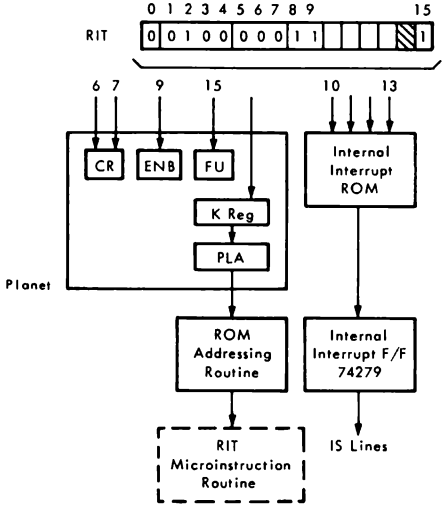
Bit 10 resets CP flip-flop. Control Panel Interrupt

Bit 11 resets PFF flip-flop. Power Failure / Automatic Restart Interrupt

Bit 12 resets RTC flip-flop. Real Time Clock Interrupt

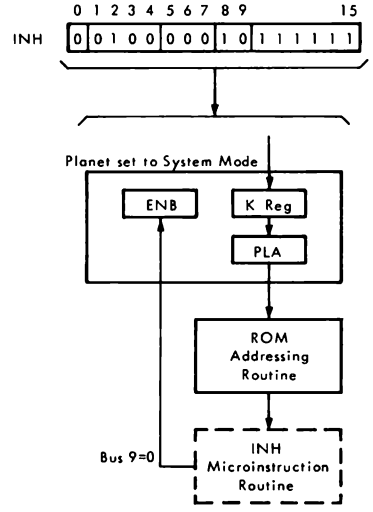
Bit 13 resets PI flip-flop. Program Interrupt

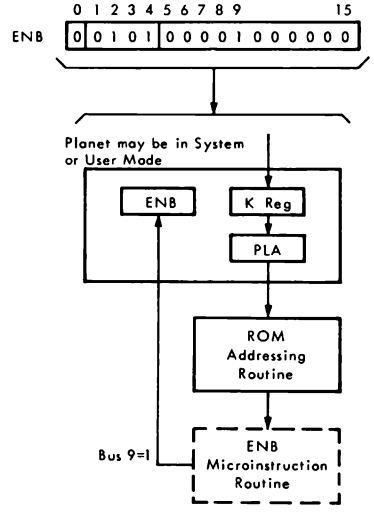
Note that the other three bits will be set to 1. Planet also reads the RIT from the I/O Bus to set up the address for the RIT microinstruction routine.



5.78 Enable and Inhibit Interrupts

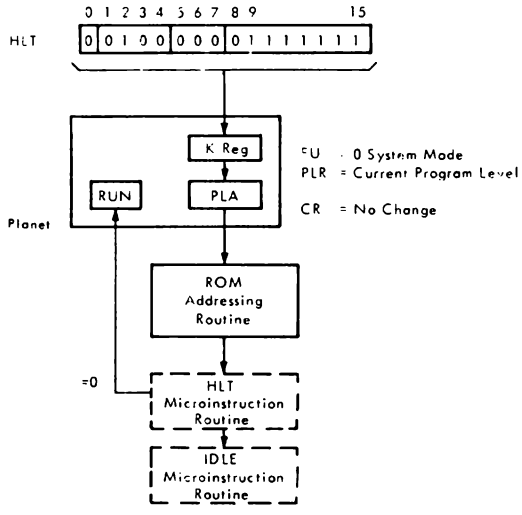
These two instructions will normally only be used in Stand Alone programs. Software instruction INH will inhibit interrupts and at the end of the routine ENB is executed to enable interrupts.

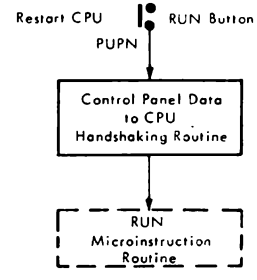




5.79 Halt CPU

CPU is in System Mode. Execution of this instruction sets up the PLA code to address the start of the HLT microinstruction routine and reset the RUN F/F at Planet. Note that the conditions of Program Status Word and Program Counter do not change. To enable a return, the RUN button is depressed which addresses the start of the RUN microinstruction routine.





5.80 MICROINSTRUCTIONS

Microinstructions are programmed in 10x4 bit Read Only Memories. They contain 1,023 40 bit words which control CPU functions such as Fetch, Multiply, Divide and Diagnostic Routines. These routines appear as Flow Charts at the end of this section.

5.81 ROM Addressing Routine (Figures 5-20 and 5-21)

The microinstruction address is selected from either the NA or PLA lines and is input to the ROM Address Register. The ROMs are addressed in two fields, bits 0-23 and 24-39. When both fields are to be addressed the ROM Address Register is clocked at CP time to enable bits 0-23 and at BP time to clock the RADA Address Register which enables bits 24-39.

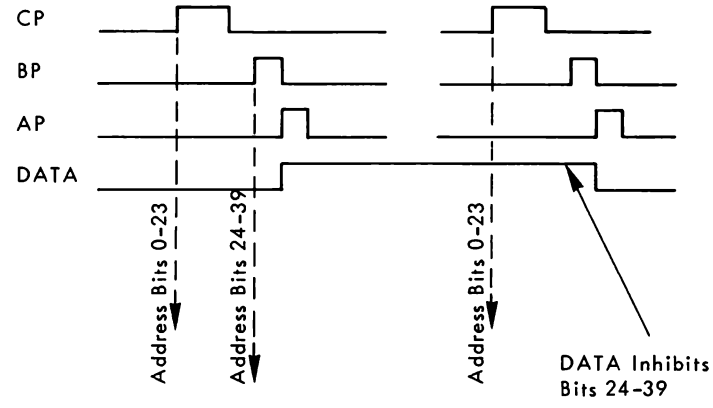


Figure 5-20 Timing for ROM Addressing

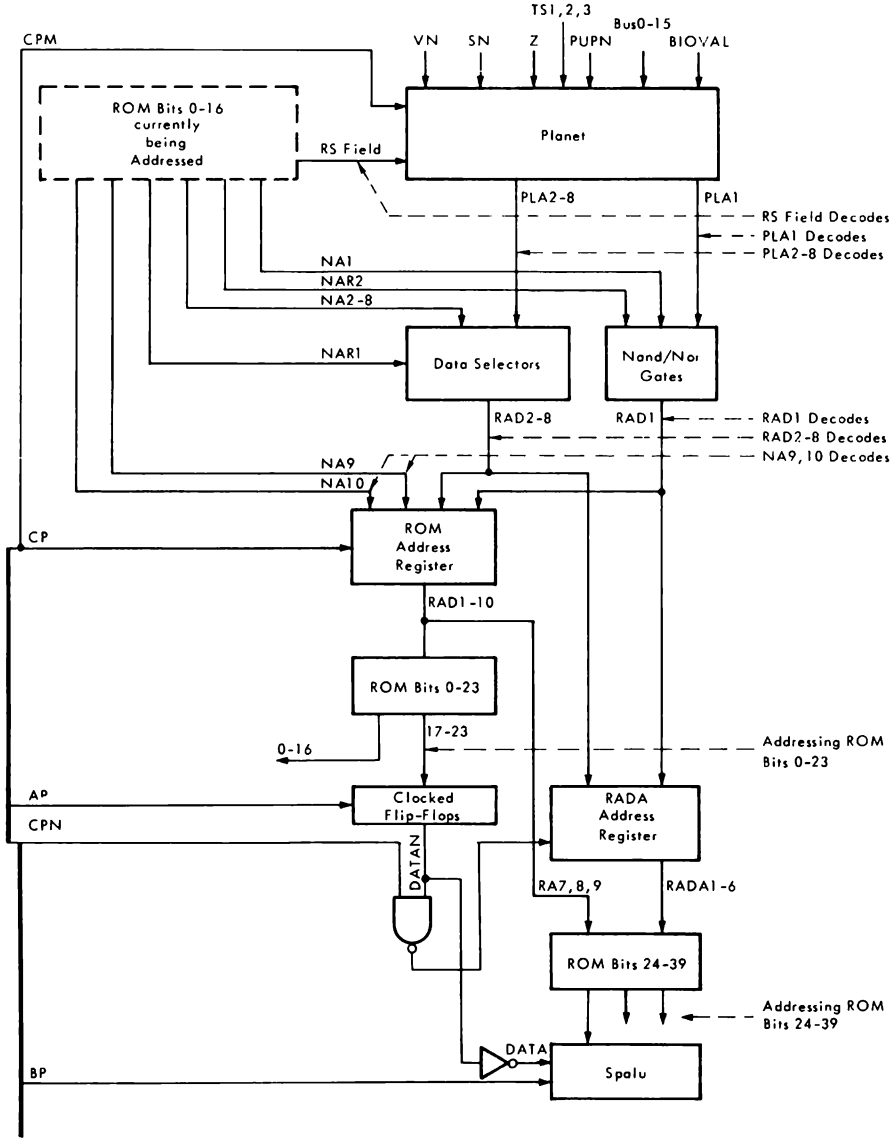


Figure 5-21 ROM Addressing Routine

5.82 RS Field Decodes

The RS Field is decoded by Planet. A description of the results is given in Table 5-4 in an abbreviated form.

5.83 PLA1 Decodes

The following table shows the way that the PLA1 values are selected at Planet

ROM Bits

36 37 38

TS1	TS2	TS3	PLA1	Meaning
0	0	0	F1	Output from PLA2-8 Decoder
0	0	1	K15	Output from K
0	1	0	CEZ	K right null
0	1	1	COK	Branch enable
1	0	0	S	Sign of ALU
1	0	1	R1Z	R1 Field null
1	1	0	Z	Zero of ALU
1	1	1	V	Overflow

5.84 PLA2-8 Decodes

The following list shows the way that the PLA2-8 values are selected at Planet.

MD0 Output from Mode F/F0

MD1 Output from Mode F/F1

K0 Output from K Register

K1 Output from K Register

K2 Output from K Register

K3 Output from K Register

K4 Output from K Register

R1Z Decoded from K0,5,6,7,8 and means decimal 0 in the R1 field

R1Q Decoded from K5,6,7,8 and means decimal 15 in the R1 field

K8FU Decoded from K8 or the USER F/F depending on the Mode

Table 5-4 RS Field Decodes

RS Field Decimal Value	RS Field Bits 0 1 2 3 4	MCKCRN MXLN MKRN MPLRN	MGFN MRUNN MIRN PLOK	MON MIN M2N M3N	Description of results	Result to I/O Bus when BIOVAL=0		MNEMONIC
0	0 0 0 0 0	1 1 1 1	1 0 1 0	0 1 1 1	RUN <= 1			RUNZ1
1	0 0 0 0 1	1 1 1 1	1 0 1 0	1 1 1 1	RUN <= 0	Log		RUNZO
2	0 0 0 1 0	1 0 1 1	1 1 1 0	1 1 1 1	KL <= BIO	Log		BIØKL
3	0 0 0 1 1	1 1 1 1	1 1 1 0	0 1 1 1	NOP	CMP	$(2N,\overline{n})+N2$	R2R1 or CMP
4	0 0 1 0 0	1 1 1 1	1 1 1 0	1 0 1 1	NOP	Log	R2R2P1	R2P1R2
5	0 0 1 0 1	1 1 1 1	1 1 1 0	1 1 0 1	NOP	AR	-PSW	PSWBUS or AR
6	0 0 1 1 0	1 1 1 1	1 1 1 0	1 1 1 0	NOP	Log	INTAD	INTADBUS or LOG
7	0 0 1 1 1	1 1 0 1	1 1 1 0	1 0 0 1	KR <= BIO	AR		BUSKR
8	0 1 0 0 0	1 1 0 1	1 1 1 0	0 1 0 1	KR <= N	CMP		NKR
9	0 1 0 0 1	1 1 0 1	1 1 1 0	0 0 1 1	KR <= CB	CMP		CBKR
10	0 1 0 1 0	1 1 1 0	1 1 1 0	1 1 1 1	PL <= BIEC	Log		BIECPLR
11	0 1 0 1 1	0 1 1 0	0 1 1 0	0 1 0 0	PL <= BIO, GF = BIO, CR = BIO	CMP		BUSPSW
12	0 1 1 0 0	1 1 1 1	0 1 1 0	1 1 1 1	GF <= BIO	Log		BIØGF
13	0 1 1 0 1	0 1 1 1	1 1 1 0	0 0 0 1	CR <= CRDA	CMP		SETCR
14	0 1 1 1 0	0 1 1 1	1 1 1 0	0 1 0 0	CR <= BIO	CMP		BIØCR
15	0 1 1 1 1	0 1 1 1	1 1 1 0	0 1 0 1	CR <= TRU	CMP		CR3
16	1 0 0 0 0	1 0 0 1	1 1 1 1	1 0 0 1	MD <= [0,0] , KL KR<=BIO	AR		BIØKPKOO
17	1 0 0 0 1	1 1 1 1	1 1 1 1	1 0 1 1	MD <= [0,1]	Log		PKO1
18	1 0 0 1 0	1 1 1 1	1 1 0 1	1 1 0 1	MD <= [1,0] ,IR<=IRD	AR		PK10IR
19	1 0 0 1 1	1 1 1 1	1 1 1 1	1 1 1 1	MD <= [1,1]	Log	ZERO	PK11
20	1 0 1 0 0	1 1 1 1	1 0 0 0	1 1 1 1	RUN <= 0 ,IR <=IRD	Log		RUNZOIRD
21	1 0 1 0 1	1 1 1 1	1 0 0 0	0 1 1 1	RUN <= 1 ,IR <=IRD			RUNZ1IRD
22	1 0 1 1 0	1 1 0 1	1 1 1 0	0 0 0 1	KR <= TRU	CMP		TRUKR
23	1 0 1 1 1	0 0 0 0	0 0 1 1	1 0 0 1	KL&KR <=BIO, PL <=(BIEC), GF<=BIO,RUN<=0,CR<=CRD	AR		RUNOBIØK
24	1 1 0 0 0	1 1 1 1	1 1 1 1	1 0 0 1	MD <= [0,0]	AR		PSWR2P1
25	1 1 0 0 1	1 1 1 0	1 1 1 0	0 1 0 0	PL <= BIO	CMP		BUSPLR
26	1 1 0 1 0	0 1 1 1	1 1 0 0	0 0 0 1	CR <= CRD, IR <=IRD	CMP		SETIR
27	1 1 0 1 1	1 1 1 0	0 0 1 1	1 0 1 1	MD <= [0,1],PL<=BIEC,GF<=BIO,RUN<=0,I	Log		BIECPL
28	1 1 1 0 0	1 1 1 1	1 1 1 1	1 1 0 1	MD <= [1,0]			PK10AR
29	1 1 1 0 1	1 0 0 0	0 1 1 0	1 0 0 0	PL <= BIO, GF <=BIO , K<=BIO	AR		BIØKPG
30	1 1 1 1 0	1 1 1 1	1 1 1 0	0 0 0 0	NOP			CMP3
31	1 1 1 1 1	1 1 1 1	1 1 1 0	1 1 0 0	NOP	AR		AR3

K9IR Decoded from K9 or the interrupt F/F depending on the Mode.

K10PUP Decoded from K10 or PUP from the Control Panel depending on the Mode.

R2Z Decoded from K11,12,13,14 and means decimal 0 in the R2 field.

R2Q Decoded from K11,12,13,14 and means decimal 15 in the R2 field.

K15RUN Decoded from K15 or RUN F/F depending on the Mode.

Note that where reference is made to the K Register, e.g. K8, this represents Bus 8. Also R1,R2 fields refer to the instruction field.

5.85 RAD1 Decodes

The most significant bit of the address code RAD1 is selected as follows.

PLA1	NA1	NAR2	RAD1
1	1	1	1
1	0	1	0
1	1	0	1
1	0	0	1
0	1	1	1
0	0	1	0
0	1	0	0
0	0	0	0

5.86 RAD2-8 Decodes

The value of RAD2-8 is taken from either PLA2-8 or NA2-8 depending on the value of NAR1.

NAR1	RAD8	RAD7	RAD6	RAD5	RAD4	RAD3	RAD2
0	PLA8	PLA7	PLA6	PLA5	PLA4	PLA3	PLA2
1	NA8	NA7	NA6	NA5	NA4	NA3	NA2

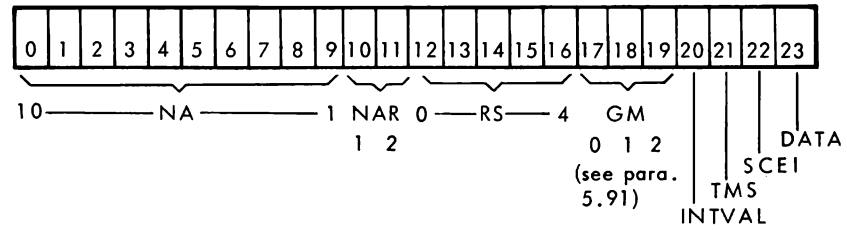
5.87 NA9,10 Decodes

At the end of a microinstruction routine PLA1,2 or 3 is enabled by NA9 or NA10 to select the next address.

NA 09	NA 10	PLA
1	0	1
0	1	2
1	1	3

5.88 Addressing ROM Bits 0-23

At time CP, ROM bits 0-23 are addressed.



NA, 10 bits -- used as next address for explicite addressing.

NAR, 2 bits -- used by the Address Selectors to select the next address.

RS, 5 bits -- decoded inside Planet to generate its own internal control signals.

GM, 3 bits -- decoded to enable five CPU control signals.

INTVAL, 1 bit -- validates an interrupt code for input to Planet.

SCEI, 1 bit -- inhibits interrupts during certain microinstruction routines.

TMS, 1 bit -- enables one of the interface signals TMRN, TMEN, TMPN.

DATA, 1 bit -- enables the RCONT field into Spalu, TEST field into Planet.

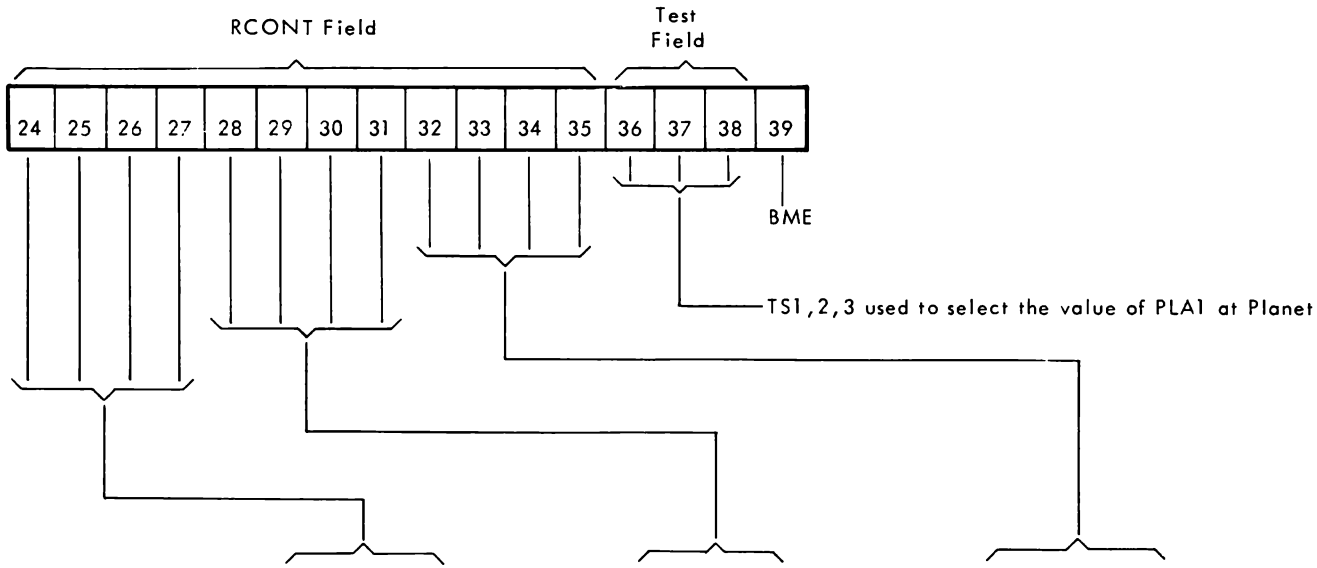
5.89 Addressing ROM Bits 24-39

If signal DATA = 0 then at the next CPN clock pulse ROM bits 24-39 will be addressed.

RCONT, 12 bits -- data for input to Spalu to control arithmetic operations.

TEST, 3 bits -- input to Planet for setting up the PLA1 output.

BME, 1 bit -- enables the Bus Input/Output Lines.



	ALU SOURCE SELECTION								OPERATION ON SELECTED SOURCE					DESTINATION FIELD (Various Load Operations available to the User)									
	C00 C01 C02 C03 inverted								C04 C05 C06 C07 inverted					C08 C09 C10 C11 inverted				ALU	Bus Contents			DATA	RX	A
0	0	0	0	0	SPA	SPA	SPB	SPB	0	0	0	0	XA + XB	0	0	0	0	RX				0		AX
1	0	0	0	1	SPA	SPA	A	A	0	0	0	1	XA + XB	0	0	0	1	RX	A			1		
2	0	0	1	0	SPA	SPA	0	0	0	1	0	1	XA + XB	0	0	1	0	RX	A	R1	R2	1		
3	0	0	1	1	SPA	SPA	0	A	0	0	1	0	XA plus XB	0	0	1	1	RX				0		
4	0	1	0	0	P	P	SPB	SPB	0	0	1	1	XA minus XB	0	1	0	0	AX				0		
5	0	1	0	1	P	P	A	A	0	1	0	0	XA . XB	0	1	0	1	AX	A			1		
6	0	1	1	0	P	P	0	0	0	1	1	0	XA ⊕ XB	0	1	1	0	AX				0	SP	
7	0	1	1	1	P	P	0	A	1	0	0	0	XA plus XB plus 1	0	1	1	1			R1	R2	1		
8	1	0	0	0	0	0	SPB	SPB	1	0	1	1	XB minus 2 (0 on XA)	1	0	1	1					0		AX
9	1	0	0	1	0	0	A	A	1	1	0	0	XB plus 2 (0 on XA)	1	1	0	0		A			1		
A	1	0	1	0	0	0	AX	AX	1	1	0	1	XA plus 2 (0 on XB)	1	0	1	0					0	SP	
B	1	0	1	1	0	0	0	A	1	1	1	0	XA minus 2 (0 on XB)	1	0	1	1			R1	R2	1	BUS	
C	1	1	0	0	A	A	SPB	0	1	1	1	1	$\overline{XA}$ plus XB	1	1	0	0					0	P	
D	1	1	0	1	0	SPA	0	A						1	1	0	1		A			1	BUS	
E	1	1	1	0	SPA	0	0	0	1	0	0	1	Shift left	1	1	1	0					1		
F	1	1	1	1	SPA	0	0	A	1	0	1	0	Shift right	1	1	1	1					1	BUS	

Note that when Register C bits 8,9,10 and 11 are 1001, then this specifies a shift. For a description of these shifts see paragraph heading, Spalu.

Figure 5-22 Addressing ROM Bits 24-39

5.90 Addressing Modes

The two high order bits NA1 and NA2 are always fetched. The 8 low order bits of the next address depend on the mode i.e. the value of bits NAR1, NAR2.

NAR1	NAR2	MODE	Content of RA8	RA7	RA6	RA5	RA4	RA3	RA2	RA1
0	0	PLA	PLA8	PLA7	PLA6	PLA5	PLA4	PLA3	PLA2	PLA1
1	0	TEST	NA8	NA7	NA6	NA5	NA4	NA3	NA2	PLA1
1	1	Explicit	NA8	NA7	NA6	NA5	NA4	NA3	NA2	NA1

5.91 GM Decodes

The GM decodes are as follows.

GM 2	1	0	Signals
0	0	0	—
0	0	1	CGFN
0	1	0	RUNZON
0	1	1	BIOVALN
1	0	0	PUPSN
1	0	1	BIOEKEYN
1	1	0	IPSN
1	1	1	—

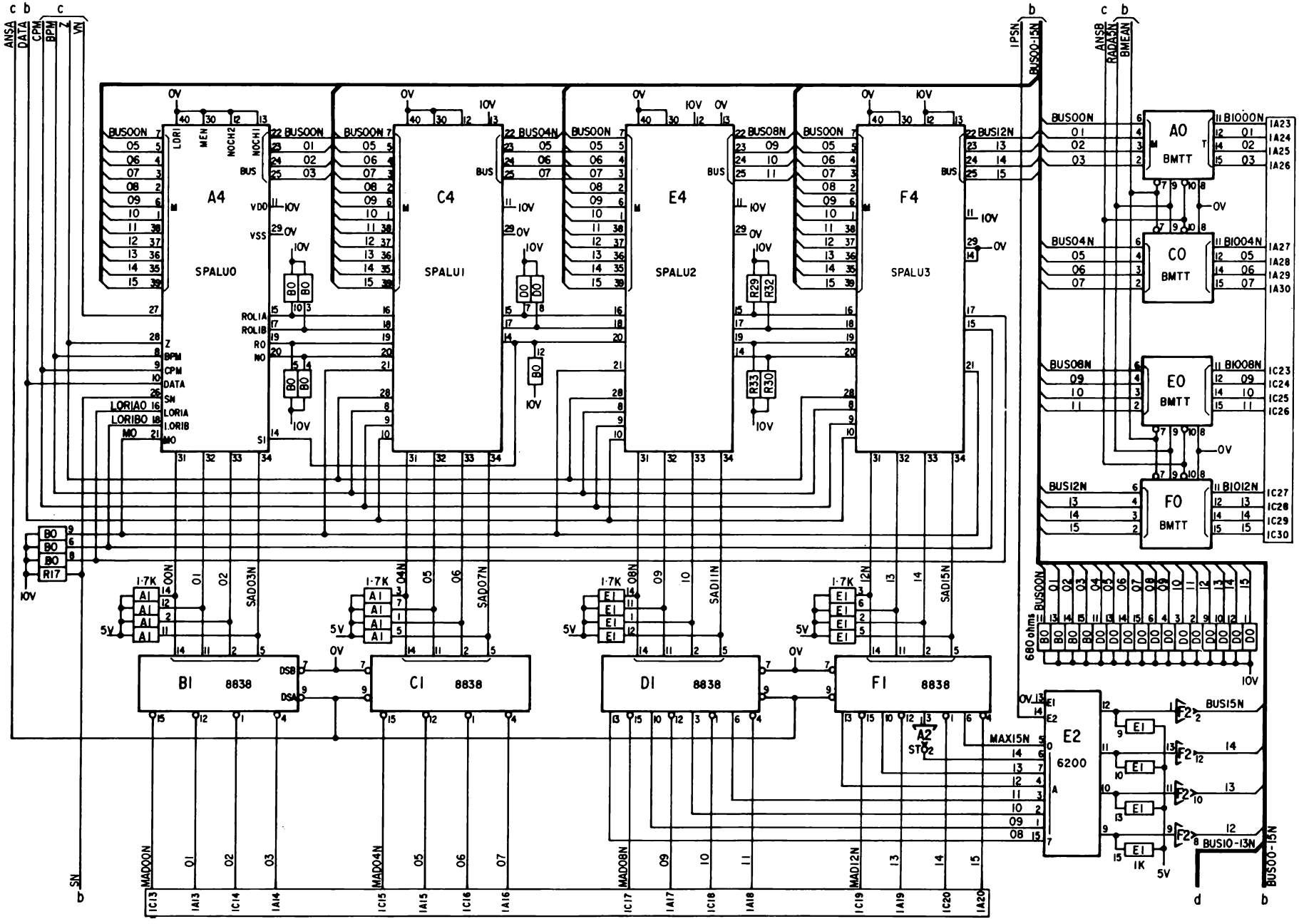


Figure 5-23a SPALU, Data In/Out, IPL Logic Diagram

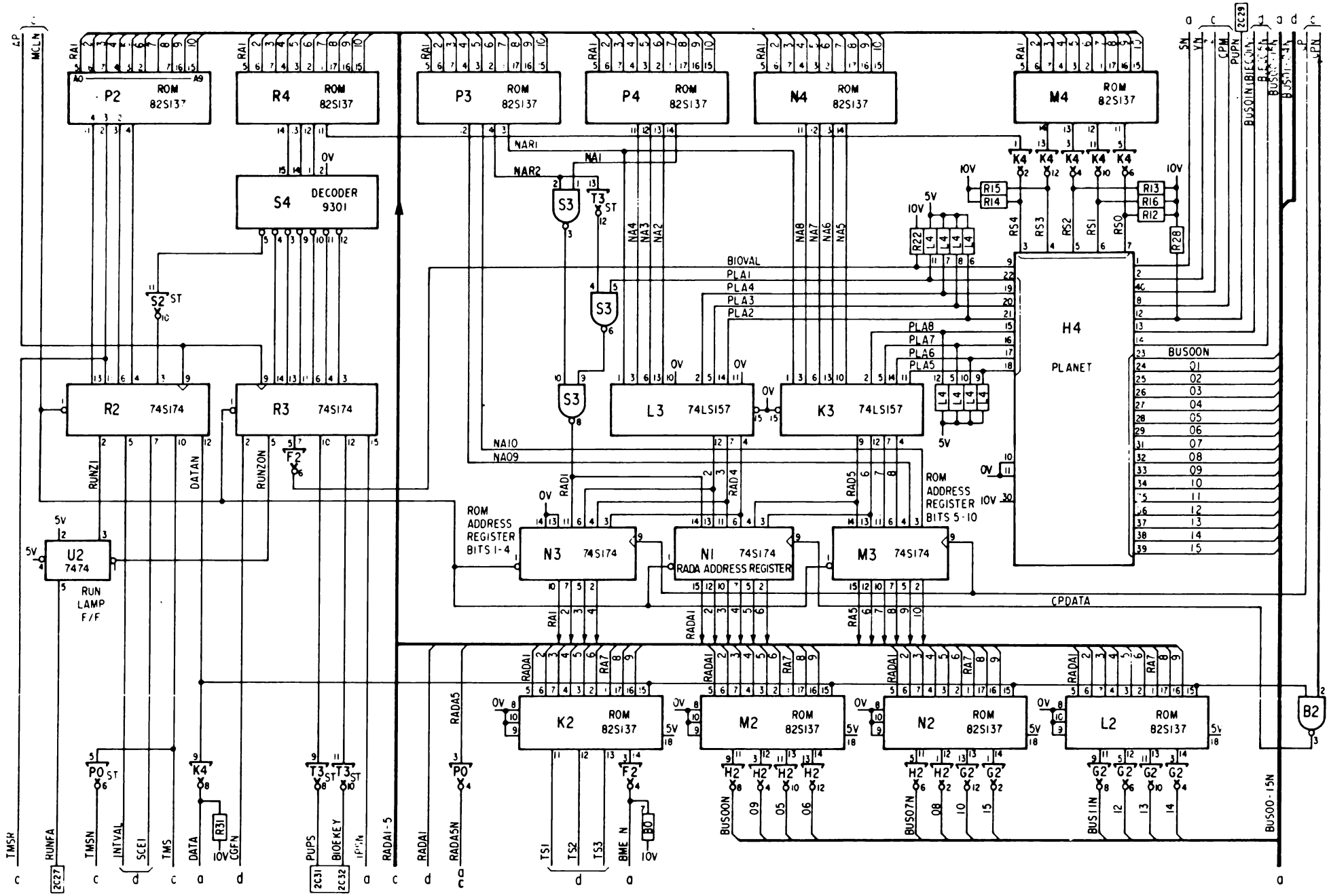


Figure 5-23b Planet and Rom Control Logic Diagram

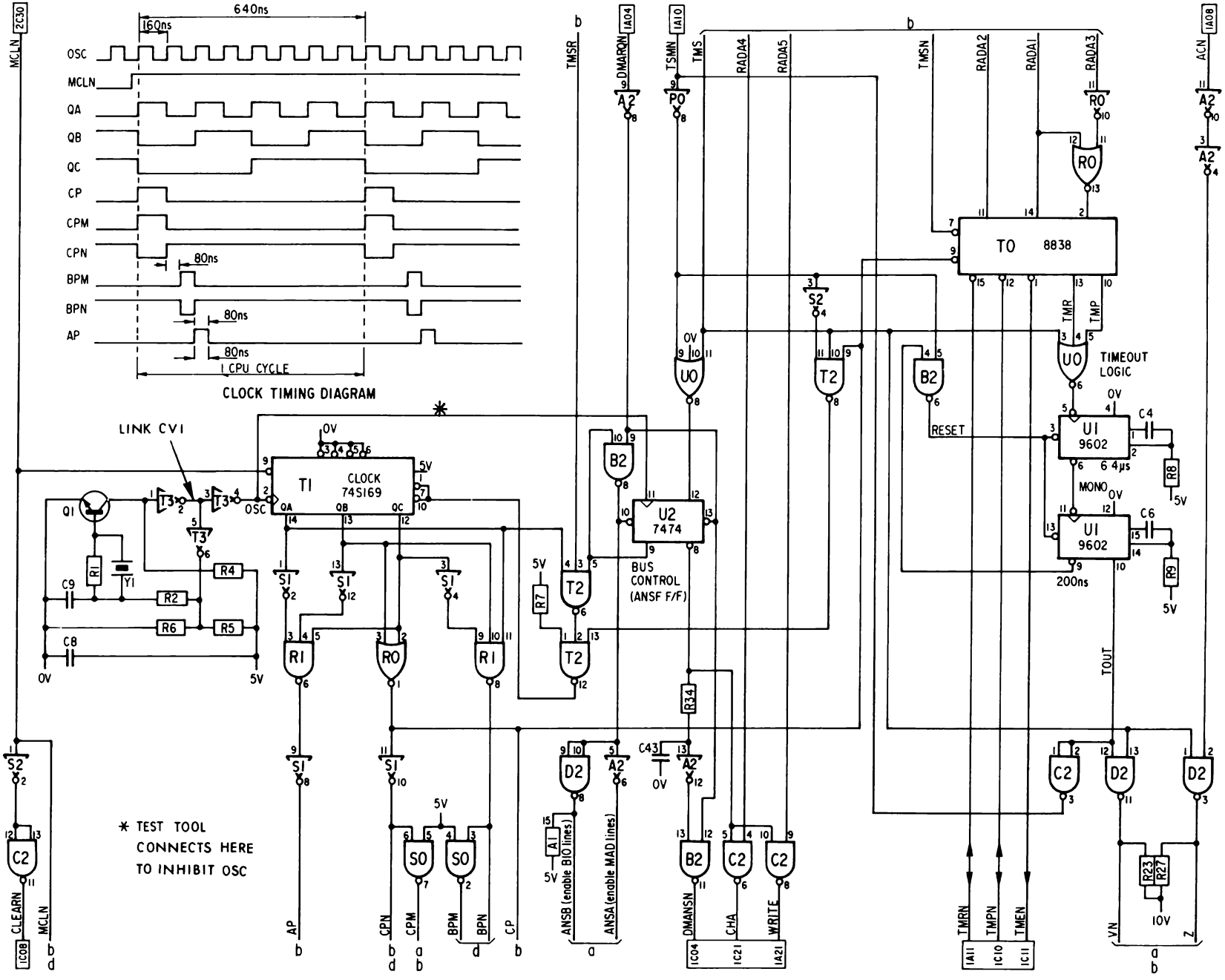


Figure 5-23c P851 Clock, Bus Control, Timeout Logic Diagram

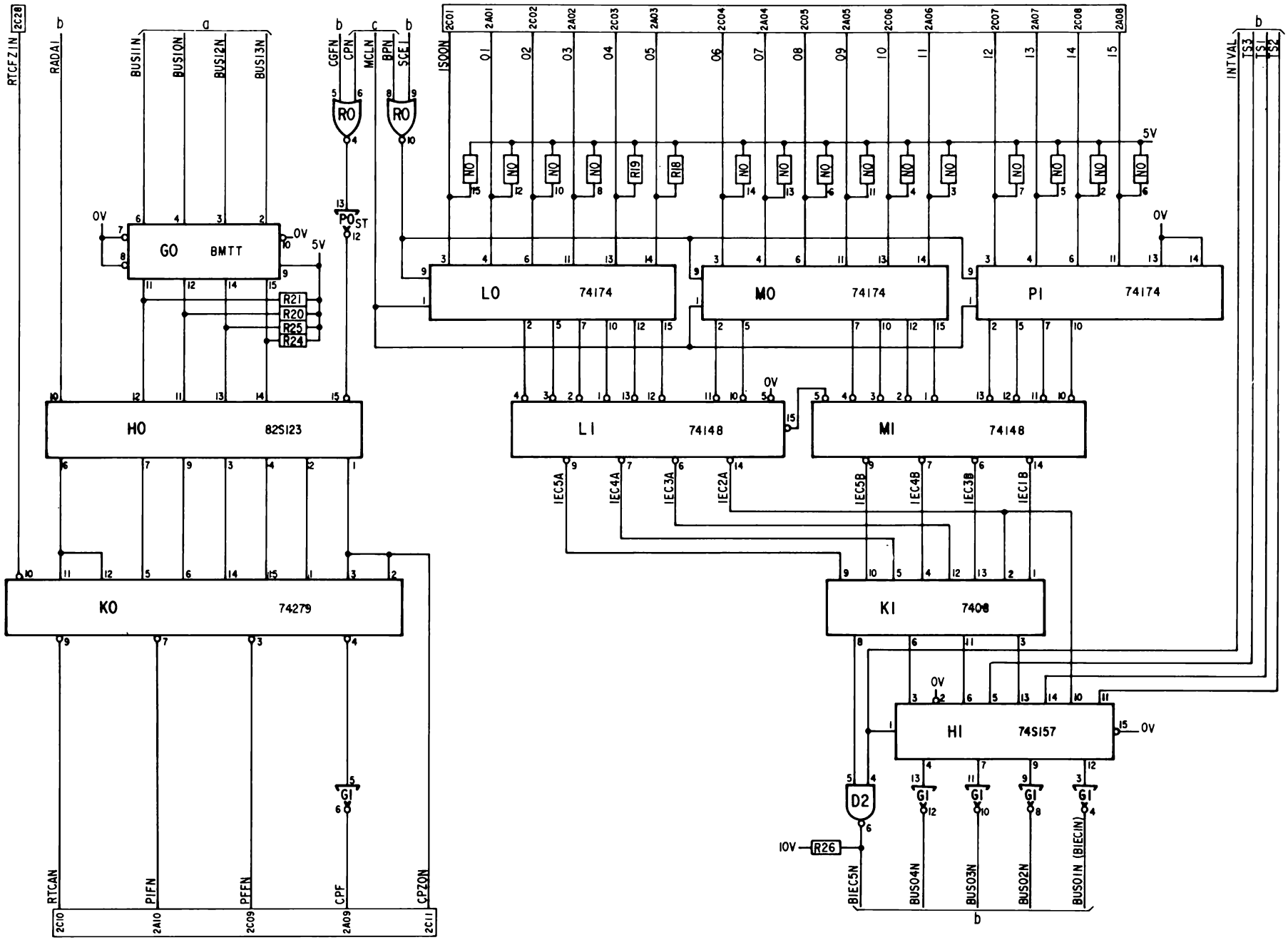


Figure 5-23d Interrupt Logic Diagram

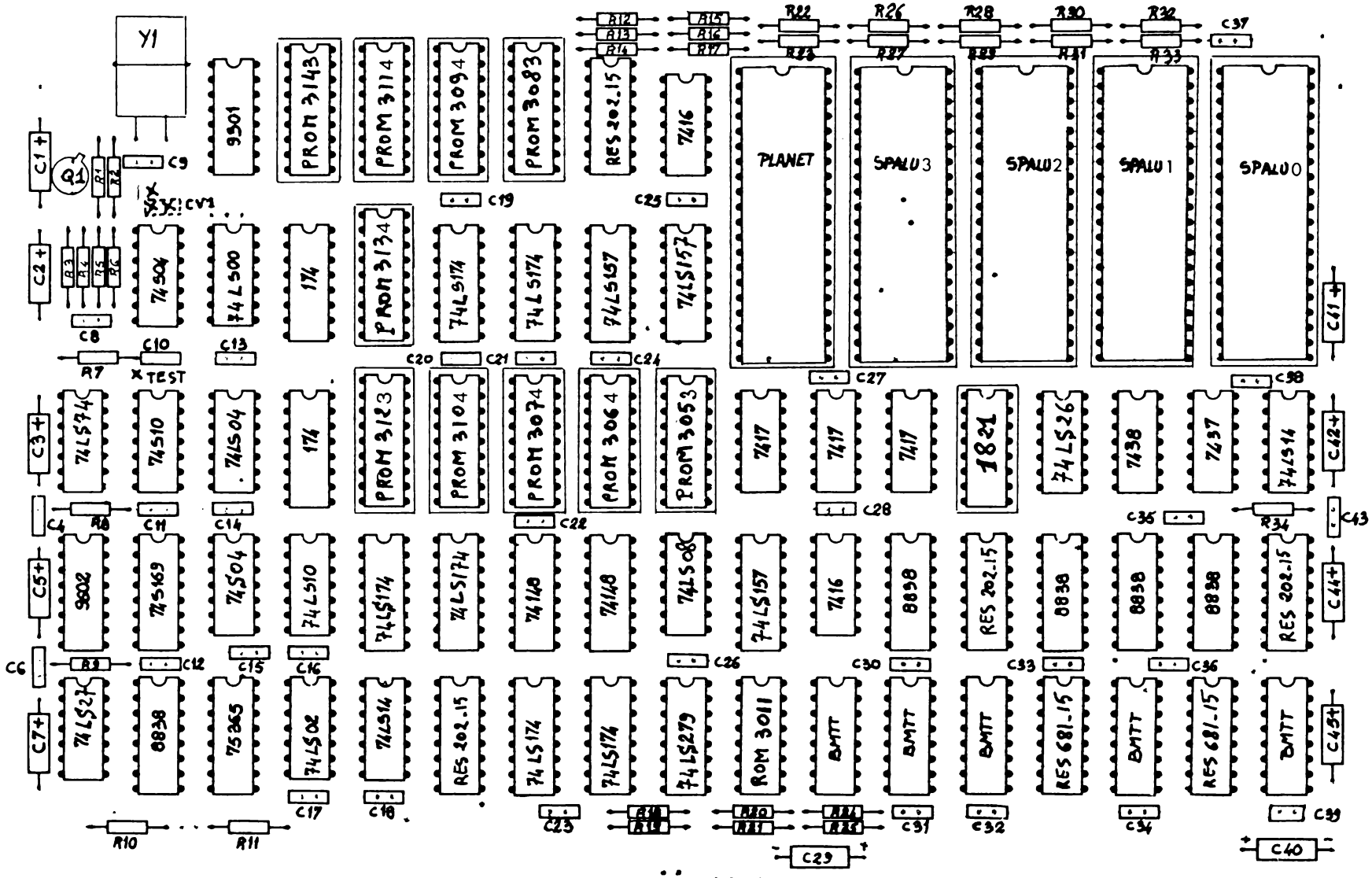


Figure 5-24 P851 CPU Component Layout

Table 5-5 P851 CPU Parts List

Reference	Description	12NC Code
	Printed circuit	5111 100 06172
K2.	Integrated circuit PROM 3053.	
L2.	Integrated circuit PROM 3064.	
M2.	Integrated circuit PROM 3074.	
M4.	Integrated circuit PROM 3083.	
N4.	Integrated circuit PROM 3094.	
N2.	Integrated circuit PROM 3104.	
P4.	Integrated circuit PROM 3114.	
P2.	Integrated circuit PROM 3123.	
P3.	Integrated circuit PROM 3134.	
R4.	Integrated circuit PROM 3143.	
K1.	Integrated circuit 74LS08.	
D4, G1.	Integrated circuit 7416.	
F2, G2, H2.	Integrated circuit 7417.	
D2.	Integrated circuit 74LS26.	
U0.	Integrated circuit 74LS27.	
C2.	Integrated circuit 7438.	
U2.	Integrated circuit 74LS74.	
L1, M1.	Integrated circuit 74148.	
H1, K3, L3.	Integrated circuit 74LS157.	
K0.	Integrated circuit 74LS279.	
A2, P0.	Integrated circuit 74LS14.	
S3.	Integrated circuit 74LS00.	
R0.	Integrated circuit 74LS02.	
S2.	Integrated circuit 74LS04.	
T2.	Integrated circuit 74S10.	
B2.	Integrated circuit 7437.	
T1.	Integrated circuit SN74S169J.	
L0, M0, M3, N1, N3, P1, R2, R3.	Integrated circuit 74LS174.	
S0.	Integrated circuit 75365.	
H0.	Integrated circuit ROM 3011.	
B1, C1, D1, F1, T0.	Integrated circuit 8838.	
S4.	Integrated circuit 9301.	
U1.	Integrated circuit 9602.	
A0, C0, E0, F0, G0.	Integrated circuit BMTT.	
H4.	Integrated circuit PLANET.	
A4, C4, E4, F4.	Integrated circuit SPALU.	
S1, T3.	Integrated circuit 74S04.	
R1.	Integrated circuit 74LS10.	

Table 5-5 contd.

Reference	Description	12NC Code
R12-16, 31, 20, 21, 24, 25.	Resistor 330Ω, 1/4W, 5%.	
R17, 22, 23, 26-30, 32, 33.	Resistor 680Ω, 1/4W, 5%.	
R3, 7, 10, 11, 18, 19.	Resistor 1KΩ, 1/4W, 5%.	
R5.	Resistor 464Ω, 1/8W, 1%.	
R34.	Resistor 511Ω, 1/8W, 1%.	
R6.	Resistor 681Ω, 1/8W, 1%.	
R2.	Resistor 2.15KΩ, 1/8W, 1%.	
R1, 4.	Resistor 3.16KΩ, 1/8W, 1%.	
R8, 9.	Resistor 31.6KΩ, 1/8W, 1%.	
C9.	Capacitor 47pF, ±2%, ceramic.	
C6.	Capacitor 22pF, ±2%, ceramic.	
C43.	Capacitor 180pF, ±2%, ceramic.	
C4.	Capacitor 680pF, ±10%, ceramic.	
C8, 10-28, 30-39.	Capacitor 10nF, ±10%, ceramic.	
C1-3, 5, 7, 29, 40-42, 44, 45.	Capacitor 10µF, 25V, FITCO.	
Y1.	Quartz QA60 12.5MHZ.	
Q1.	Transistor BSX20.	
A1, E1, N0, L4.	Resistor Network 202-15.	
B0, D0.	Resistor Network 681-15.	
J1, J2.	Connectors 100 964 053.	

Table 5-6 ROM Contents

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
000	0	01010001100	11	11100	000	1000	0000	0000	0000	000	1
	1	00111101110	10	11001	000	1000	0000	0000	0000	000	1
	2	01000110100	11	11100	000	1000	0100	0011	1111	000	1
	3	00001010001	11	11100	000	0100	0000	0000	0000	000	1
	4	00111111100	10	11100	000	1000	0000	0000	0000	000	1
	5	00000011000	11	11100	000	1010	0000	0000	0000	000	1
	6	00000000000	00	11100	000	1000	0000	0000	0000	000	1
	7	10000011111	11	11100	000	0000	0000	0000	0000	000	1
	8	10001010001	11	01101	000	1011	1111	1111	1111	000	0
	9	00000011101	11	11100	000	1000	1111	1111	1111	000	0
00A	10	00111110110	11	10010	000	1000	0000	0000	0000	111	0
	11	10001110001	11	11100	000	0010	0000	0000	0000	000	1
	12	10000100000	11	01101	000	1011	1110	1111	1101	000	1
	13	10111010000	10	11100	000	1000	0000	0000	0000	000	1
	14	00001100001	11	00001	000	0100	1111	1111	0111	000	1
	15	10000000000	00	11010	000	1000	1011	1101	0000	000	1
010	16	10000000011	11	00101	000	1011	0000	0000	0000	000	1
	17	10000010001	11	10010	000	0110	0000	0000	0000	000	1
	18	10000000011	11	00101	000	1011	0000	0000	0000	000	1
	19	10000000011	11	01101	000	1011	0000	0000	0000	000	1
	20	00000000000	00	11100	000	1000	0000	0000	0000	000	1
	21	10001001110	11	11100	000	0010	0000	0000	0000	000	1
	22	00001110000	11	11000	000	0000	0000	0000	0000	000	1
	23	00010101110	10	11100	000	1010	0000	0000	0000	000	1
	24	00000000000	00	11100	000	1000	0111	1111	1111	000	0
	25	10011101100	11	11100	000	1000	1111	1111	1111	001	1
01A	26	00001010101	11	11100	000	0100	0000	0000	0000	000	1
	27	00001111111	11	11100	000	0000	0000	0000	0000	000	1
	28	10000010000	11	01101	000	1011	0000	0000	0000	000	1
	29	01101000111	11	11100	000	1000	1111	1111	1111	000	0
	30	01011111110	11	11100	000	1000	0100	0011	1111	000	1
	31	00000000010	11	11000	000	0000	1111	0000	0000	000	1
020	32	10000000010	11	01101	000	1011	0000	0000	0000	000	1
	33	10000000011	11	01101	000	1011	0000	0000	0000	000	1
	34	00001000011	11	11100	000	1010	0000	0000	0000	000	1
	35	10000000010	11	01101	000	1011	0000	0000	0000	000	1
	36	00010111110	10	11100	000	1010	0000	0000	0000	000	1
	37	00000000000	00	11100	000	1000	0000	0000	0000	000	1
	38	01110000000	10	01101	000	1000	1110	1111	1111	000	1
	39	11111111111	11	11100	000	1000	0111	1111	1111	000	1
	40	00000110001	11	11010	011	0000	0000	0000	0000	000	1
	41	11010011101	11	11100	000	1000	1111	1111	1111	000	0
02A	42	00011010110	11	11100	000	1000	0000	0000	0000	000	1
	43	00100001000	11	11100	000	1000	0000	0000	0000	000	1
	44	00000010111	11	11100	000	1010	0000	0000	0000	000	1
	45	00010111101	11	11100	000	1000	1111	1111	1111	000	0
	46	00000000000	00	11100	000	1000	0000	0000	0000	000	1
	47	00000011101	11	10001	000	0100	0000	0000	0000	000	1
030	48	00011101011	11	10110	000	1000	1111	1111	1111	000	1
	49	10000010011	11	00001	000	1010	1111	1111	1111	000	0
	50	00001010100	11	11100	000	1000	0000	0000	0000	000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
033	51	10101101101	11	11100	000	1000	0000	0000	0000	000	1
	52	01101001110	11	11100	000	1000	1011	1001	1111	000	1
	53	00000000000	00	11100	000	1000	0000	0000	0000	000	1
	54	10000010010	11	00101	000	1011	0000	0000	0000	000	1
	55	10000100010	11	01101	000	1011	1111	1111	1111	000	1
	56	10000011101	11	11100	000	1000	1111	1111	1111	000	0
	57	00000000000	00	11100	000	1000	1111	1111	1111	000	0
03A	58	01100111100	11	10110	000	1000	0000	0000	0000	000	1
	59	11100000100	10	11100	000	1010	0000	0000	0000	000	1
	60	10000111110	10	11100	000	1000	0000	0000	0000	000	1
	61	00000000000	00	11100	000	1000	0000	0000	0000	000	1
	62	01000000101	11	11100	000	1000	0000	0000	0000	000	1
	63	10101110010	11	11100	000	1000	1111	0111	1011	000	1
040	64	00011000000	11	00011	000	0100	0000	0000	0000	000	1
	65	10000110011	11	11100	000	1000	0000	0000	0000	000	1
	66	10010100000	11	10011	000	0000	0000	0000	0000	000	1
	67	10010100001	11	10011	000	0000	0000	0000	0000	000	1
	68	00000000000	00	11100	000	1000	0000	0000	0000	000	1
	69	00000000000	00	11100	000	1000	0000	0000	0000	000	1
	70	01000010100	10	11100	000	1000	0000	0000	0000	000	1
	71	01011101110	11	11100	000	1000	1111	1111	1111	000	1
	72	01001000100	10	11100	000	1000	0000	0000	0000	000	1
	73	01001000100	11	11100	000	1000	1111	1111	1111	000	0
04A	74	00011110000	11	11100	000	1000	0000	0000	0000	000	1
	75	00000101000	11	10010	000	1000	0000	0000	0000	000	1
	76	00100000111	11	11100	000	1000	0000	0000	0000	000	1
	77	10101000110	11	11100	000	1000	1111	1111	1111	000	0
	78	00001000100	11	11100	000	1010	0000	0000	0000	000	1
	79	10010101111	11	11100	000	1010	0000	0000	0000	000	1
050	80	10010110001	11	11100	000	0100	1011	1111	1111	000	1
	81	10000100010	11	11001	000	1010	1111	1111	1111	000	1
	82	01111000000	11	10110	000	1011	1111	1111	1111	000	1
	83	10000000011	11	01101	000	1011	1111	1111	1101	000	1
	84	10000000011	11	01101	000	1011	0000	0000	0000	000	1
	85	00001011100	11	11100	000	1000	0000	0000	0000	111	1
	86	10000000011	11	01101	000	1011	0011	1101	0000	000	1
	87	00011101110	11	00011	001	0000	0000	0000	0000	000	1
	88	10010101100	11	11100	000	0000	0111	1101	1111	000	1
	89	00001110110	11	11100	000	1000	1111	1111	1111	000	0
05A	90	00011000011	11	11100	000	1000	1011	1111	1111	000	1
	91	10100001110	10	11100	000	1000	1111	0111	1111	000	1
	92	11001111111	11	11100	000	1000	0000	0000	0000	000	1
	93	10111111010	10	11100	000	1000	1111	1111	1111	000	0
	94	00001110001	11	11100	000	0110	0000	0000	0000	000	1
	95	10000010000	11	01101	000	1011	0000	0000	0000	000	1
060	96	00010000010	10	11100	000	1000	1111	1111	1100	000	0
	97	00011000010	11	11100	000	1000	0000	0000	0000	000	1
	98	10010101101	11	11100	000	0000	0000	0000	0000	000	1
	99	11011011100	10	11100	000	1000	0000	0000	0000	000	1
	100	10011000010	11	11100	000	1000	0000	0000	0000	000	1
	101	10000000010	11	01101	000	1011	1111	1111	1111	000	0
	102	00011111000	11	01111	000	0000	0000	0000	0000	000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	153	0001001101	11	11100	000	1000	1111	1111	1111	000	1
	154	1101111001	11	01110	000	1000	1111	1111	1111	000	1
	155	0000000000	00	11100	000	1000	1111	1111	1111	000	0
06A	156	1001000111	11	01111	000	0000	0000	0000	0000	000	1
	157	0000100110	10	11100	000	1000	1111	1111	1111	000	1
	158	1000101100	11	11100	000	1000	0000	0000	0000	000	1
	159	0010011000	10	11100	000	1000	0000	0000	0000	000	1
	160	1000000011	11	11100	000	1000	1111	1111	1101	000	1
	161	1010000010	11	11100	000	1000	0000	0000	0000	000	1
070	162	0000110011	11	11100	000	1000	0000	0000	0000	000	1
	163	0111001000	11	11110	000	1010	1111	1111	1111	000	0
	164	0100010100	11	10110	000	1000	0000	0000	0000	000	1
	165	1111000110	10	11100	000	1000	0000	0000	0000	000	1
	166	0001011000	11	11100	000	1000	0000	0000	0000	000	1
	167	0001111010	11	11100	000	1000	1110	0111	1111	001	1
	168	0110010001	11	11100	000	1000	1111	1111	1111	000	0
07A	169	0110011001	11	11100	000	1000	0000	0000	0000	000	1
	170	0101001100	10	11100	000	1000	1111	1111	1111	000	1
	171	0000000101	11	11100	000	1010	0000	0000	0000	000	1
	172	1110010001	11	11100	000	1000	1111	1111	1111	000	0
	173	1001010010	11	11011	011	0000	0000	0000	0000	000	1
	174	1011000000	11	11100	000	1000	0000	0000	0000	000	1
080	175	0010001100	11	11100	000	0110	0000	0000	0000	000	1
	176	1010001001	11	11100	000	0110	1111	1110	1111	000	1
	177	1000000011	00	00101	000	1010	0000	0000	0000	000	1
	178	0000010001	11	11010	000	1010	0000	0000	0000	000	1
	179	1000000011	00	00101	000	1011	0000	0000	0000	000	1
	180	0000010001	11	11010	000	1010	1111	1111	1111	000	1
	181	0001111010	11	10010	000	1000	1111	0111	1111	000	1
	182	1000100010	10	10111	000	1000	0000	0000	0000	000	1
	183	0010010100	11	11100	000	1000	0000	0000	0000	000	1
08A	184	0100010001	11	11010	000	1010	1111	1111	1111	000	0
	185	1010011001	11	11100	000	1000	1111	1111	1111	000	0
	186	1010001001	11	11100	000	1000	0111	1111	1111	000	1
	187	1000000010	11	01101	000	1011	1111	1111	1111	000	0
	188	0110000110	10	11100	000	1000	0000	0000	0000	000	1
	189	1010010100	11	11100	000	1000	1111	1111	1111	000	1
090	190	0010110100	11	11100	000	1000	1110	1111	1111	000	1
	191	0010011100	11	10001	000	0110	0000	0000	0000	000	1
	192	1011111101	11	11100	000	1000	1111	1111	0011	000	1
	193	1010000101	11	11100	011	0000	0000	0000	0000	000	1
	194	1010000001	11	11100	011	0000	0000	0000	0000	000	1
	195	0010010101	11	11100	000	1000	0000	0000	0000	000	1
	196	0010011011	11	11100	011	0000	0000	0000	0000	000	1
	197	0010001101	11	10110	000	0110	0000	0000	0000	000	1
	198	0010000001	11	10110	000	1000	0000	0000	0000	000	1
	199	1010111001	11	11100	000	0100	1111	1111	1111	000	1
09A	200	0011001101	11	11100	000	1000	0000	0000	0000	000	1
	201	0100010001	11	11100	000	1000	1111	1111	1111	000	0

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	155	0010010110	10	10110	000	1000	1111	1111	1111	000	1
	156	1111101111	11	11100	000	1010	1111	1111	1111	000	0
	157	0010100000	10	11010	000	1010	1111	1111	1111	000	0
	158	0011000000	11	11100	000	1000	0000	0000	0000	000	1
	159	1001111111	11	10010	000	1000	0000	0000	0000	000	1
0A0	160	0010100010	11	11100	000	1000	0000	0000	0000	000	1
	161	1000000011	11	00101	000	1011	0000	0000	0000	000	1
	162	1000001011	11	11100	000	1000	0000	0000	0000	000	1
	163	1000000011	11	01101	000	1011	1111	1111	1111	000	1
	164	1010010101	11	11100	000	1000	0000	0000	0000	000	1
	165	1100110100	11	11100	000	1000	0000	0000	0000	000	1
	166	0010101000	11	11100	000	1000	0000	0000	0000	000	1
	167	0010101011	11	11100	000	0000	0100	0011	1111	000	1
	168	1010100011	11	11100	000	0000	0000	0000	0000	000	1
	169	0100101010	10	11100	000	1000	0000	0000	0000	000	1
0AA	170	0010000110	11	10010	000	0000	0000	0000	0000	000	1
	171	0111100010	11	11100	000	1000	0100	0011	1111	000	1
	172	1010100100	11	11100	000	1000	0000	0000	0000	000	1
	173	1011011100	11	11100	000	1000	0000	0000	0000	000	1
	174	0011100011	11	11010	000	1000	1110	1111	1111	000	1
	175	0010001111	11	11100	000	0000	0000	0000	0000	000	1
0B0	176	0011011011	11	11100	000	1000	1110	1111	1111	000	1
	177	0010111000	11	11100	000	1000	1110	1111	1111	000	1
	178	0010101111	11	11100	000	1000	0000	0000	0000	000	1
	179	0010110001	11	11100	000	0000	0000	0000	0000	000	1
	180	0010111001	11	11100	000	0100	0000	0000	0000	000	1
	181	0110000111	11	11010	000	1010	0000	0000	0000	000	1
	182	1110111001	11	11100	000	1000	1111	1110	1000	000	1
	183	0010111010	11	11100	000	0010	0000	0000	0000	000	1
	184	0011100011	11	11100	000	1000	0000	0000	0000	000	1
	185	0010101110	11	11100	000	1000	1111	1111	1111	000	0
0BA	186	0101110010	11	11100	000	1010	1111	0111	1011	000	1
	187	0111101000	11	10010	000	1010	1110	1111	1101	000	1
	188	1100001011	11	11100	000	1000	0000	0000	0000	000	1
	189	1100011111	11	11100	000	1000	0000	0000	0000	000	1
	190	0011110010	11	11010	000	1000	0000	0000	0000	000	1
	191	1010111111	11	11100	000	0000	1111	1111	1101	000	1
0C0	192	0011001010	10	10110	000	1000	0000	0000	0000	000	1
	193	0011000100	10	11100	000	1000	0000	0000	0000	000	1
	194	0011000110	11	11100	000	1000	1111	1110	1000	000	1
	195	0001111110	11	11100	000	1000	0000	0000	0000	000	1
	196	0011000010	11	11100	000	1000	0000	0000	0000	000	1
	197	1000000000	00	10000	000	1000	1111	1110	1111	000	1
	198	1011000010	11	11000	000	0000	1111	1111	1111	000	1
	199	0011010000	10	11100	000	1000	0000	0000	0000	000	1
	200	0011010100	10	10110	000	1000	1111	1111	1111	010	1
0CA	201	0100000110	10	11100	000	1000	0000	0000	0000	000	1
	202	1010011110	10	11010	000	1000	0000	0000	0000	000	1
	203	0010011110	10	11010	000	1000	1111	1111	1111	000	1
	204	1010111000	11	11100	000	1000	0000	0000	0000	000	1
	205	1010101001	11	11100	000	1000	1111	1111	1111	000	0
	206	1011000111	11	10110	000	1000	0000	0000	0000	000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	207	0010111110	10	1110	000	1000 1111 0111 1111	000	1
0D0	208	0011001000	11	1110	000	1000 0000 0000 0000	000	1
	209	1000000000	00	1000	000	1000 0000 0000 0000	000	1
	210	1000000000	00	1000	000	1000 1111 1110 1000	000	1
	211	0011101000	11	1110	000	1000 1111 1111 1111	000	1
	212	0011101110	10	1110	000	1000 1111 1111 1111	000	0
	213	0011001100	11	1110	000	1000 0000 0000 0000	000	1
	214	0011001000	11	1110	000	1000 1110 1111 1101	000	1
	215	0011001000	11	1110	000	1000 1111 0111 1111	000	1
	216	0000000000	00	1110	000	1000 0000 0000 0000	000	1
	217	0001111011	11	1110	000	1000 1111 1111 1111	000	0
0DA	218	1011001000	11	1101	011	0000 0000 0000 0000	000	1
	219	1001111110	11	1110	000	1000 0000 0000 0000	000	1
	220	0011100101	11	1110	000	0000 0000 0000 0000	000	1
	221	0000000000	00	1110	000	1000 0000 0000 0000	000	1
	222	0000000000	00	1110	000	1000 0000 0000 0000	000	1
	223	1000110010	10	0011	000	1000 0000 0000 0000	000	1
0E0	224	1011010110	11	1110	000	0010 0000 0000 0000	000	1
	225	1011100010	11	1110	000	1000 0000 0000 0000	000	1
	226	0011101100	11	1110	000	1000 0000 0000 0000	000	1
	227	0011001111	11	1001	000	0000 0000 0000 0000	000	1
	228	1011010010	11	1100	000	0000 0000 0000 0000	000	1
	229	1111101000	11	1110	000	1000 1111 1111 1111	000	1
	230	1101011001	11	1110	000	1000 1111 1111 1101	000	1
	231	0000000000	00	1110	000	1000 0000 0000 0000	000	1
	232	0011101010	10	0001	000	1000 1111 1110 1111	000	0
	233	1011010011	11	1001	000	0000 1111 1111 1111	000	0
0EA	234	0011101000	11	0001	000	0100 0000 0000 0000	000	1
	235	1000011011	11	1110	000	1000 1111 1110 1110	000	1
	236	1011101011	11	1110	000	0000 0000 0000 0000	000	1
	237	0001101101	11	1110	000	1000 0000 0000 0000	000	1
	238	0011110100	10	1001	000	1000 0000 0000 0000	000	1
	239	0011101010	11	1110	000	1000 0000 0000 0000	000	1
0F0	240	1011110000	11	1110	101	0100 1111 1111 1111	000	0
	241	0001011100	11	1110	000	1000 0000 0000 0000	000	1
	242	1010011100	11	1001	000	1000 0000 0000 0000	000	1
	243	0100010100	11	1110	000	1000 0000 0000 0000	000	1
	244	0011101010	11	1110	000	1000 0000 0000 0000	000	1
	245	1011000110	11	1001	000	0000 0000 0000 0000	000	1
	246	1011011100	10	1110	000	1000 0000 0000 0000	000	1
	247	0000000000	00	1110	000	1000 0000 0000 0000	000	1
	248	0011100010	11	1110	000	1000 0000 0000 0000	000	1
	249	0011101101	11	1110	000	1000 0000 0000 0000	000	1
0FA	250	0000000000	00	1110	000	1000 0000 0000 0000	000	1
	251	1000011010	11	1110	000	1000 1111 1111 1110	000	1
	252	0011101010	11	1110	000	1000 0000 0000 0000	000	1
	253	1011000101	11	1110	000	0000 0000 0000 0000	000	1
	254	0100001010	10	1110	000	1000 0000 0000 0000	000	1
	255	0011101001	11	1101	011	0100 0000 0000 0000	000	1
100	256	0101111000	11	1110	000	1000 1111 1111 1111	000	1
	257	0000101011	11	1110	000	1000 0000 0000 0000	000	1
	258	0000101010	11	1110	000	1000 1111 1111 1111	000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	259	0011100110	11	1110	000	1000 1111 1100 0000	000	1
	260	1100011100	10	1011	000	1000 0000 0000 0000	000	1
	261	0100100001	11	1110	000	0100 0000 0000 0000	000	1
	262	0011100010	11	1110	000	1000 0000 0000 0000	000	1
	263	0000001001	11	1110	000	1000 0000 0000 0000	000	1
	264	1000011010	11	1110	000	1000 0000 0000 0000	000	1
	265	1001111111	11	1001	000	1010 0000 0000 0000	000	1
10A	266	0011101010	11	1110	000	1000 0000 0000 0000	000	1
	267	0100100011	11	1100	000	0000 0000 0000 0000	000	1
	268	1000000110	11	1101	000	1000 1111 1111 1111	000	1
	269	0101100111	11	1110	000	1000 1111 1111 1111	000	0
	270	1100000000	00	1101	000	1000 0000 0000 0000	000	1
	271	1000000000	00	1110	000	1000 0000 0000 0000	000	1
110	272	0001101111	11	1110	000	1000 0000 0000 0000	000	1
	273	1101001110	11	1110	000	1000 0000 0000 0000	000	1
	274	0100110101	11	1110	000	1000 1110 1111 1111	000	1
	275	0110100110	11	1110	000	1000 1011 1000 1000	000	1
	276	1000100100	10	1110	000	1000 0000 0000 0000	000	1
	277	0100010011	11	1110	000	0000 0000 0000 0000	000	1
	278	0100010010	11	1110	000	0000 0000 0000 0000	000	1
	279	0100000011	11	1110	000	0000 0000 0000 0000	000	1
	280	0100011001	11	1101	011	0000 1111 1111 1111	000	1
	281	0011111110	11	1110	000	1000 1111 1111 1111	000	1
11A	282	1101001010	10	1110	000	1000 0000 0000 0000	000	1
	283	0100011000	11	1010	000	0000 0000 0000 0000	000	1
	284	0001100111	11	1110	000	1000 0000 0000 0000	000	1
	285	1011101000	11	1110	000	1000 0000 0000 0000	000	1
	286	0100101110	11	1110	000	1000 0000 0000 0000	000	1
	287	1000000010	11	1110	010	1000 0000 0000 0000	000	1
120	288	0001110000	11	1110	000	1000 1111 1111 1111	000	1
	289	0110101011	11	1110	000	1000 1111 1111 1111	000	0
	290	1001101000	11	1110	000	1000 1111 1111 1111	000	0
	291	0100010111	11	1110	000	1000 1111 1111 0111	000	1
	292	1000000100	10	1110	000	1000 0000 0000 0000	000	1
	293	0100010011	11	1110	000	0000 0000 0000 0000	000	1
	294	0100011010	11	1110	000	1000 1111 0111 1111	001	1
	295	0000000000	00	1110	000	1000 0000 0000 0000	000	1
	296	0010100100	11	1110	000	1000 0000 0000 0000	000	1
	297	0000000000	00	1110	000	1000 0000 0000 0000	000	1
12A	298	0011101010	11	1110	000	1000 0000 0000 0000	000	1
	299	0100100110	11	1110	000	0000 0000 0000 0000	000	1
	300	1100101100	11	1110	000	1000 1111 1111 1101	000	1
	301	0100110000	11	1001	000	0010 1111 1111 1111	000	0
	302	1000011010	11	1110	000	1000 0000 0000 0000	000	1
	303	1000011010	11	1110	000	1000 0000 0000 0000	000	1
130	304	0100111000	11	1010	000	0011 1111 1111 1111	000	1
	305	1100110101	11	1110	000	1000 1111 1111 1111	110	1
	306	1000011010	11	1110	000	1000 1111 1111 1111	000	1
	307	0010011000	11	1110	000	1000 1111 0111 1111	000	1
	308	1100101010	11	1110	000	1000 0000 0000 0000	000	1
	309	1000110100	11	1110	000	1000 0000 0000 0000	000	1
	310	1001001110	11	1110	000	1000 0000 0000 0000	000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	311	0100111001	11	01111	000 0100	0000 0000	0000	000 1
	312	0100001100	11	11001	011 0000	1111 1111	1111	000 1
	313	0101111010	10	11100	000 1000	1111 1111	1111	000 0
13A	314	0011101010	11	11100	000 1000	0000 0000	0000	000 1
	315	0001001011	11	11010	000 1000	0000 0000	0000	000 1
	316	1001110001	11	01101	000 1011	0000 0000	0000	000 1
	317	1101011100	10	11100	000 1000	0000 0000	0000	000 1
	318	0100000010	11	11100	000 0000	0000 0000	0000	000 1
	319	0100110010	11	11010	011 0000	0000 0000	0000	000 1
140	320	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	321	0001111100	11	11100	000 1000	1111 1111	1111	000 1
	322	1001111000	11	11100	000 1010	1111 1111	1111	000 1
	323	0000000001	11	11100	000 1000	1101 1110	1000	000 1
	324	0101010011	11	11100	000 0000	0110 0100	1110	000 1
	325	0110000100	10	11010	000 1010	0000 0000	0000	000 1
	326	0011100100	11	11100	000 1000	0111 1111	1111	000 1
	327	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	328	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	329	0101111111	11	11100	000 1000	1111 1111	1111	000 0
14A	330	0011101010	11	11100	000 1000	0000 0000	0000	000 1
	331	1101011011	11	01000	000 0000	0000 0000	0000	000 1
	332	0101011001	11	11100	000 0100	0000 0000	0000	000 1
	333	1011000001	11	11100	000 1000	0000 0000	0000	000 1
	334	1001011111	11	11100	000 1000	0000 0000	0000	000 1
	335	0011110001	11	11100	000 1000	0000 0000	0000	000 1
150	336	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	337	0000110010	11	11100	000 1000	1111 1111	1111	000 1
	338	1000000000	00	11100	000 1000	1111 1101	1111	000 1
	339	1101011111	11	11110	010 1000	1111 1111	1111	000 1
	340	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	341	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	342	0100001000	11	11100	000 1000	0000 0000	0000	000 1
	343	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	344	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	345	1100011110	11	11100	000 1000	1111 1111	1111	000 0
15A	346	1010001101	11	11100	000 1000	0000 0000	0000	000 1
	347	0101001010	10	11010	000 1000	1110 1111	1111	000 1
	348	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	349	0101010110	11	11100	000 1000	1111 1111	1111	000 0
	350	0010101010	11	11010	000 1000	0000 0000	0000	000 1
	351	0101001001	11	11100	000 0100	0000 0000	0000	000 1
160	352	0110000011	11	11100	000 1000	1111 0111	1011	000 1
	353	1101111101	11	11100	000 0100	0000 0000	0000	000 1
	354	1010111110	11	01101	000 1011	0000 0000	0000	000 1
	355	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	356	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	357	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	358	0001001100	11	11100	000 1000	1111 1111	1111	000 1
	359	0011111111	11	11100	000 1000	0000 0000	0000	000 1
	360	0010001101	11	11100	000 1000	0000 0000	0000	000 1
	361	0101000110	11	11100	000 0000	0000 0000	0000	000 1
16A	362	0000000000	00	11100	000 1000	0000 0000	0000	000 1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	363	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	364	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	365	1001000100	10	11100	000 1000	0000 0000	0000	000 1
	366	0011110011	11	11100	000 1000	0000 0000	0000	000 1
	367	0101010010	11	11100	001 0000	0000 0000	0000	000 1
170	368	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	369	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	370	0011111000	10	11100	000 1000	0000 0000	0000	000 1
	371	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	372	0011101010	11	11100	000 1000	0000 0000	0000	000 1
	373	0101011010	10	10110	000 1000	0000 0000	0000	000 1
	374	0010101101	11	11100	000 1000	0000 0000	0000	000 1
	375	0010111100	11	11100	000 1000	0000 0000	0000	000 1
	376	0101000001	11	11011	011 0000	0000 0000	0000	000 1
	377	0101010001	11	11011	011 0000	0000 0000	0000	000 1
17A	378	0100111110	11	11100	000 1000	0000 0000	0000	000 1
	379	1110001100	10	11100	000 1000	0000 0000	0000	000 1
	380	1010010101	11	11100	000 1000	1111 1110	1110	000 1
	381	1110100111	11	11100	000 1000	1111 1111	1111	000 0
	382	0101111101	11	11100	000 1000	0000 0000	0000	000 1
	383	0101011101	11	11100	000 0100	0000 0000	0000	000 1
180	384	1100000000	00	11010	000 1000	0000 0000	0000	000 1
	385	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	386	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	387	0100001110	11	11100	000 1000	0000 0000	0000	000 1
	388	1000000011	11	00101	000 1011	0000 0000	0000	000 1
	389	1000010101	11	10010	000 1000	0000 0000	0000	000 1
	390	0011101010	11	11100	000 1000	0000 0000	0000	000 1
	391	0101101000	10	10110	000 1000	0000 0000	0000	000 1
	392	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	393	1010001010	10	10111	000 1000	0100 0011	1111	000 1
18A	394	0101110110	10	11100	000 1000	0000 0000	0000	000 1
	395	0010111100	11	11100	000 1000	0000 0000	0000	000 1
	396	1000011001	11	11100	000 1000	0000 0000	0000	000 1
	397	1000011111	11	11100	000 1000	0000 0000	0000	000 1
	398	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	399	1110011100	11	11000	001 0000	0000 0000	0000	000 1
190	400	0001010000	11	11100	000 1000	0000 0000	0000	000 1
	401	1001110100	11	11100	000 1000	0000 0000	0000	000 1
	402	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	403	0100111100	11	11100	000 1010	1111 1111	1111	000 1
	404	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	405	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	406	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	407	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	408	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	409	1001101010	11	11100	000 1000	0000 0000	0000	000 1
19A	410	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	411	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	412	0000111010	10	10000	000 1010	1111 1111	0111	000 1
	413	1000101010	10	11010	000 1000	0000 0000	0000	000 1
	414	0000000000	00	11100	000 1000	0000 0000	0000	000 1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	415	1100001010	11	01011	010 1011	0000 0000 0000	000	1
1A0	416	1001111010	11	11100	000 1000	1111 1111 1111	000	1
	417	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	418	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	419	0001110111	11	11100	000 1000	1111 1111 1111	000	1
	420	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	421	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	422	0110110000	11	11100	000 0000	0000 0000 0000	000	1
	423	0110010011	11	11011	011 0010	0000 0000 0000	000	1
	424	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	425	0010010100	11	11100	000 1000	0000 0000 0000	000	1
1AA	426	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	427	0001101100	11	11100	000 1000	0000 0000 0000	000	1
	428	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	429	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	430	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	431	0000000000	00	11100	000 1000	0000 0000 0000	000	1
1B0	432	1100111011	11	11100	000 1000	1111 1111 1111	000	1
	433	0001010010	11	11100	000 1010	1111 1111 1011	000	1
	434	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	435	0110110001	11	11000	000 0010	0000 0000 0000	000	1
	436	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	437	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	438	0110111011	11	10011	000 0000	0000 0000 0000	000	1
	439	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	440	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	441	1100000110	10	11100	000 1000	0000 0000 0000	000	1
1BA	442	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	443	1010111100	11	11100	000 1000	0100 0011 1111	000	1
	444	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	445	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	446	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	447	0000100001	11	11111	111 1000	0000 0000 0000	000	1
1C0	448	0111000010	11	11100	000 0000	0000 0000 0000	000	1
	449	0111010011	11	11100	000 0000	0000 0000 0000	000	1
	450	1000111100	11	11100	000 1000	1111 1111 1111	000	1
	451	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	452	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	453	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	454	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	455	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	456	1000010000	11	01101	000 1011	0000 0000 0000	000	1
	457	0000000000	00	11100	000 1000	0000 0000 0000	000	1
1CA	458	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	459	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	460	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	461	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	462	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	463	0111010010	11	11100	001 0000	0000 0000 0000	000	1
1D0	464	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	465	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	466	1000000000	00	11100	000 1000	1111 1111 0111	000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	467	0111000010	11	11100	000 1000	1111 0111 1011	000	1
	468	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	469	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	470	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	471	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	472	1000111110	11	11100	000 1000	1111 1111 1111	000	0
	473	0000000000	00	11100	000 1000	0000 0000 0000	000	1
1DA	474	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	475	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	476	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	477	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	478	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	479	0000000000	00	11100	000 1000	0000 0000 0000	000	1
1E0	480	0001010010	10	11100	000 1010	0000 0000 0000	000	1
	481	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	482	1100111100	11	11100	000 1000	0000 0000 0000	000	1
	483	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	484	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	485	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	486	1101111110	11	11100	000 1000	1111 1111 1111	000	1
	487	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	488	0011000001	11	00101	000 1011	0000 0000 0000	000	1
	489	0010110010	11	11100	000 1000	0000 0000 0000	000	1
1EA	490	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	491	0100010000	11	11100	000 1000	0000 0000 0000	000	1
	492	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	493	0111010001	11	11100	000 1000	0000 0000 0000	000	1
	494	0100111101	11	11100	000 1000	0000 0000 0000	000	1
	495	0100111101	11	11100	000 1000	0000 0000 0000	000	1
1F0	496	1111010000	10	11100	000 1000	0000 0000 0000	000	1
	497	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	498	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	499	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	500	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	501	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	502	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	503	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	504	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	505	0000000000	00	11100	000 1000	0000 0000 0000	000	1
1FA	506	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	507	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	508	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	509	0000000000	00	11100	000 1000	0000 0000 0000	000	1
	510	0100111101	11	11100	000 1000	0000 0000 0000	000	1
	511	0100111101	11	11100	000 1000	0000 0000 0000	000	1
200	512	1000011010	11	11100	000 1000	0101 1111 1100	000	1
	513	0000000000	00	11100	000 1000	1101 1011 1011	110	1
	514	1000000000	00	11100	000 1000	1001 1111 1011	000	1
	515	1000000000	00	11100	000 1000	1111 1111 0000	000	1
	516	1011110010	10	11100	000 1000	1111 1111 0000	111	1
	517	1000110100	10	11100	000 1000	1111 1111 0101	101	1
	518	0101100001	11	01101	000 1000	0100 1111 1001	000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	519	1000111001	11	11100	000 0110	1111 1111 0110	000	1
	520	1000011010	10	11110	010 1000	1111 1111 0101	000	1
	521	1000000010	11	01101	000 1011	1011 1001 1100	000	1
20A	522	1001001100	11	11100	000 1000	1101 0001 1011	000	1
	523	1000110111	11	11100	000 0000	1111 1111 0010	000	1
	524	0011011111	11	00101	000 1011	0010 1001 1100	000	1
	525	1000011100	10	11100	000 1000	1001 1111 1011	110	1
	526	0000011011	11	11100	000 1000	0010 1100 1110	000	1
	527	1000000110	11	11100	000 1000	1001 1111 1011	000	1
210	528	1000000000	00	11100	000 1000	1001 1111 1001	000	1
	529	1001001001	11	11100	000 0110	1111 1111 0000	000	1
	530	1000000000	00	10010	000 1000	1001 1111 1001	000	1
	531	1000000000	00	10010	000 1000	1001 1111 1011	000	1
	532	0000010010	10	00011	000 1010	0101 1111 1100	000	1
	533	1000011100	11	10010	000 0010	1111 1111 1000	000	1
	534	0000000000	00	11100	000 1000	1111 1111 0000	000	1
	535	1000001110	11	11000	000 0000	1101 1011 1100	110	1
	536	1000101000	11	11100	000 1000	0000 0000 0000	000	1
	537	1000111001	11	01111	000 0110	0101 0011 1101	000	1
21A	538	1000001000	11	00011	000 0100	1111 1111 0000	000	1
	539	1000011000	11	01111	100 0100	1111 1111 1000	000	1
	540	0000011010	11	11100	000 1000	0101 1111 1100	000	1
	541	0101001100	11	11100	000 1000	1101 1011 1011	000	1
	542	1001001010	11	11100	000 1000	0100 1111 1011	000	1
	543	1000011110	11	11000	000 0000	0101 1111 1101	000	1
220	544	1000010010	11	00101	000 1011	1111 1111 0000	000	1
	545	1000010011	11	00101	000 1011	1111 1111 0000	000	1
	546	1000000000	00	10010	000 1000	1111 1111 0000	000	1
	547	1001000010	10	11100	000 1000	1111 1111 0000	000	1
	548	0001110010	10	11010	000 1000	1111 1111 0000	100	1
	549	1000110100	11	11100	000 1000	1111 1111 0101	000	1
	550	0000011100	11	11100	000 1010	1101 1111 1100	101	1
	551	1100000000	00	00011	000 1010	1111 1111 0000	000	1
	552	1000111000	11	01100	101 0110	0110 1111 1110	000	1
	553	1000001010	11	11100	000 1000	0000 0000 0000	000	0
22A	554	0110011100	10	10110	000 1010	1101 1111 1111	111	1
	555	1011010101	11	11010	000 1010	0110 0101 0110	000	1
	556	0101110010	11	11100	000 1000	0001 1111 1100	000	1
	557	0000001010	11	11010	000 0100	0011 1111 1110	000	1
	558	1000100111	11	01110	000 0010	1111 1111 0100	000	1
	559	0000011101	11	10100	000 0100	0101 0011 1101	000	1
230	560	1000110001	11	11010	000 1000	0100 0111 1100	000	1
	561	1000000000	00	10010	000 1000	1001 1111 1011	000	1
	562	1000000000	00	11100	000 1000	1001 1111 1001	000	1
	563	1000110000	10	10010	000 1000	1111 1111 0101	110	1
	564	0000110100	11	11100	011 0000	1101 1011 1010	000	1
	565	0100010110	11	11100	000 1000	0101 1111 1100	000	1
	566	1000101001	11	11100	000 0100	1111 1111 0000	000	1
	567	1010001011	11	11100	000 1000	0111 1100 1100	000	1
	568	0100000000	00	00011	000 1010	1011 1001 1100	000	1
	569	1100000000	00	00011	000 1010	0000 0000 0000	000	1
23A	570	1100010100	11	10110	000 1000	1101 1111 1111	000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	571	0110000100	10	10010	000 1010	0110 0101 0110	101	1
	572	0110000000	11	11100	000 1000	0110 0110 0110	010	1
	573	0000010000	10	11100	000 1000	0011 1100 1100	001	1
	574	0000111100	11	10110	000 1000	1111 1111 0111	000	1
	575	0100110110	11	11100	000 1000	1110 1111 1100	000	1
240	576	1001010000	11	11100	001 0000	1111 1111 0000	000	1
	577	1001010001	11	11100	001 0000	1111 1111 0000	000	1
	578	1100000000	00	00011	000 1000	1101 1011 1011	000	1
	579	0001101000	11	11100	000 0000	1111 1111 0001	000	1
	580	0111000001	11	11100	000 1000	1111 1111 0000	000	1
	581	1001111011	11	00111	011 0000	0110 1111 1110	000	1
	582	0000000000	00	11100	000 1000	1101 1111 1100	110	1
	583	0000001011	11	11100	000 1000	1101 1111 1111	000	1
	584	0000000000	00	11100	000 1000	1101 1011 1011	110	1
	585	1101100111	11	11100	000 1000	1101 1111 1111	000	1
24A	586	1001001011	11	11100	000 1000	1101 1111 1011	000	1
	587	1001001001	11	11100	000 0100	0001 1111 0000	000	1
	588	1001101001	11	11010	011 0110	0101 0100 1100	000	1
	589	1001101011	11	11100	000 0000	1111 1111 0010	000	1
	590	0000111110	11	11100	000 1000	0110 0101 0110	000	1
	591	1011011110	10	11100	000 1000	1101 1111 1100	100	1
250	592	0000100010	11	11100	000 1010	1001 0010 1110	000	1
	593	0000100010	11	11100	000 1010	1101 0000 1100	000	1
	594	1010110011	11	11100	000 1000	1111 0110 0110	000	1
	595	1111001111	11	11100	000 1000	1001 1111 1011	000	1
	596	0001100001	11	11100	000 1000	0110 1111 1001	000	1
	597	1001100011	11	11100	000 1000	0110 1111 1001	000	1
	598	0001100100	11	11100	000 1000	1111 1111 0000	000	1
	599	0001100101	11	11100	000 0110	1111 1111 0000	000	1
	600	1001100111	11	11100	000 1000	1110 1111 1110	000	1
	601	1010010001	11	11100	000 1000	0000 0000 0000	000	1
25A	602	0001011010	11	11100	000 0000	1110 1011 1110	000	1
	603	1100011011	11	11100	000 1000	1111 1111 0000	100	1
	604	0000000000	00	11100	000 1000	1111 1111 0011	000	1
	605	1010001000	11	11100	000 1000	1111 0101 0110	100	1
260	606	1001011101	11	11100	000 0100	1111 1111 0110	000	1
	607	0000001110	10	11010	000 1010	1111 1111 0000	100	1
	608	0000101111	11	11100	000 1000	0111 0011 1011	000	1
	609	0001001010	11	11100	000 1000	1111 1111 0101	000	1
	610	1001011000	11	11100	000 0000	1110 1001 1110	000	1
	611	1001100100	10	11100	000 1000	0101 1111 1100	110	1
	612	1001100110	11	11100	000 1000	1111 1111 0101	000	1
	613	1100011011	11	11100	000 1000	1111 1111 0000	000	1
	614	1001010110	11	11100	000 0000	1110 1011 1110	000	1
	615	1001011010	10	11100	000 1000	1111 1111 0000	110	1
	616	1001010011	11	11000	000 0000	1001 0111 1001	000	1
26A	617	0001010100	10	11100	000 1010	0000 0000 0000	000	1
	618	1001110111	11	11100	000 0000	1111 1111 0100	000	1
	619	1000111101	11	11100	000 1000	1111 1111 0000	010	1
	620	0001100110	11	11100	000 1000	1111 1111 0101	000	1
	621	0100011100	10	11100	000 1000	1001 1111 1011	110	1
	622	1000100001	11	00001	000 1010	1110 1100 1100	000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	623	0101001110	11	00101	000 1011	0110 0101	0110 000	1
270	624	1000111010	11	11100	000 1000	1001 1111	1100 000	1
	625	0001111101	11	11100	000 0100	1111 1111	0110 000	1
	626	1100000000	00	10110	000 1000	1101 1111	1111 000	1
	627	1000010100	10	10010	000 1010	1111 1111	0101 101	1
	628	1001111001	11	11100	000 0100	1111 1111	0110 000	1
	629	1001111101	11	11100	000 0100	1111 1111	0110 000	1
	630	1001110100	11	11100	000 1000	0011 1101	1011 000	1
	631	1100100110	11	11100	000 1000	1101 0010	1100 000	1
	632	1000011010	11	01101	000 1011	0110 0000	1100 000	1
	633	0110010000	11	11100	000 1000	0000 0000	0000 000	1
27A	634	1101111100	11	11100	000 1000	1111 1111	0101 000	1
	635	0000010110	11	01110	000 1000	1001 1111	1011 110	1
	636	0000000000	00	11100	000 1000	0101 0000	1100 000	1
	637	1001110110	11	11100	000 1000	0000 0000	0000 000	1
	638	0001011011	11	10010	000 0000	1111 1111	1000 000	1
	639	1010101111	11	00101	000 1011	0110 0101	0110 000	1
280	640	1010001010	11	11010	000 0110	1101 1011	0000 000	1
	641	1010001100	11	11010	000 1000	1111 1111	0000 000	1
	642	0110101001	11	11100	000 1000	1010 1101	1011 000	1
	643	1011100100	10	11010	000 1000	0110 0111	1100 100	1
	644	0000000000	00	11100	000 1000	1111 1111	0000 000	1
	645	1010001100	11	11100	000 1000	0110 1100	1100 000	1
	646	0010011010	11	11100	000 1000	0000 0000	0000 000	1
	647	0111101101	11	11100	000 1000	1111 1111	0101 001	1
	648	1001111110	11	11010	000 1000	1110 1101	1011 000	1
	649	0010100010	10	11010	000 1010	0000 0000	0000 010	1
28A	650	1000000010	11	00101	000 1011	0110 0110	0110 000	1
	651	0010010101	11	11100	000 1000	1111 1111	0111 000	1
	652	0100000001	11	11100	000 1000	0110 0110	0110 000	1
	653	0101110100	10	11100	000 1000	1101 1111	1111 110	1
	654	0011110010	11	00011	000 1000	1111 1101	1100 110	1
	655	0011110010	11	00011	000 1000	1111 0111	1100 000	1
290	656	1010011010	11	11010	000 0110	1101 1011	0110 000	1
	657	1100000000	00	00011	000 1010	1111 1111	0011 000	1
	658	1010000110	11	11100	011 0000	0110 1111	1010 000	1
	659	1010011000	11	11100	011 0000	0110 1111	1010 000	1
	660	1110101001	11	11100	000 1000	1111 1111	0011 000	1
	661	1010011011	11	11100	011 0000	0110 0101	0001 000	1
	662	1010011101	11	10110	000 0110	0110 1111	1110 000	1
	663	1010101011	11	10110	000 1000	1101 1111	1100 000	1
	664	1011000100	11	11100	000 1000	0110 0111	1110 000	1
	665	0000000000	00	11100	000 1000	1101 1011	1011 000	1
29A	666	1111101111	11	10010	000 1010	0110 0110	0110 000	1
	667	1010010110	10	10110	000 1000	0000 0000	0000 100	1
	668	0100001001	11	10010	000 1010	0110 0110	0110 000	1
	669	1010100000	10	11010	000 1010	0000 0000	0000 010	1
	670	1011000000	11	11100	000 1000	1110 0101	0110 000	1
	671	1001111111	11	10010	000 1000	1111 1111	0101 000	1
2A0	672	0010100110	11	11100	000 1000	1111 1111	0101 000	1
	673	1000000011	11	00101	000 1011	1001 1111	1001 000	1
	674	0000000000	00	11100	000 1000	0101 0100	1100 000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	675	0010001011	11	11100	000 1000	1001 1111	1011 000	1
	676	1010101010	11	11100	000 1000	1111 1111	0101 000	1
	677	0011011010	11	11100	000 1000	1111 1111	0101 000	1
	678	1011100111	11	11100	000 1000	0101 0011	1100 000	1
	679	0000000000	00	11100	000 1000	1111 1111	1000 000	1
	680	0000000000	00	11100	000 1000	1111 1111	0010 000	1
	681	0001100111	11	11100	000 1000	1111 1111	0101 011	1
2AA	682	1010101110	11	11100	000 0000	1111 1111	0001 000	1
	683	0010001001	11	11100	000 0110	1111 1111	0000 000	1
	684	1000010000	11	11010	000 1010	0110 1111	1100 110	1
	685	1010110000	10	11010	000 1000	0110 1111	1100 111	1
	686	1001011110	11	11100	000 1000	1110 1100	1011 000	1
	687	0010001110	11	10010	000 0010	1111 1111	0001 000	1
2B0	688	1000000000	00	10010	000 1000	1001 1111	1001 000	1
	689	1000000000	00	11100	000 1000	1001 1111	1001 000	1
	690	0001111111	11	11100	000 1000	1111 1111	0101 000	1
	691	0010110000	11	11100	000 0000	1111 1111	0001 000	1
	692	0000000000	00	11100	000 1000	1111 1111	0110 000	1
	693	1000101110	11	11100	000 1000	0110 1111	1100 000	1
	694	0000000000	00	11100	000 1000	1111 1111	0011 000	1
	695	0010111011	11	10010	000 0010	1111 1111	1000 000	1
	696	1011011010	10	11100	000 1000	1111 1100	1011 100	1
	697	1011001001	11	11100	000 1000	0000 0000	0000 000	1
2BA	698	1010011110	10	11010	000 1000	1101 1111	1111 010	1
	699	0010011110	10	11010	000 1000	1111 1100	1111 010	1
	700	1000011001	11	11111	111 1000	1101 1011	1001 000	1
	701	0110001010	10	11100	000 1000	0101 0011	1100 011	1
	702	1011001110	10	11100	000 1000	1111 1100	1100 100	1
	703	0101011110	11	11100	000 1000	1111 1111	1000 000	1
2C0	704	1010111010	10	10110	000 1000	1111 1111	0000 100	1
	705	1011101100	11	11100	000 1000	1111 1111	0101 110	1
	706	1011000011	11	11010	000 1000	0111 1111	1111 000	1
	707	0010110111	11	11100	000 1000	1111 1111	0101 000	1
	708	0000011110	11	11100	000 1000	0110 0110	0110 000	1
	709	0010111001	11	11100	000 1000	1001 1111	1011 000	1
	710	1011101110	10	11100	000 1000	1111 1111	0001 100	1
	711	0011010010	10	11100	000 1000	1111 1111	0000 111	1
	712	0010110011	11	11100	000 1000	1101 0110	0110 010	1
	713	0011111000	10	11100	000 1000	0101 0011	1100 110	1
2CA	714	0000000000	00	11100	000 1000	1111 1101	1111 010	1
	715	0010001010	11	11100	000 1000	1101 1111	1111 010	1
	716	1001001111	11	11100	000 1000	1111 1001	1100 000	1
	717	1010111101	11	10010	000 1000	0101 0111	1100 000	1
	718	0011000111	11	10110	000 1000	1101 0110	0110 000	1
	719	0011001110	11	11100	000 1000	1111 1101	1100 100	1
2D0	720	0000000000	00	11100	000 1000	1111 1101	1111 000	1
	721	0000000000	00	11100	000 1000	1001 1111	1011 000	1
	722	0101110101	11	11100	000 1000	1001 1111	1011 000	1
	723	0000000111	11	11100	000 1000	1111 1100	1111 000	1
	724	1011111110	11	11010	000 1000	1111 1111	0000 111	1
	725	0101000101	11	11100	000 1010	1111 1111	0101 000	1
	726	1011011000	11	11100	000 1010	1111 1101	1111 000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	727	1011001100	11	11100	000 1000	1111 1100	1111	000 1
	728	1010000011	11	01101	000 1011	1111 1001	1100	000 1
	729	0011001001	11	11100	000 1000	1011 1001	1100	000 1
2DA	730	1011010111	11	11100	000 0000	1111 1111	1000	000 1
	731	1011011010	11	11100	000 1000	1111 1101	1011	000 1
	732	1011001101	11	11010	000 0100	1111 1111	0010	000 1
	733	0011111011	11	11100	000 0000	0110 1111	1110	000 1
	734	1011100000	10	11100	000 1000	0101 1111	1100	100 1
	735	1011100010	10	11100	000 1000	0101 1111	1100	110 1
2E0	736	1011010110	11	11100	000 0010	1111 1111	1000	000 1
	737	0011100000	10	11100	000 1000	1111 1111	0000	110 1
	738	0011100000	11	11100	000 1000	1111 1111	0101	000 1
	739	1011010110	11	11100	000 0010	1111 1111	1000	000 1
	740	1011100101	11	11010	000 1000	0110 1111	1100	000 1
	741	1000000000	00	10010	000 1000	1001 1111	1001	000 1
	742	1000011010	11	11100	000 1000	1111 1111	0101	000 1
	743	1011101101	11	11100	000 1000	1111 1111	0011	000 1
	744	1011011001	11	11100	000 0100	1110 1001	1110	000 1
	745	1011101010	11	11100	000 1000	1101 1011	1011	110 1
2EA	746	0100000100	11	11100	000 1000	1101 1111	1100	000 1
	747	1000011010	11	11100	000 1000	1111 1111	0000	000 1
	748	1011100110	11	11100	000 0000	0000 1111	1110	000 1
	749	1001101101	11	11100	000 1000	1011 1001	1100	000 1
	750	1011110100	10	11100	000 1000	1111 1111	0000	110 1
	751	0011101010	11	11100	000 1000	1111 1111	0000	000 1
2F0	752	0110110110	11	11100	000 1000	0110 1111	1110	000 1
	753	0000000000	00	11100	000 1000	0101 1111	1100	000 1
	754	1000110100	10	11100	000 1000	1111 1111	0101	101 1
	755	0100100100	11	10110	000 1000	1101 1111	1111	000 1
	756	0011101010	11	11100	000 1000	1111 1111	0000	000 1
	757	1011010100	11	11100	000 0100	0110 1111	1110	000 1
	758	0001000000	11	11100	000 1000	1101 1011	1011	011 1
	759	0001011101	11	11100	000 1000	0101 0000	1100	000 1
	760	0000000000	00	11100	000 1000	0101 1111	1100	000 1
	761	0000000000	00	11100	000 1000	1111 1111	0011	000 1
2FA	762	0011101010	11	11100	000 1000	1111 1111	0000	000 1
	763	0000000100	11	11100	000 1000	1101 0111	1100	000 1
	764	1101010111	11	11100	000 1000	0110 0110	0110	000 1
	765	1011001011	11	11100	011 0000	0110 1111	1010	000 1
	766	1011110110	10	11100	000 1000	1111 1111	0000	110 1
	767	0011101010	11	11100	000 1000	0101 0100	1100	000 1
300	768	1000010000	11	01101	000 1011	0100 1111	1100	000 1
	769	1000000011	11	01101	000 1011	0100 1111	1011	000 1
	770	1000010010	11	01101	000 1011	0111 1111	1100	000 1
	771	1000100010	11	01101	000 1011	0111 1111	1011	000 1
	772	1000101010	10	11010	000 1000	1110 0110	0110	010 1
	773	0100000000	11	11000	000 0000	1111 1111	0000	000 1
	774	0011100010	11	11100	000 1000	0101 1111	1100	000 1
	775	0011011001	11	11100	000 1000	1111 1111	0011	000 1
	776	1000010010	11	01101	000 1011	0110 1111	1100	000 1
	777	1000100010	11	01101	000 1011	0110 1111	1011	000 1
30A	778	0110001100	10	11100	000 1000	1111 1111	0000	000 1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	779	0100110011	11	11010	000 0000	1111 1111	1000	000 1
	780	1000010001	11	11100	000 1010	1101 1111	1100	000 1
	781	1000010001	11	11100	000 1010	1001 1111	1100	000 1
	782	0000000011	11	11100	000 1000	1101 1111	1100	000 1
	783	0000000011	11	11100	000 1000	1001 1111	1100	000 1
310	784	0000100000	10	11100	000 1010	0100 1111	1011	011 1
	785	0000100000	10	11100	000 1010	0100 1111	1011	011 1
	786	0000100000	10	11100	000 1010	0111 1111	1011	011 1
	787	0000100000	10	11100	000 1010	0111 1111	1011	011 1
	788	1000111010	10	10010	000 1010	1101 0101	0110	010 1
	789	0000000000	00	11100	000 1000	1101 1011	1010	000 1
	790	0100100000	11	11100	000 0000	1111 1111	0001	000 1
	791	0000000000	00	11100	000 1000	1110 1011	1110	000 1
	792	0000100000	10	11100	000 1010	0110 1111	1011	011 1
	793	0000100000	10	11100	000 1010	0110 1111	1011	011 1
31A	794	0100010110	11	11100	000 1000	0110 1111	1100	110 1
	795	0100110001	11	11100	000 0000	0110 1111	1110	000 1
	796	0001100111	11	11100	000 1000	1101 1111	1100	000 1
	797	1000011100	11	11100	000 1000	1101 1011	1011	000 1
	798	1101011010	11	11100	000 1000	1111 1111	0101	000 1
	799	0000111000	11	11100	000 1000	1111 1111	0011	000 1
320	800	1000010010	11	01101	000 1011	1100 1101	1100	000 1
	801	1000100010	11	01101	000 1011	1000 1101	1011	000 1
	802	1000010010	11	01101	000 1011	1111 1101	1100	000 1
	803	1000100010	11	01101	000 1011	1011 1101	1011	000 1
	804	1001110010	10	10010	000 1000	1110 0110	0110	010 1
	805	0000000000	00	11100	000 1000	1101 1011	1010	000 1
	806	1000000111	11	01110	000 1000	0111 1111	1011	000 1
	807	0000000000	00	11100	000 1000	0000 0000	0000	000 1
	808	0000010000	10	11010	000 1010	1110 1101	1100	001 1
	809	0010000010	10	11010	000 1010	1010 1101	0000	001 1
32A	810	0100001101	11	11100	000 0100	1111 1111	0000	000 1
	811	0000000000	00	11100	000 1000	1110 1001	1110	000 1
	812	0100101101	11	11010	011 0110	0101 0100	1110	000 1
	813	0100101100	11	11100	001 0000	1111 1111	0110	000 1
	814	0000000000	00	11100	000 1000	1101 1111	1100	000 1
	815	0000000000	00	11100	000 1000	0101 1111	1100	000 1
330	816	1000010010	11	01101	000 1011	1100 1100	1100	000 1
	817	1000100010	11	01101	000 1011	1000 1100	1011	000 1
	818	0000110110	10	00011	000 1010	1111 1100	1100	001 1
	819	1000100010	11	01101	000 1011	1011 1101	1011	000 1
	820	0010101100	11	11100	000 1000	1101 1111	1011	000 1
	821	1001110010	10	10010	000 1000	1111 0101	0110	010 1
	822	0001010000	11	01110	000 1000	1001 1111	1011	000 1
	823	0000000000	00	11100	000 1000	1111 1111	0110	000 1
	824	0000010000	10	11010	000 1010	1110 1100	1100	001 1
	825	0010000100	10	11010	000 1010	1010 1100	0000	001 1
33A	826	0000000000	00	11100	000 1000	1111 1111	0000	000 1
	827	1100000101	11	11100	000 1000	1101 1111	1100	000 1
	828	0000101100	11	11100	000 1000	1001 1111	1001	000 1
	829	0000000000	00	11100	000 1000	1111 1111	0101	111 1
	830	0000000000	00	11100	000 1000	1111 1111	0010	000 1

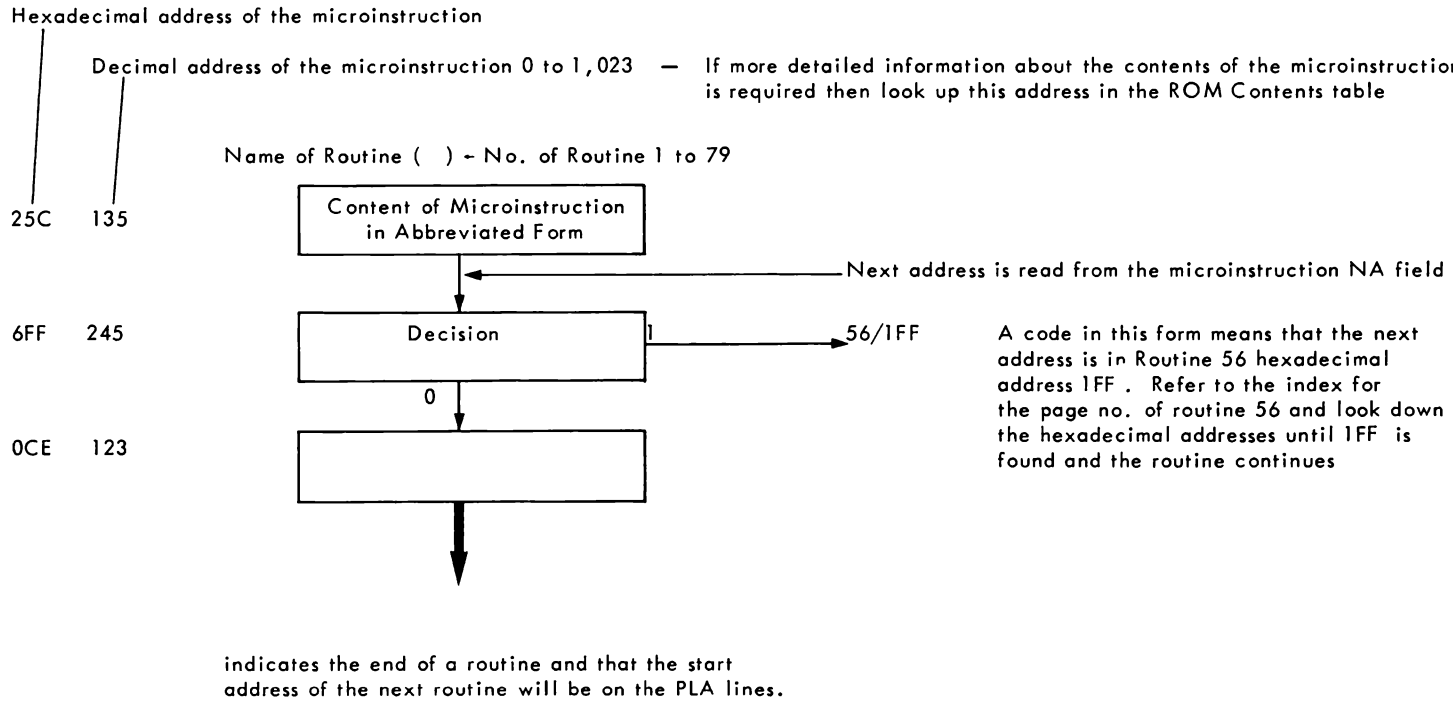
Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	831	0100010000	11	11100	110 0000	0110 1111	1110 000	1
340	832	1000010010	11	01101	000 1011	1100 1011	1100 000	1
	833	1001000001	11	11100	000 1000	0100 1111	1100 000	1
	834	1000100000	10	11001	000 1010	1111 1011	1100 001	1
	835	0100101111	11	11110	010 1000	1001 1111	1011 000	1
	836	0000000000	00	11100	000 1000	0110 0100	1110 000	1
	837	0001001000	10	11100	000 1000	1101 1111	1100 101	1
	838	0001110100	11	01110	000 1000	1001 1111	1011 000	1
	839	1011111100	11	11100	000 1000	1111 1111	0111 000	1
	840	0000010000	10	11010	000 1010	1110 1011	1100 001	1
	841	1000010001	11	11100	000 1010	0110 1011	1100 000	1
34A	842	0011101010	11	11100	000 1000	1111 1111	0000 000	1
	843	0100011011	11	11100	000 1000	1111 0101	0110 000	1
	844	0000000000	00	11100	000 1000	0101 1111	1110 000	1
	845	0101111001	11	11100	000 1000	0111 0100	1100 000	1
	846	0111101011	11	11100	000 1000	0110 0101	0110 000	1
	847	0000000000	00	11100	000 1000	1101 1011	1011 000	1
350	848	1000011010	11	01101	000 1011	1100 1111	1100 000	1
	849	0001000010	10	11100	000 1000	0100 1111	1100 001	1
	850	1000011001	11	01101	000 1011	1111 1111	1100 000	1
	851	1001000000	11	11100	000 1000	0100 1111	1100 000	1
	852	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	853	0001000110	10	11100	000 1000	1101 1111	1011 101	1
	854	1001110100	11	01110	000 1000	1001 1111	1011 000	1
	855	0010000111	11	11100	000 1000	0110 0100	1100 000	1
	856	0000010000	10	11001	000 1010	1110 1111	1100 001	1
	857	1000101101	11	11100	000 1000	1111 1111	1011 000	1
35A	858	0110111001	11	11100	000 1000	1110 1001	1100 000	1
	859	0100111010	10	11100	000 1000	1101 1111	1100 011	1
	860	0101000011	11	11000	000 0000	0110 1111	1101 000	1
	861	0011101010	11	11100	000 1000	1111 1111	0000 000	1
	862	0000000000	00	11100	000 1000	1111 0000	1100 000	1
	863	0101000010	11	00010	000 0010	1111 1111	0000 000	1
360	864	1000010010	11	01101	000 1011	1100 1001	1100 000	1
	865	0000000000	00	11100	000 1000	0001 1111	1110 000	1
	866	1000100000	10	11001	000 1010	1111 1001	1100 001	1
	867	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	868	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	869	1001110010	10	10010	000 1000	1111 0110	0110 010	1
	870	1001110101	11	01110	000 1000	1001 1111	1011 000	1
	871	1100101100	11	11100	000 1000	1101 0001	1011 110	1
	872	0000010000	10	11001	000 1010	1110 1001	1100 001	1
	873	0100101101	11	11100	000 1000	0110 1111	1010 000	1
36A	874	0100110111	11	11100	000 1000	1001 0001	1011 001	1
	875	1100011011	11	11100	000 1000	0000 0000	0000 000	1
	876	1101101010	10	11100	000 1000	1111 1111	0101 100	1
	877	1101101010	11	11100	000 1000	1111 1111	0101 000	1
	878	0000000000	00	11100	000 1000	1101 0101	0110 000	1
	879	0000000000	00	11100	000 1000	1111 1111	0110 000	1
370	880	1000100011	11	01110	000 1000	1101 1111	1100 000	1
	881	0101101101	11	01110	000 1000	0110 1111	1100 000	1
	882	0100001110	11	01110	000 1000	1101 1111	1011 110	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	883	0101100000	11	01110	000 0000	1111 1111	1000 000	1
	884	0000000000	00	11100	000 1000	1111 1111	0000 000	1
	885	1001110010	10	10010	000 1000	1011 0101	0110 010	1
	886	0001110101	11	01110	000 1000	1001 1111	1011 000	1
	887	0000000000	00	11100	000 1000	0101 1111	1100 000	1
	888	0111100110	11	11100	000 0000	1111 1111	0001 000	1
	889	0111100110	11	11100	000 0000	1111 1111	0001 000	1
37A	890	0010010001	11	10111	000 1000	0101 0100	1100 000	1
	891	0000000000	00	11100	000 1000	0101 1111	1100 010	1
	892	0101111100	11	11010	000 0000	1110 1101	1010 000	1
	893	0111110000	11	11100	000 1000	1110 1100	1100 000	1
	894	1101000111	11	11100	000 1000	0000 0000	0000 000	1
	895	0000000000	00	11100	000 1000	0101 0011	1101 000	1
380	896	1010000000	11	11100	000 1010	1101 1111	1111 000	1
	897	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	898	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	899	1000010111	11	11100	000 1000	0111 1111	1011 000	1
	900	1000000011	11	01101	000 1011	1001 1111	1001 000	1
	901	0000010101	11	11100	000 1010	0110 1111	1001 000	1
	902	1000110100	11	11100	000 1000	1111 1111	0000 000	1
	903	1100011010	11	11100	000 1000	1110 0110	0110 010	1
	904	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	905	1000010111	11	11100	000 1000	0110 1111	1011 001	1
38A	906	0000000000	00	11100	000 1000	1111 1111	0011 110	1
	907	0000000000	00	11100	000 1000	0101 1111	1100 000	1
	908	0100111110	11	11100	000 1000	1111 1111	0000 000	1
	909	0101111010	11	11100	000 1000	1111 1111	0000 000	1
	910	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	911	0000000000	00	11100	000 1000	1111 1111	0110 000	1
390	912	1010010000	11	11100	000 1000	1111 1111	0111 000	1
	913	1000000000	00	11100	000 1000	1111 1111	0111 000	1
	914	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	915	1010110111	11	11100	000 1000	0111 1111	1100 000	1
	916	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	917	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	918	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	919	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	920	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	921	1010110111	11	11100	000 1000	0110 1111	1100 000	1
39A	922	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	923	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	924	1101101001	11	11111	111 1010	1110 0110	0110 010	1
	925	0000000000	00	11100	000 1000	1110 0110	0110 010	1
	926	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	927	0000000000	00	11100	000 1000	1111 1111	0000 000	1
3A0	928	0000100000	10	11100	000 1010	1000 1101	1011 011	1
	929	0000100000	10	11100	000 1010	1000 1101	1011 011	1
	930	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	931	0011000011	11	11100	000 1000	0111 1111	1100 000	1
	932	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	933	0000000000	00	11100	000 1000	0000 0000	0000 000	1
	934	0110001001	11	11010	000 0000	1111 1111	0001 000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	935	0110100000	11	11100	000	0000	1111	1111	0001	000	1
	936	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	937	0010100101	11	11100	000	1000	0101	0011	1100	000	1
3AA	938	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	939	1010010100	11	11100	000	1000	1001	0010	1100	000	1
	940	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	941	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	942	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	943	0000000000	00	11100	000	1000	0000	0000	0000	000	1
3B0	944	0000100000	10	11100	000	1010	1000	1100	1011	011	1
	945	0000100000	10	11100	000	1010	1000	1100	1011	011	1
	946	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	947	1010100101	11	11100	000	1000	0111	1111	1100	000	1
	948	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	949	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	950	0110100011	11	11100	011	0000	1111	1111	0001	000	1
	951	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	952	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	953	1010000111	11	11100	000	1000	0101	0011	1100	110	1
3BA	954	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	955	0010110110	11	11100	000	1000	1001	0010	1100	000	1
	956	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	957	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	958	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	959	0000000000	00	11100	000	1000	1111	1111	0000	000	1
3C0	960	0000000000	00	11100	000	1000	1111	1111	0100	000	1
	961	0000000000	00	11100	000	1000	1111	1111	1000	000	1
	962	0110110011	11	11100	000	1010	0111	1111	1011	000	1
	963	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	964	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	965	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	966	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	967	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	968	0001001110	10	11100	000	1000	0000	1111	1100	001	1
	969	0000000000	00	11100	000	1000	0000	0000	0000	000	1
3CA	970	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	971	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	972	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	973	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	974	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	975	0111011000	11	11100	110	0000	1111	1111	0110	000	1
3D0	976	1000000000	00	11100	000	1000	0000	0000	0000	000	1
	977	0001010110	10	11100	000	1000	0000	0000	0000	010	1
	978	1000100001	11	00001	000	1010	1111	1100	1100	000	1
	979	1000100001	11	00001	000	1010	1011	1100	1100	000	1
	980	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	981	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	982	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	983	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	984	1001101110	10	00001	000	1010	0010	1100	1100	001	1
	985	1000100001	11	00001	000	1010	1010	1100	1100	000	1
3DA	986	0000000000	00	11100	000	1000	0000	0000	0000	000	1

Address		NA	NAR	RS	Control Signals	RCONT	TEST	BME
Hexa	Decimal							
	987	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	988	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	989	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	990	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	991	0000000000	00	11100	000	1000	0000	0000	0000	000	1
3E0	992	0010000000	11	11100	000	1010	1101	1111	1111	010	1
	993	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	994	0011011100	11	11100	000	1010	0111	1111	1100	000	1
	995	1001100001	11	11100	000	1000	0111	1111	1100	000	1
	996	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	997	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	998	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	999	0011101110	11	11100	000	1010	0111	1111	1100	000	1
	1000	1001110000	11	01110	000	1000	1101	1111	1011	000	1
	1001	1001100000	10	11100	000	1000	0110	1111	1100	001	1
3EA	1002	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1003	1000101111	11	11100	000	1000	0111	0011	1011	000	1
	1004	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1005	1001110000	11	01110	000	1000	1101	1111	1011	000	1
	1006	0000000000	00	11100	000	1000	0110	1111	1100	000	1
	1007	1000010000	11	01101	000	1011	0110	1111	1100	000	1
3F0	1008	0010010000	11	11100	000	1010	1111	1111	0111	110	1
	1009	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1010	0000001000	11	11100	000	1010	0111	1111	1100	000	1
	1011	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1012	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1013	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1014	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1015	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1016	1001001101	11	11100	000	1000	0011	0000	1100	000	1
	1017	1001001101	11	11100	000	1000	0011	0000	1100	000	1
3FA	1018	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1019	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1020	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1021	0000000000	00	11100	000	1000	0000	0000	0000	000	1
	1022	0000000000	00	11100	000	1000	0110	1111	1100	000	1
3FF	1023	0000011111	11	11100	000	1000	0110	1111	1100	000	1

5.92 Cross Referencing between Microinstruction Routines



Microinstruction Abbreviations

- , separates each mnemonic
- . AND
- ⊕ OR
- \+ plus
- → represents data flow e.g. BUS → A means data from the Bus to Spalu Register A.

Planet Registers K, IR, ENB, FU, CR, PLR.

Spalu Registers A, AX, RX, SP(A0-A15), P, R1, R2, ALU.

T — the letter T before a mnemonic means that a condition is being tested. Test mode is selected with NAR1=1, and NAR2=0.

Plok — an internal Planet signal selects mode of operation. See table under paragraph heading Planet.

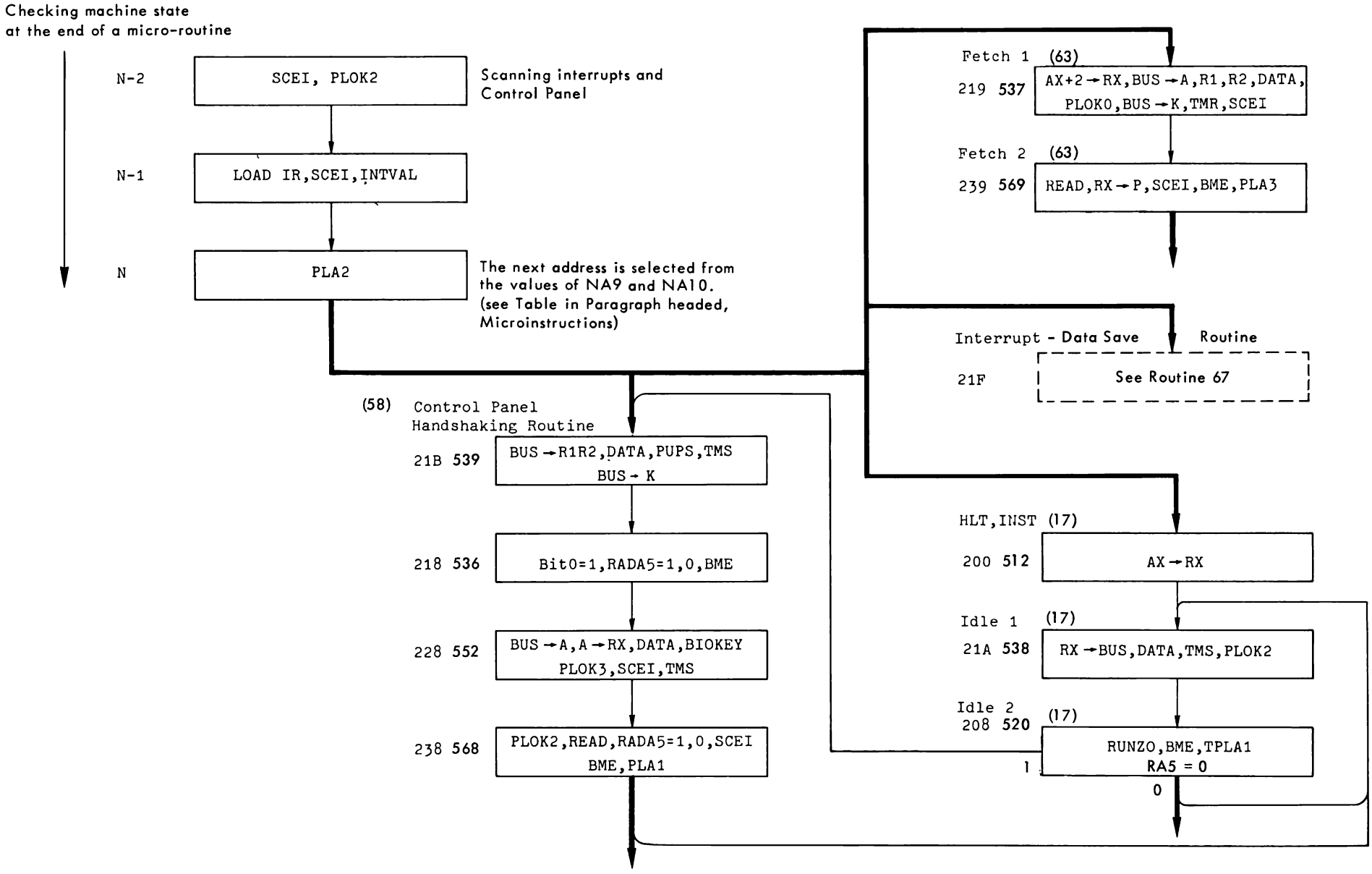
( ) — The code contained in the brackets gives the value of the RCONT field. For the precise function see paragraph, Addressing ROM Bits 24-39.

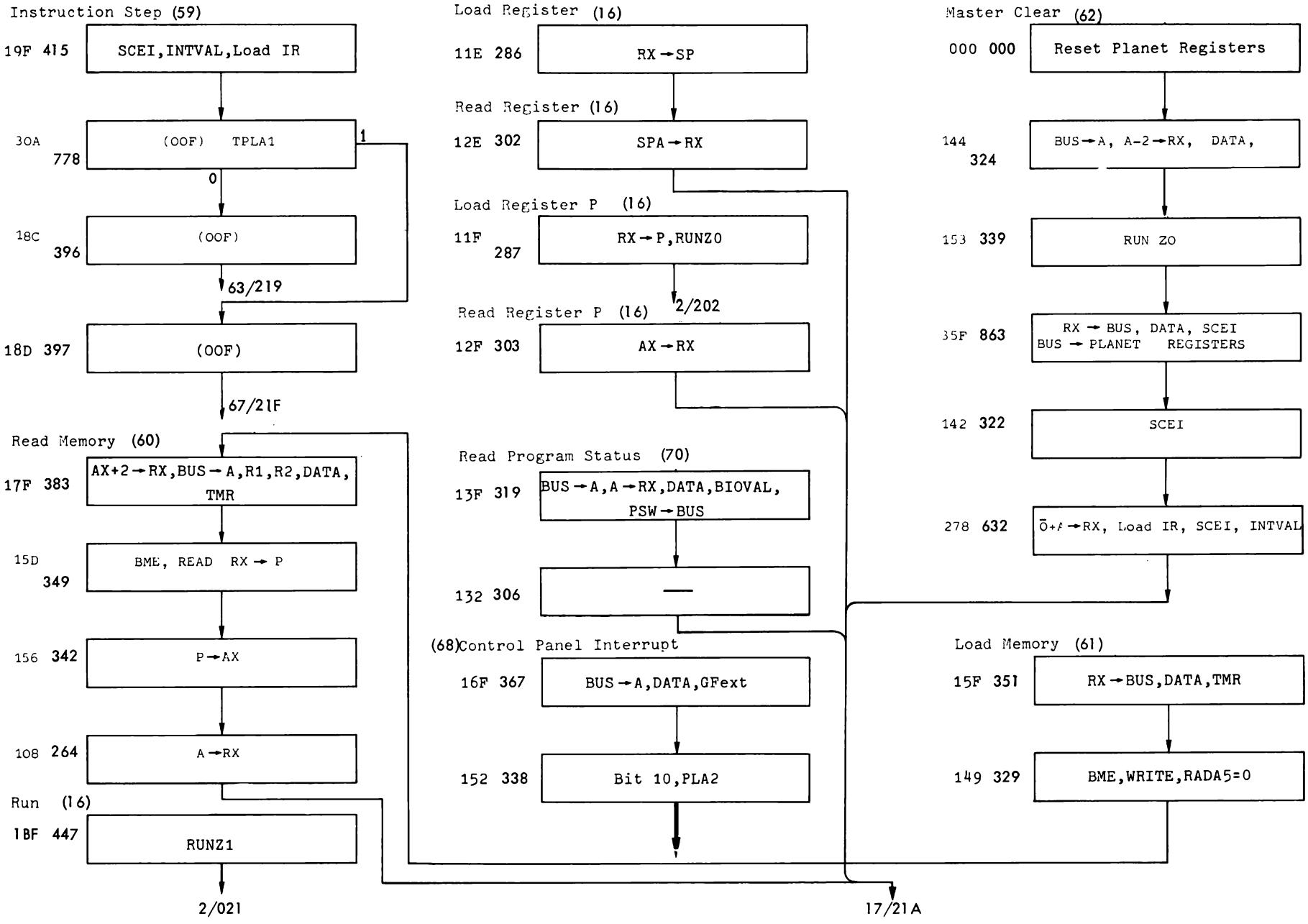
RX $\overset{R}{\rightarrow}$ SP — 1 bit shifted right between registers.

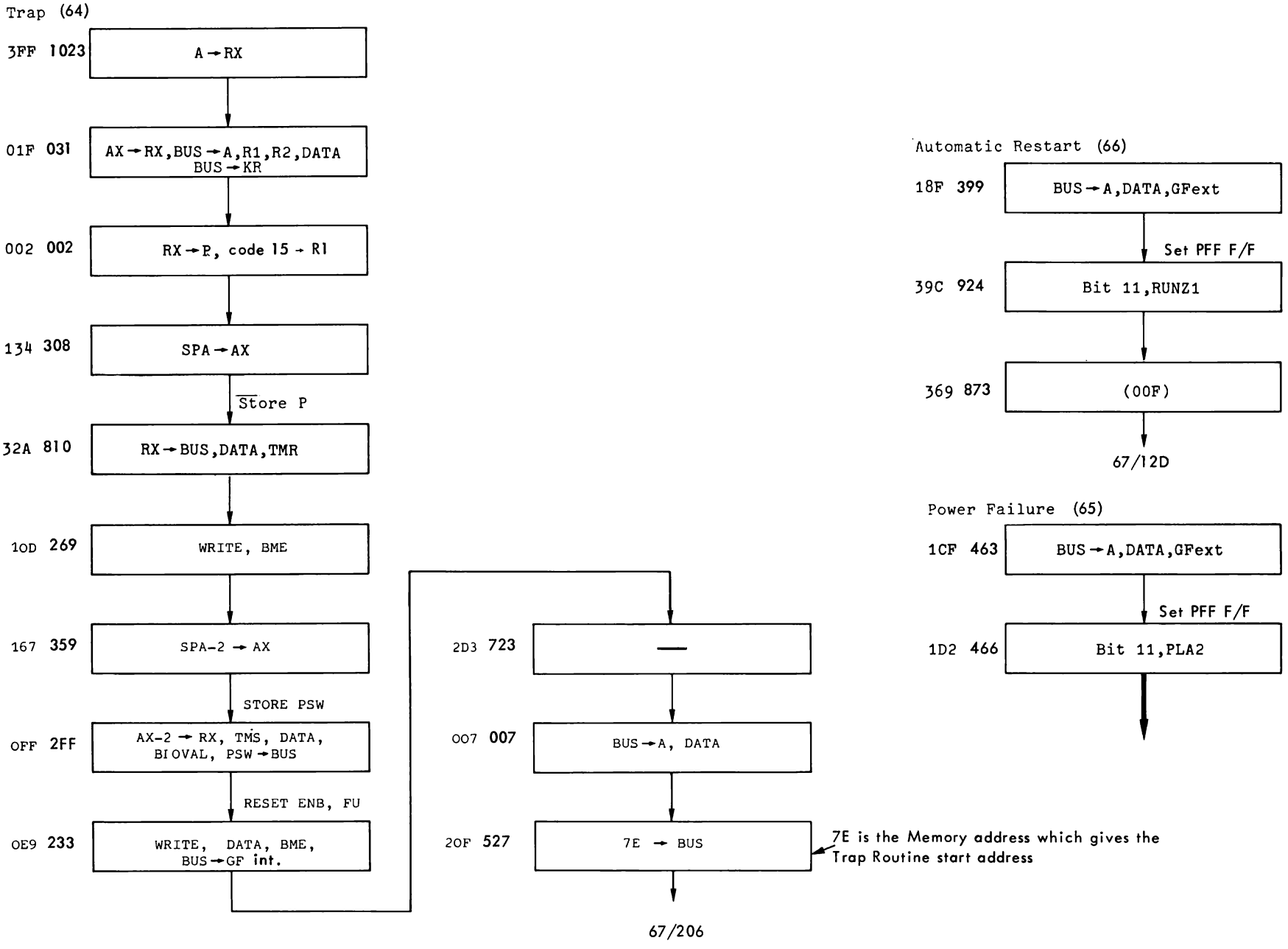
OR1 — 0 → Register R1, OR2 — 0 → Register R2.

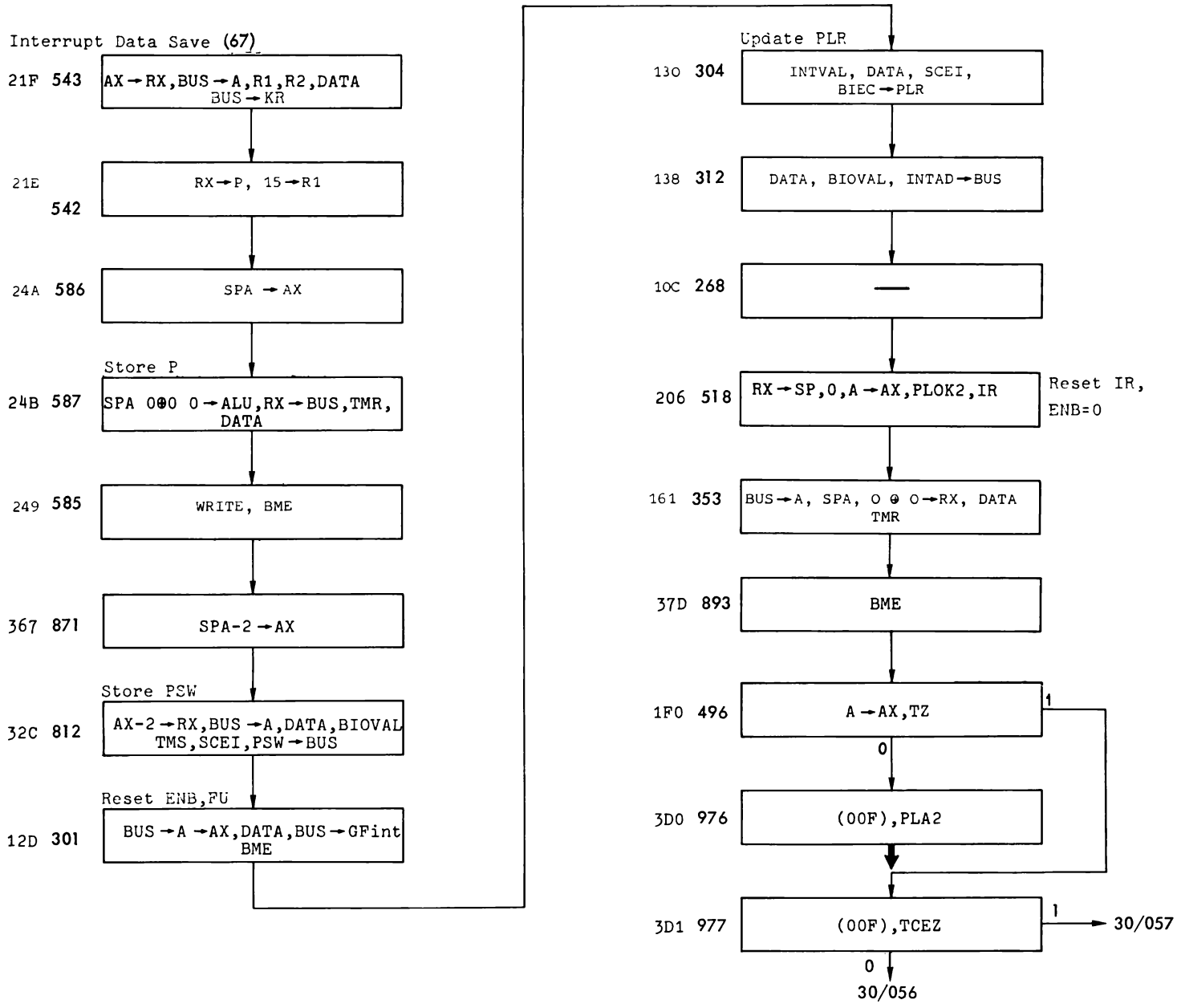
For other abbreviations see end of this section.

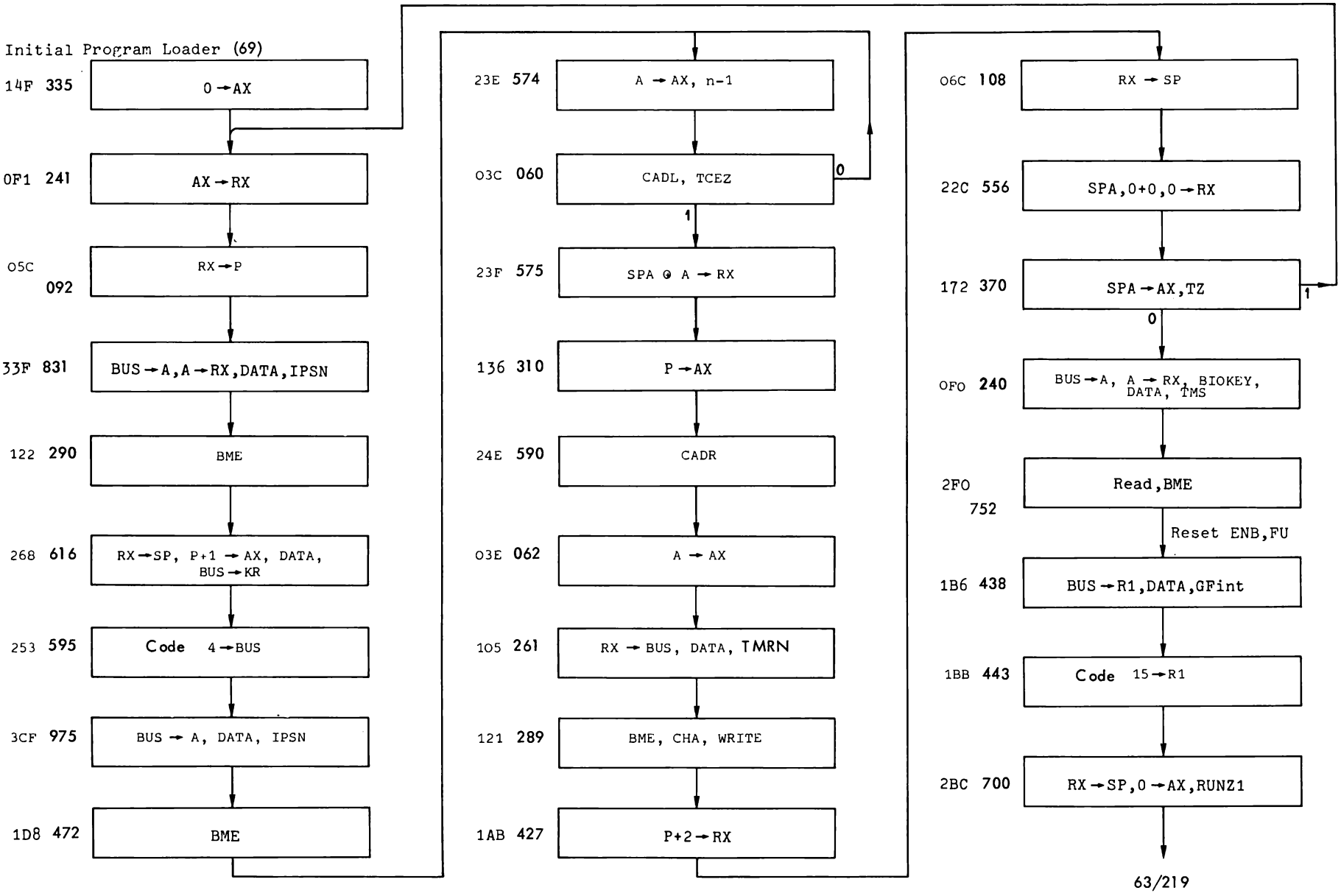
Figure 5-25 Microinstruction Routines 1-79

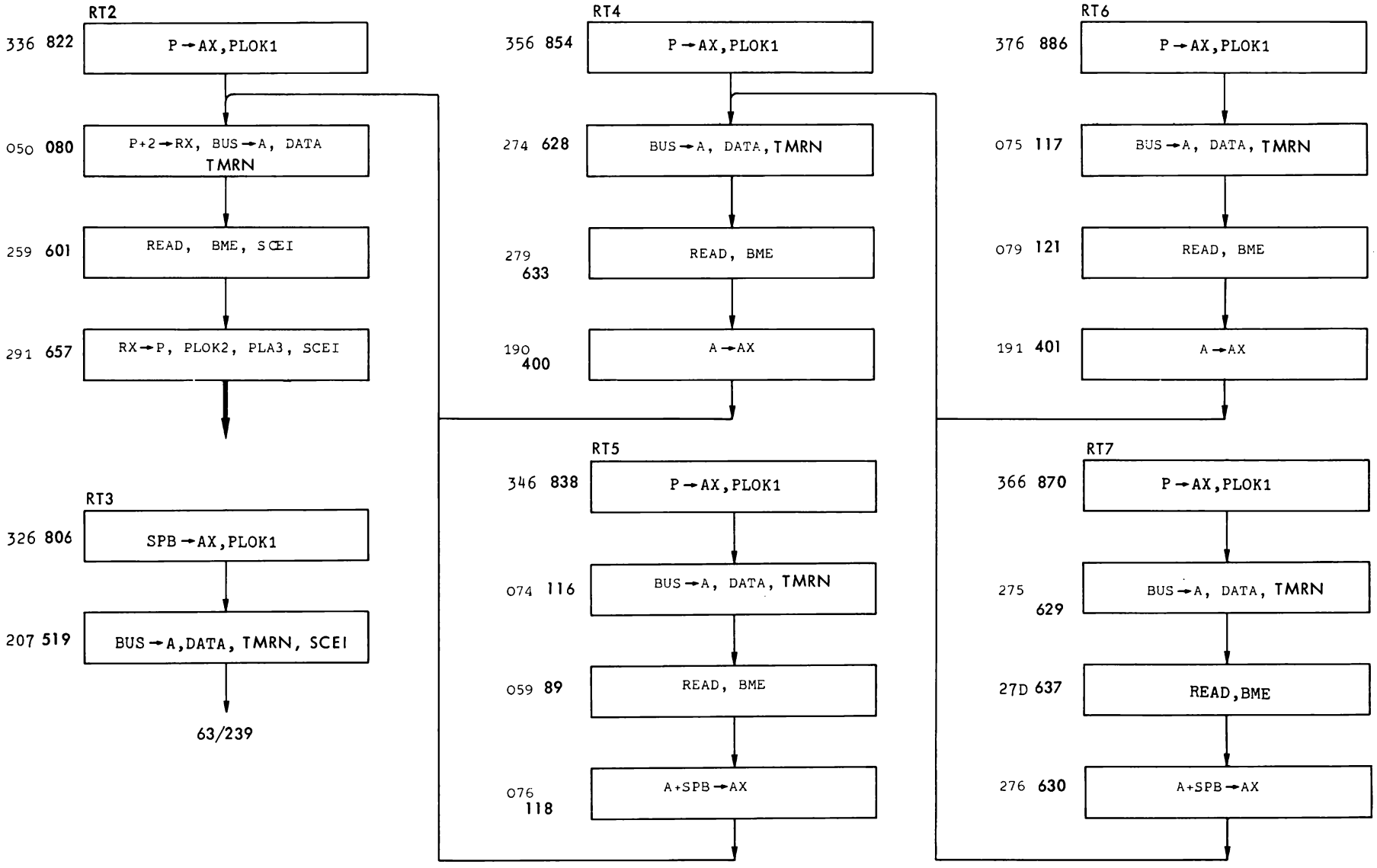



Instruction Step (59)
19F 415
SCEI,INTVAL,Load IR
30A 778
(OOF) TPLA1
1
0
18C 396
(OOF)
63/219
18D 397
(OOF)
67/21F
Read Memory (60)
17F 383
AX+2→RX,BUS→A,R1,R2,DATA, TMR
15D 349
BME, READ RX→P
156 342
P→AX
108 264
A→RX
Run (16)
1BF 447
RUNZ1
2/021
Load Register (16)
11E 286
RX→SP
Read Register (16)
12E 302
SPA→RX
Load Register P (16)
11F 287
RX→P,RUNZ0
2/202
Read Register P (16)
12F 303
AX→RX
Read Program Status (70)
13F 319
BUS→A,A→RX,DATA,BIOVAL, PSW→BUS
132 306
(68)Control Panel Interrupt
16F 367
BUS→A,DATA,GFext
152 338
Bit 10,PLA2
17/21A
Master Clear (62)
000 000
Reset Planet Registers
144 324
BUS→A, A-2→RX, DATA,
153 339
RUN ZO
35F 863
RX→BUS, DATA, SCEI
BUS→PLANET REGISTERS
142 322
SCEI
278 632
Load Memory (61)
15F 351
RX→BUS,DATA,TMR
149 329
BME,WRITE,RADA5=0

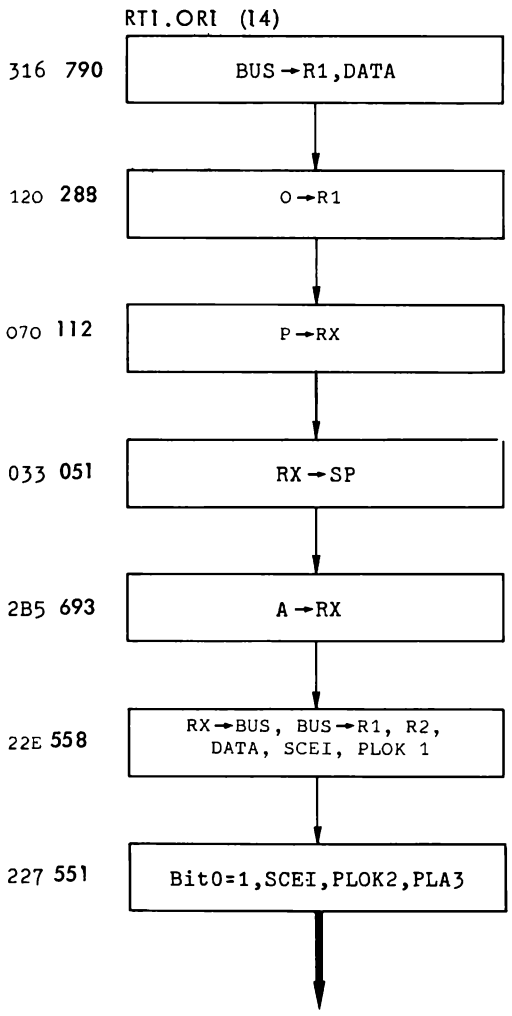

Trap (64)
3FF 1023
A → RX
01F 031
AX → RX, BUS → A, R1, R2, DATA
BUS → KR
002 002
RX → P, code 15 → R1
134 308
SPA → AX
Store P
32A 810
RX → BUS, DATA, TMR
10D 269
WRITE, BME
167 359
SPA-2 → AX
STORE PSW
0FF 2FF
AX-2 → RX, TMS, DATA,
BI OVAL, PSW → BUS
RESET ENB, FU
0E9 233
WRITE, DATA, BME,
BUS → GF int.
2D3 723
—
007 007
BUS → A, DATA
20F 527
7E → BUS
7E is the Memory address which gives the
Trap Routine start address
67/206
Automatic Restart (66)
18F 399
BUS → A, DATA, GFext
Set PFF F/F
39C 924
Bit 11, RUNZ1
369 873
(00F)
67/12D
Power Failure (65)
1CF 463
BUS → A, DATA, GFext
Set PFF F/F
1D2 466
Bit 11, PLA2


Interrupt Data Save (67)
21F 543
AX→RX,BUS→A,R1,R2,DATA
BUS→KR
21E 542
RX→P, 15→R1
24A 586
SPA →AX
Store P
24B 587
SPA 0⊕0 0→ALU,RX→BUS,TMR,
DATA
249 585
WRITE, BME
367 871
SPA-2 →AX
Store PSW
32C 812
AX-2→RX,BUS →A,DATA,BIOVAL
TMS,SCEI,PSW→BUS
Reset ENB,FU
12D 301
BUS→A →AX,DATA,BUS→GFint
BME
Update PLR
130 304
INTVAL, DATA, SCEI,
BIEC→PLR
138 312
DATA, BIOVAL, INTAD→BUS
10C 268
206 518
RX→SP,0,A→AX,PLOK2,IR
Reset IR,
ENB=0
161 353
BUS→A, SPA, O ⊕ O→RX, DATA
TMR
37D 893
BME
1F0 496
A→AX,TZ
1
0
3D0 976
(00F),PLA2
3D1 977
(00F),TCEZ
1
30/057
0
30/056

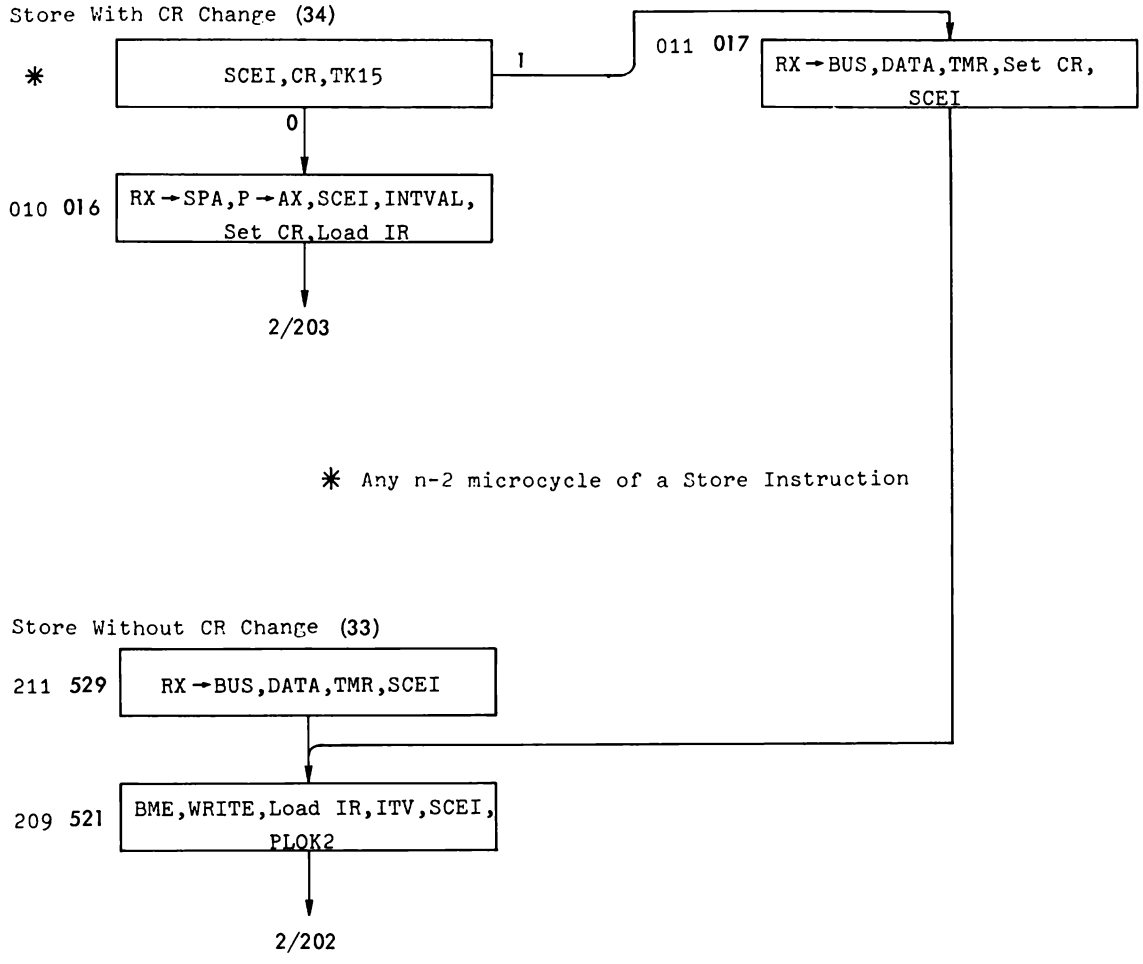

Initial Program Loader (69)
14F 335
0 → AX
0F1 241
AX → RX
05C 092
RX → P
33F 831
BUS → A, A → RX, DATA, IPSN
122 290
BME
268 616
RX → SP, P+1 → AX, DATA, BUS → KR
253 595
Code 4 → BUS
3CF 975
BUS → A, DATA, IPSN
1D8 472
BME
23E 574
A → AX, n-1
03C 060
CADL, TCEZ
0
1
23F 575
SPA ⊕ A → RX
136 310
P → AX
24E 590
CADR
03E 062
A → AX
105 261
RX → BUS, DATA, TMRN
121 289
BME, CHA, WRITE
1AB 427
P+2 → RX
06C 108
RX → SP
22C 556
SPA, 0+0, 0 → RX
172 370
SPA → AX, TZ
1
0
0F0 240
BUS → A, A → RX, BIOKEY, DATA, TMS
2F0 752
Read, BME
Reset ENB, FU
1B6 438
BUS → R1, DATA, GFint
1BB 443
Code 15 → R1
2BC 700
RX → SP, 0 → AX, RUNZ1
63/219

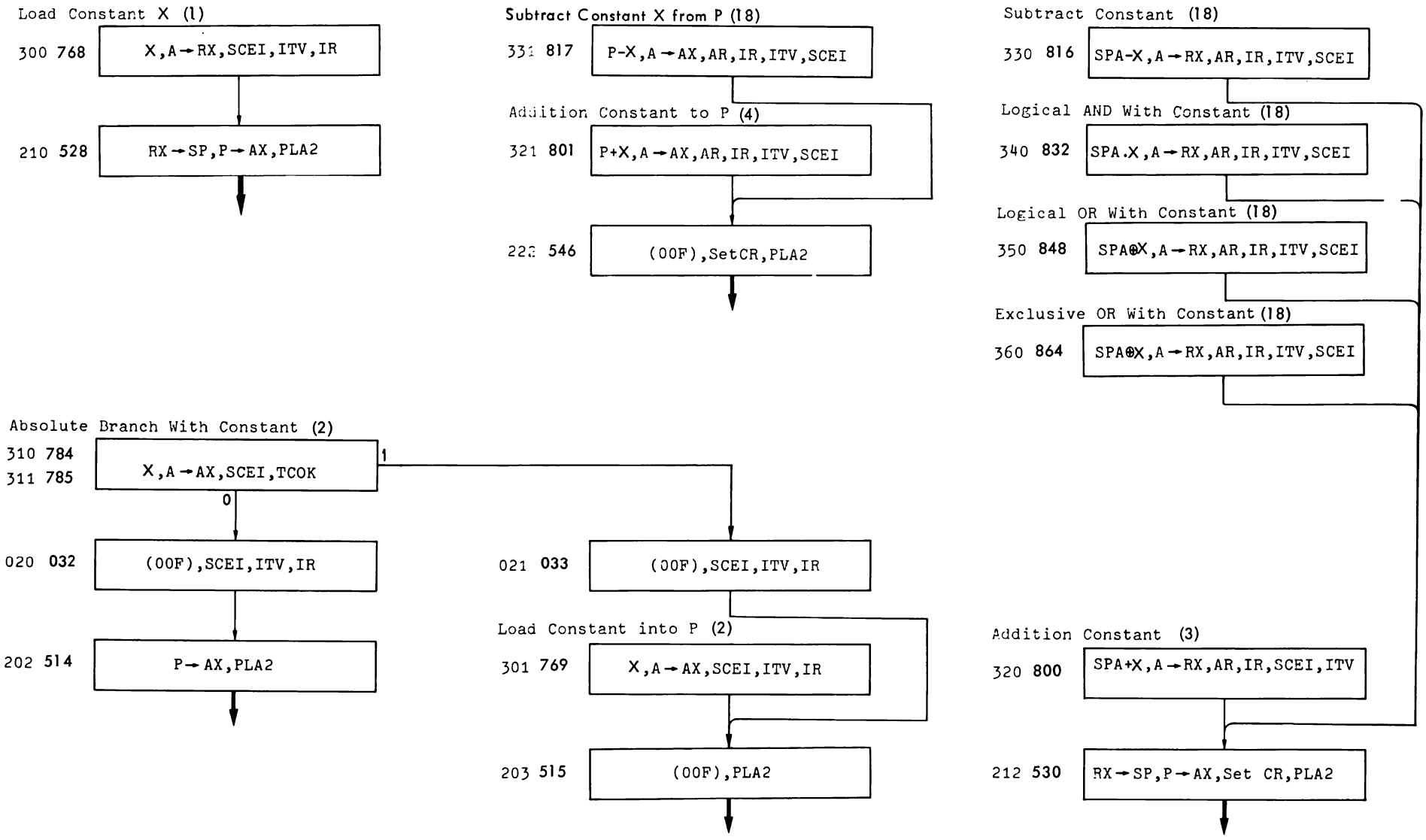

RT2
336 822
P→AX,PLOK1
050 080
P+2→RX, BUS→A, DATA
TMRN
259 601
READ, BME, SCEI
291 657
RX→P, PLOK2, PLA3, SCEI
RT3
326 806
SPB→AX,PLOK1
207 519
BUS→A,DATA, TMRN, SCEI
63/239
RT4
356 854
P→AX,PLOK1
274 628
BUS→A, DATA,TMRN
279
633
READ, BME
190
400
A→AX
RT5
346 838
P→AX,PLOK1
074 116
BUS→A, DATA, TMRN
059 89
READ, BME
076
118
A+SPB→AX
RT6
376 886
P→AX,PLOK1
075 117
BUS→A, DATA, TMRN
079 121
READ, BME
191 401
A→AX
RT7
366 870
P→AX,PLOK1
275
629
BUS→A, DATA, TMRN
27D 637
READ,BME
276 630
A+SPB→AX

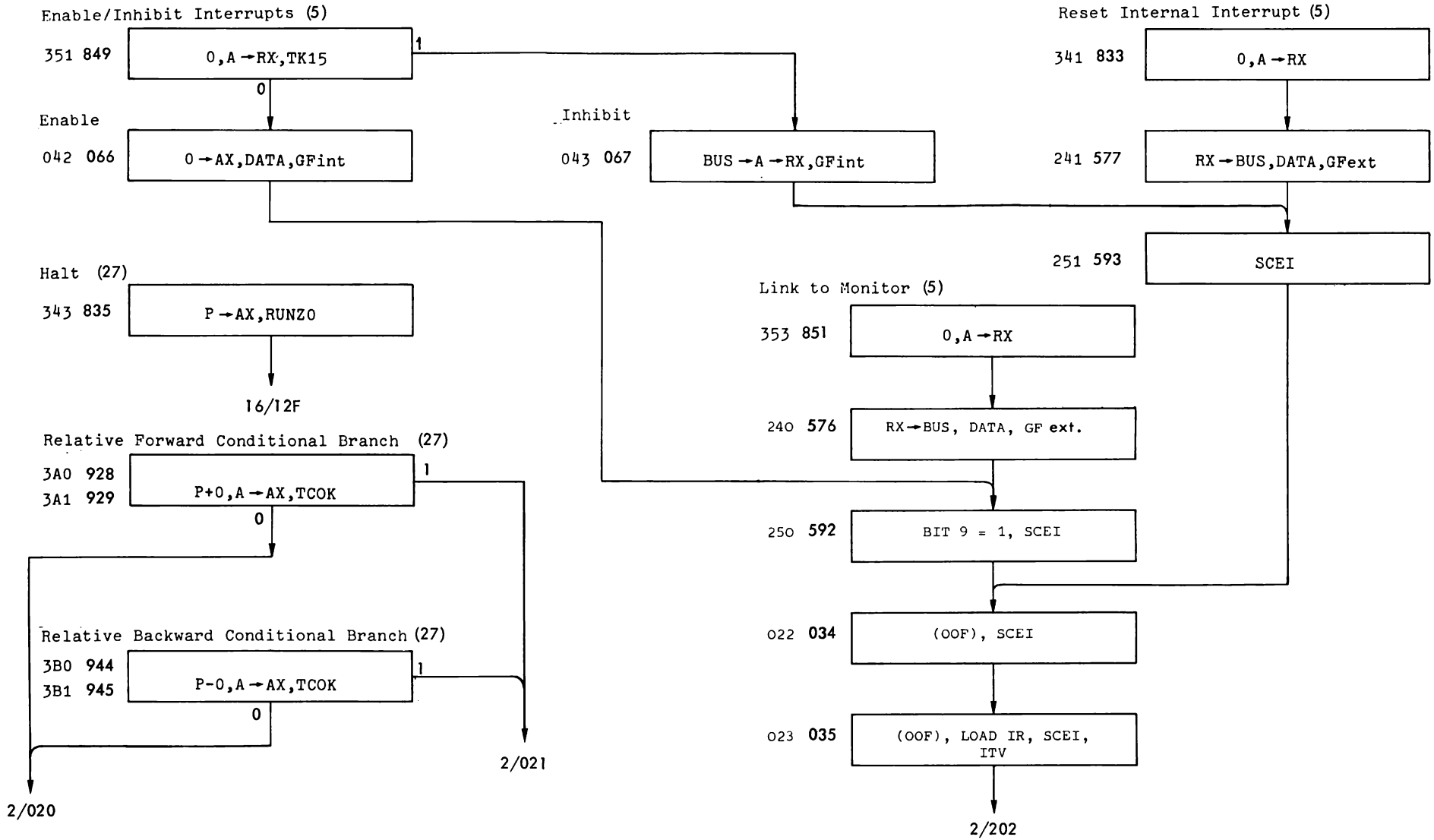
Addressing Routines contd. (23)

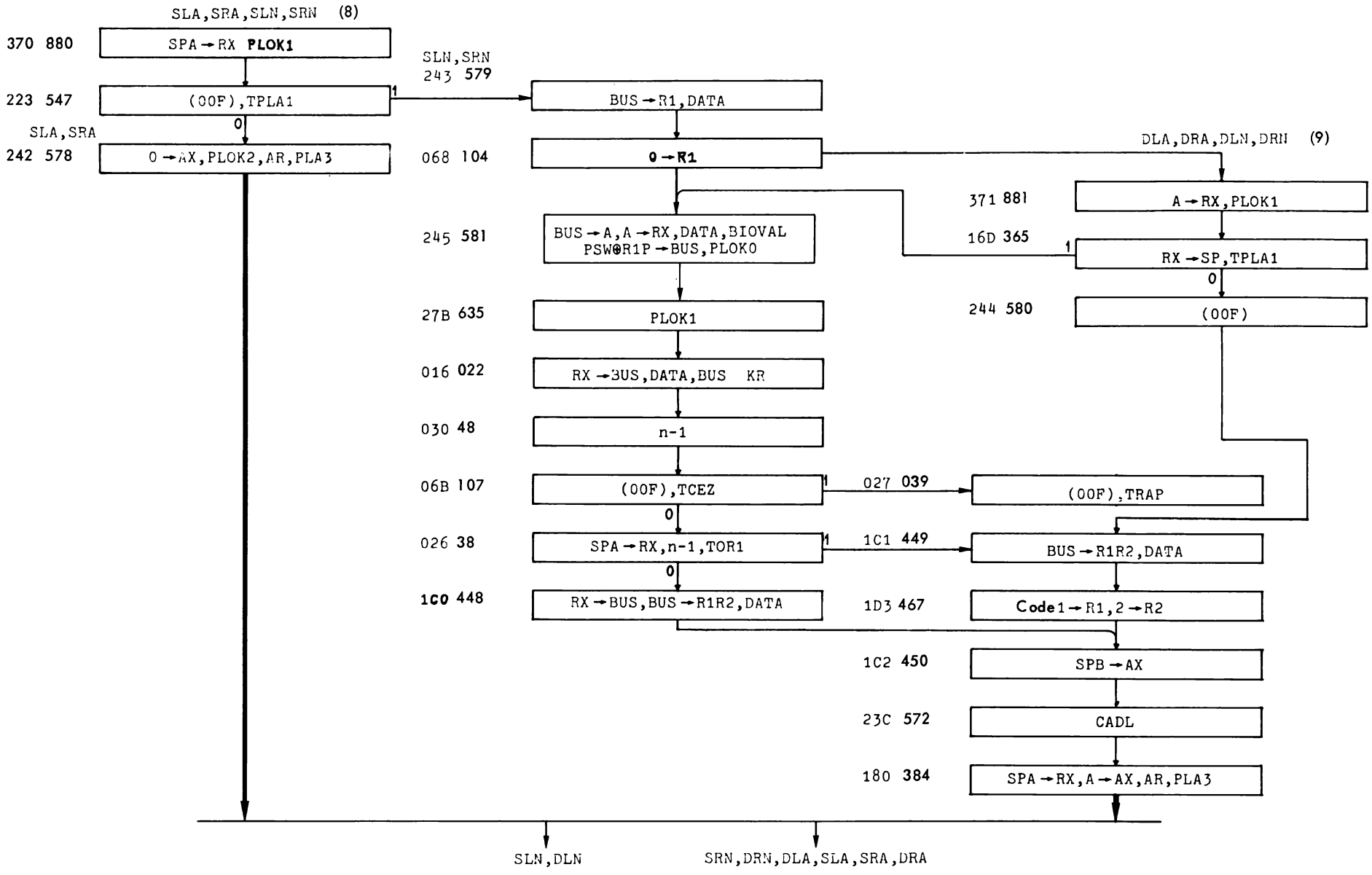


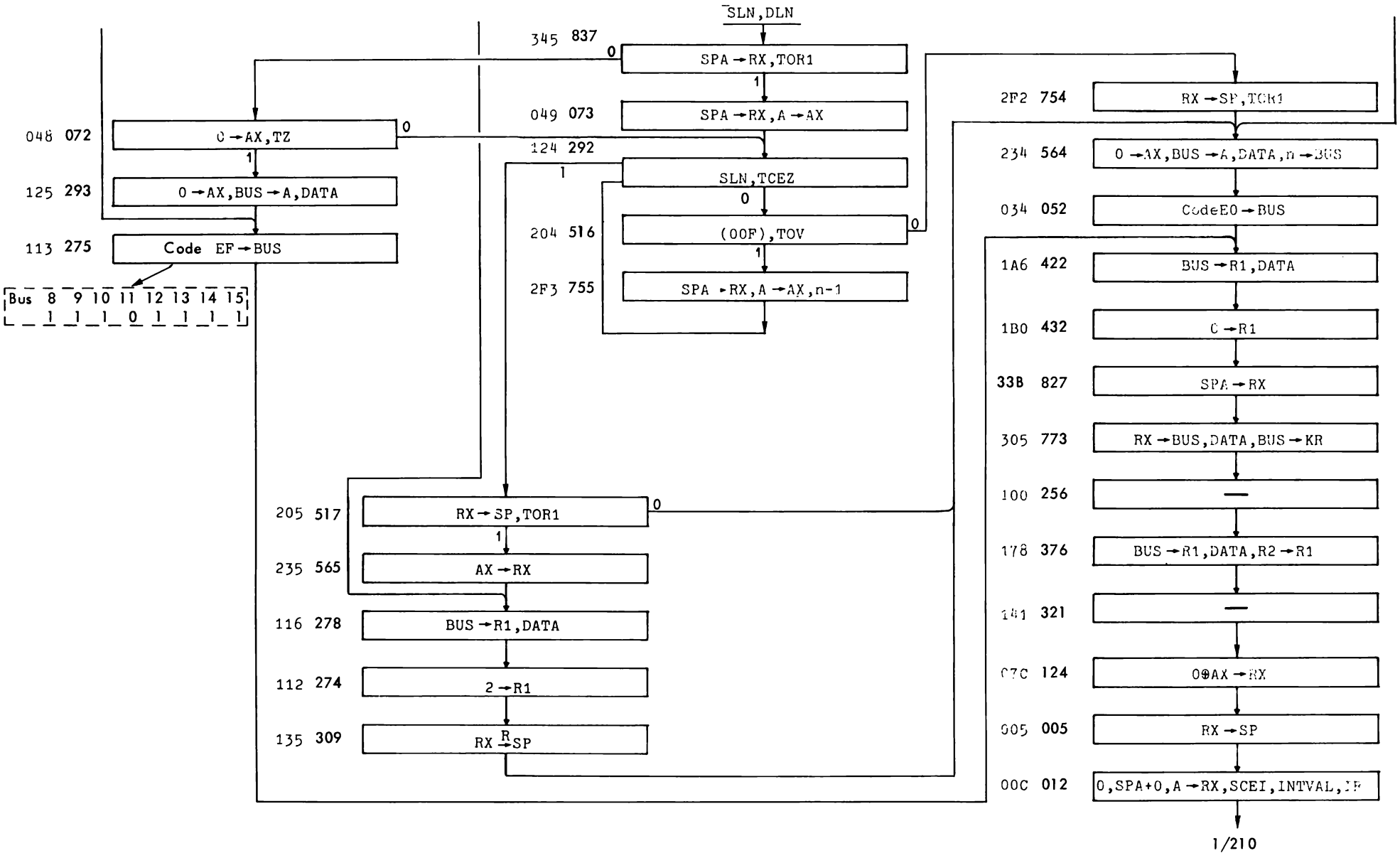
Store Routines (33 and 34)

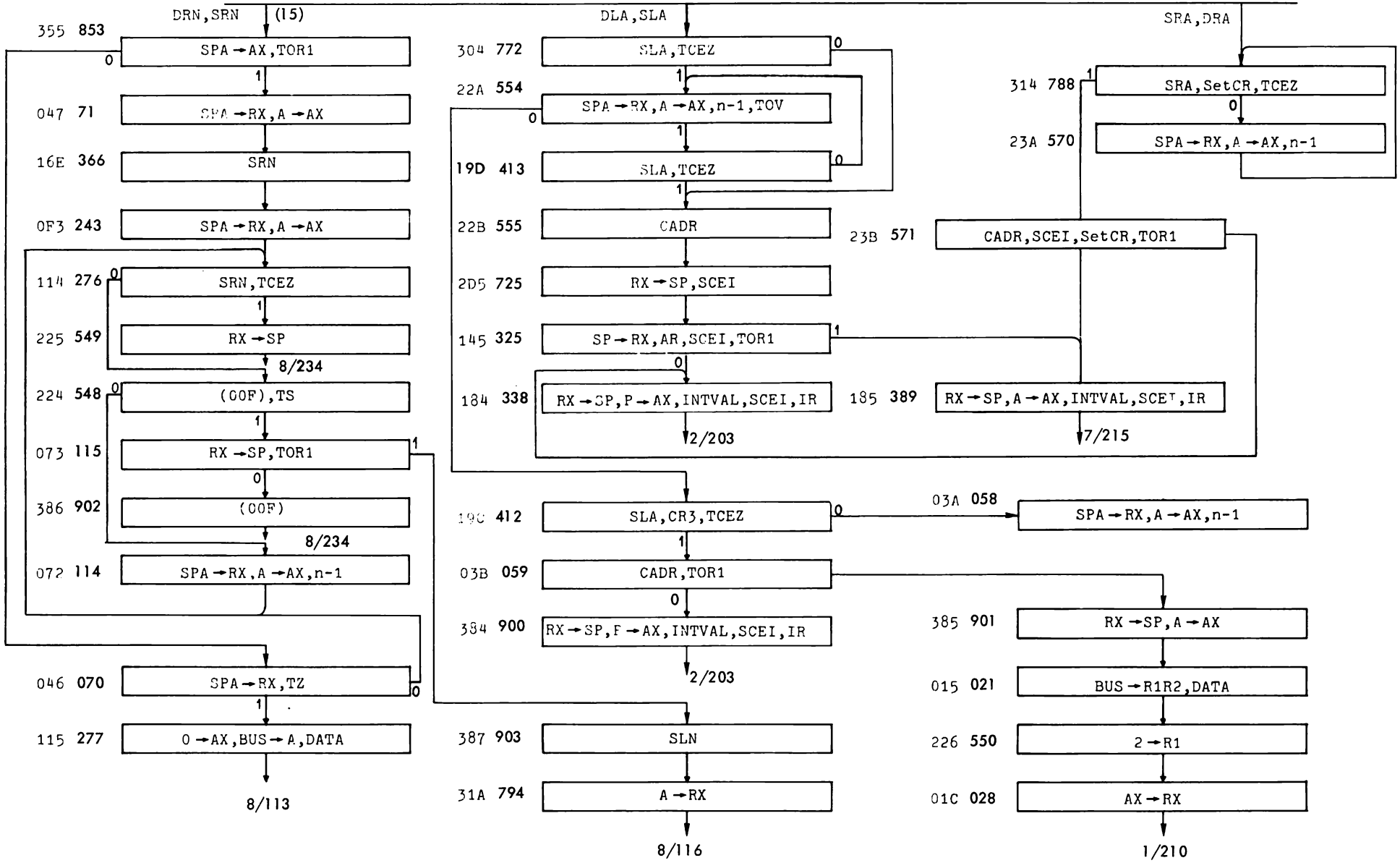


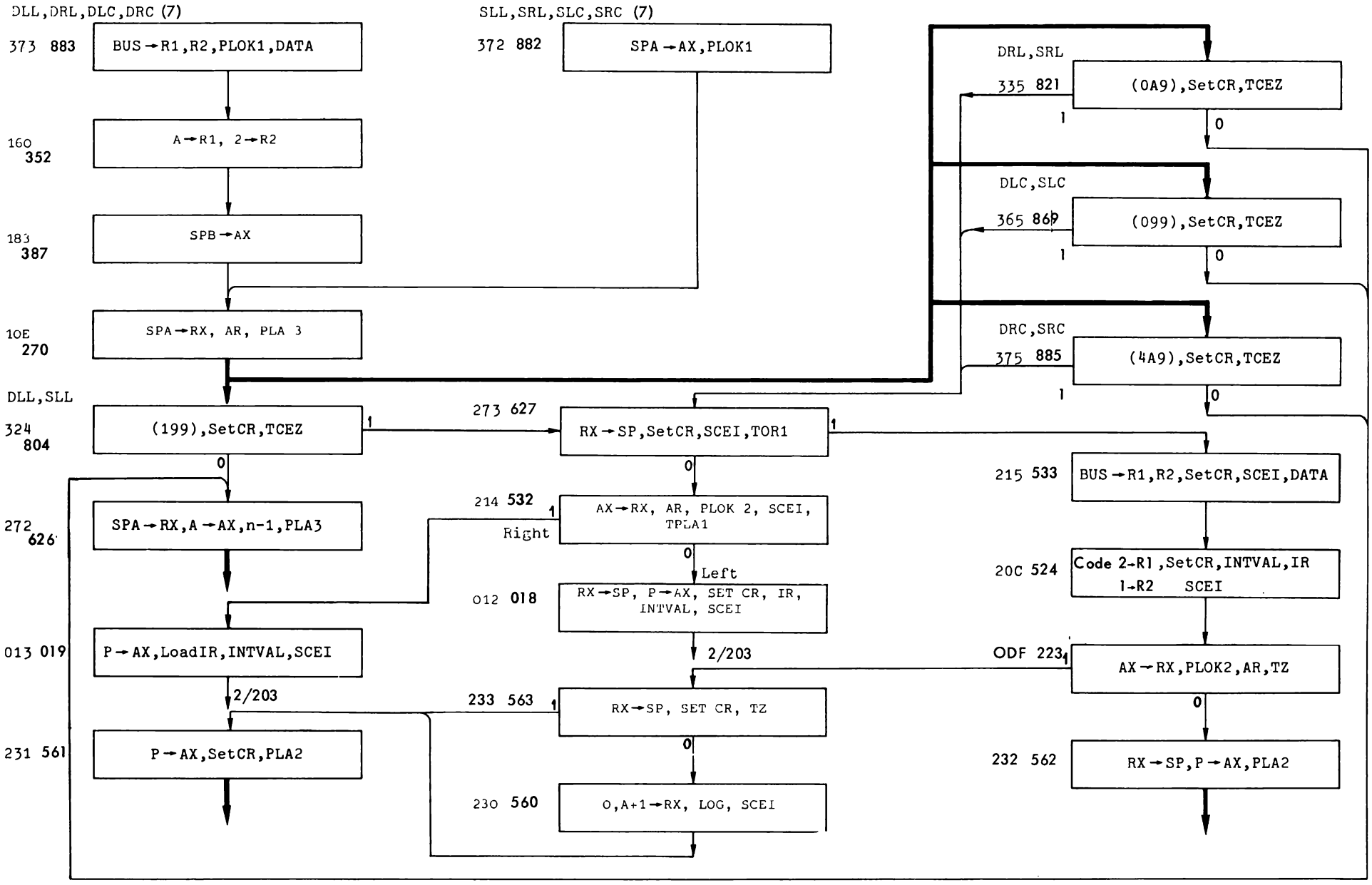

Load Constant X (1)
300 768
X,A→RX,SCEI,ITV,IR
210 528
RX→SP,P→AX,PLA2
Subtract Constant X from P (18)
331 817
P-X,A→AX,AR,IR,ITV,SCEI
Addition Constant to P (4)
321 801
P+X,A→AX,AR,IR,ITV,SCEI
223 546
(00F),SetCR,PLA2
Subtract Constant (18)
330 816
SPA-X,A→RX,AR,IR,ITV,SCEI
Logical AND With Constant (18)
340 832
SPA.X,A→RX,AR,IR,ITV,SCEI
Logical OR With Constant (18)
350 848
SPA⊕X,A→RX,AR,IR,ITV,SCEI
Exclusive OR With Constant (18)
360 864
SPA⊕X,A→RX,AR,IR,ITV,SCEI
Absolute Branch With Constant (2)
310 784
311 785
X,A→AX,SCEI,TCOK
1
0
020 032
(00F),SCEI,ITV,IR
202 514
P→AX,PLA2
021 033
(00F),SCEI,ITV,IR
Load Constant into P (2)
301 769
X,A→AX,SCEI,ITV,IR
203 515
(00F),PLA2
Addition Constant (3)
320 800
SPA+X,A→RX,AR,IR,SCEI,ITV
212 530
RX→SP,P→AX,Set CR,PLA2

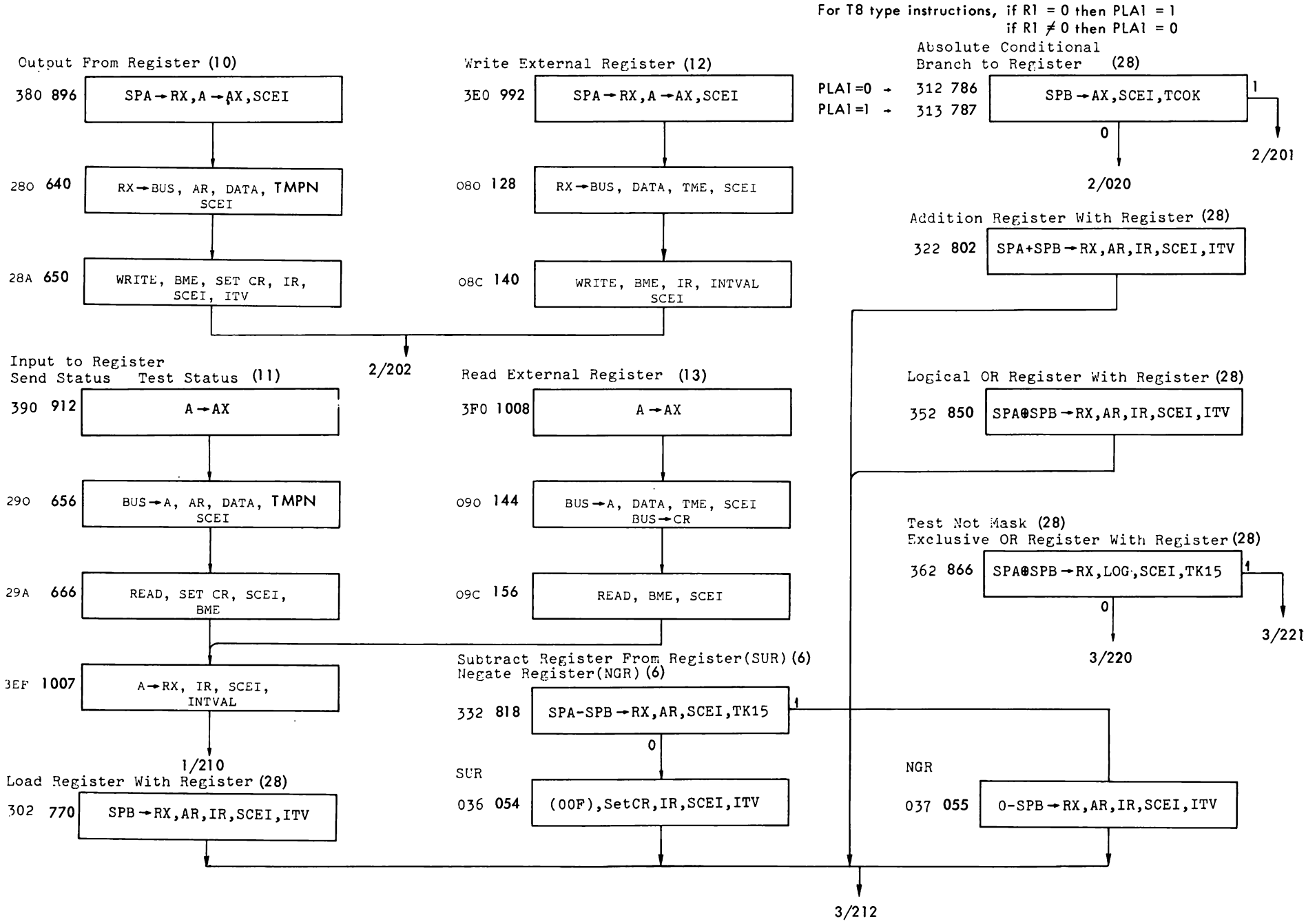

Enable/Inhibit Interrupts (5)
351 849
0,A → RX,TK15
1
0
Enable
042 066
0 → AX,DATA,GFint
Inhibit
043 067
BUS → A → RX,GFint
Reset Internal Interrupt (5)
341 833
0,A → RX
241 577
RX → BUS,DATA,GFext
251 593
SCEI
Halt (27)
343 835
P → AX,RUNZ0
16/12F
Link to Monitor (5)
353 851
0,A → RX
240 576
RX → BUS, DATA, GF ext.
250 592
BIT 9 = 1, SCEI
022 034
(OOF), SCEI
023 035
(OOF), LOAD IR, SCEI, ITV
2/202
Relative Forward Conditional Branch (27)
3A0 928
3A1 929
P+0,A → AX,TCOK
1
0
2/021
Relative Backward Conditional Branch (27)
3B0 944
3B1 945
P-0,A → AX,TCOK
1
0
2/020

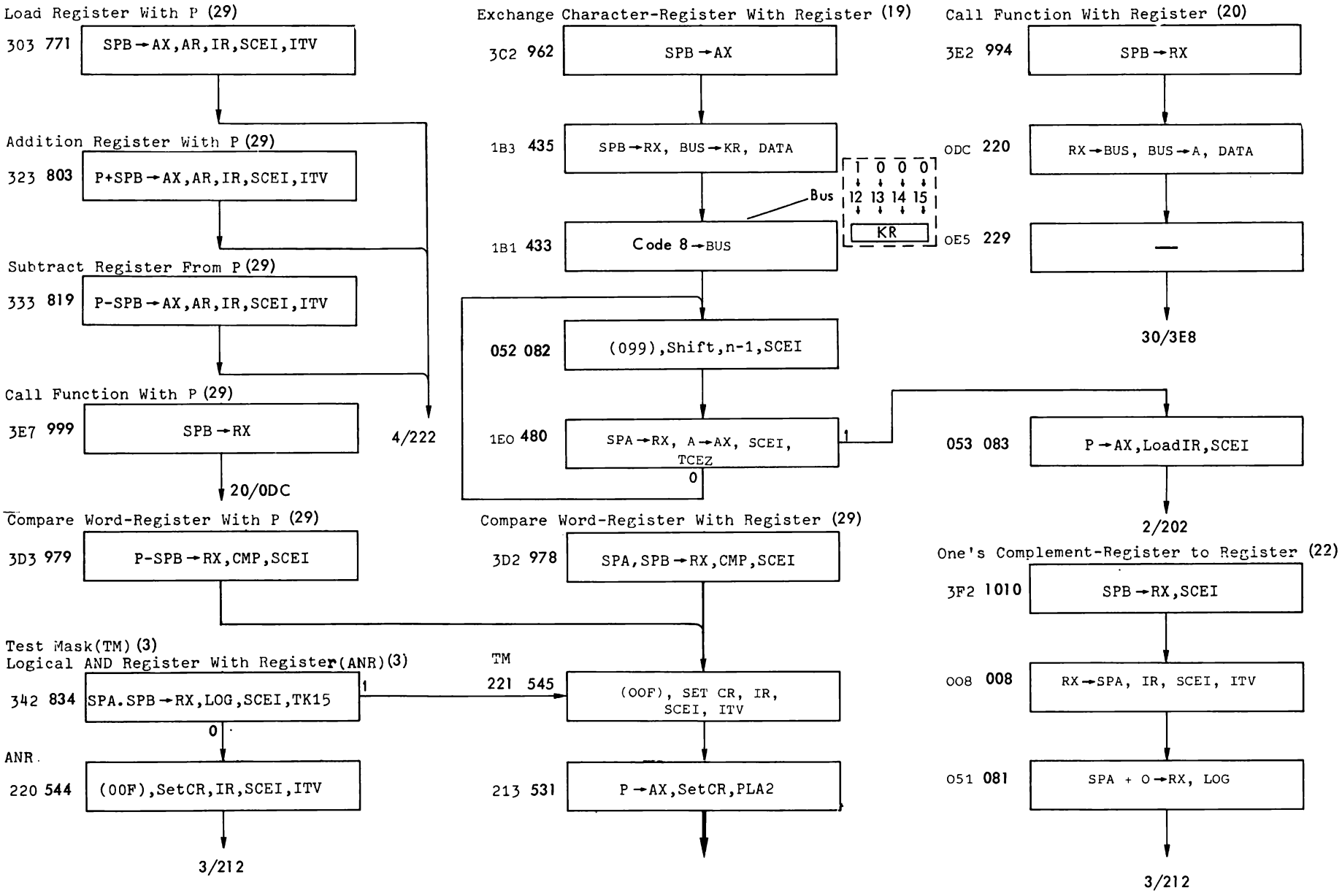

SLA,SRA,SLN,SRN (8)
370 880
SPA→RX PLOK1
223 547
(00F),TPLA1
1
0
SLA,SRA
242 578
0→AX,PLOK2,AR,PLA3
SLN,SRN
243 579
BUS→R1,DATA
068 104
0→R1
245 581
BUS→A,A→RX,DATA,BIOVAL
PSW@R1P→BUS,PLOK0
27B 635
PLOK1
016 022
RX→BUS,DATA,BUS KR
030 48
n-1
06B 107
(00F),TCEZ
1
027 039
(00F),TRAP
0
026 38
SPA→RX,n-1,TOR1
1
1C1 449
BUS→R1R2,DATA
0
1C0 448
RX→BUS,BUS→R1R2,DATA
1D3 467
Code1→R1,2→R2
1C2 450
SPB→AX
23C 572
CADL
180 384
SPA→RX,A→AX,AR,PLA3
DLA,DRA,DLN,DRN (9)
371 881
A→RX,PLOK1
16D 365
1
RX→SP,TPLA1
0
244 580
(00F)
SLN,DLN
SRN,DRN,DLA,SLA,SRA,DRA

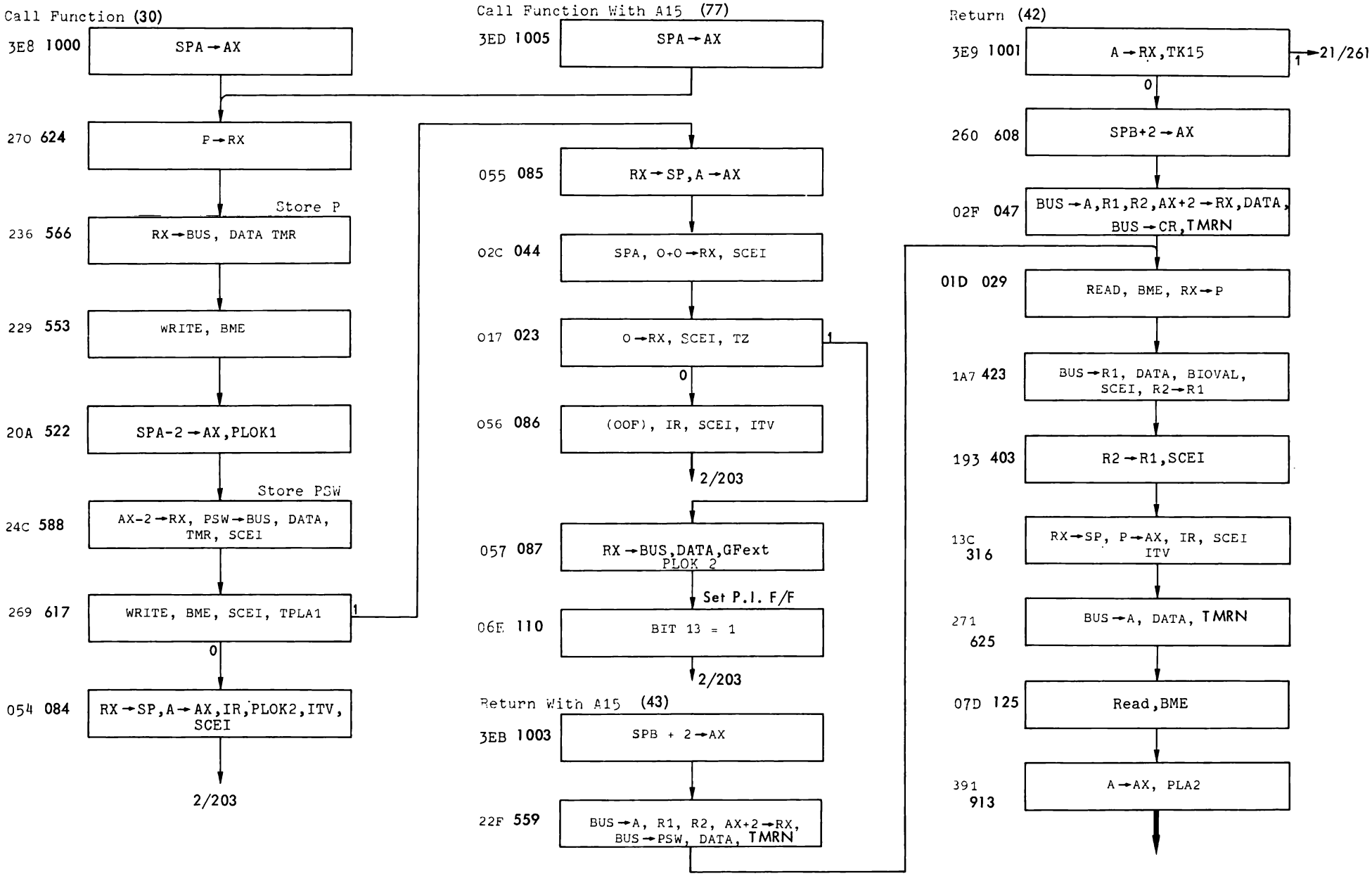

SLN,DLN
345 837
SPA → RX,TOR1
049 073
SPA → RX,A → AX
124 292
SLN,TCEZ
204 516
(00F),TOV
2F3 755
SPA → RX,A → AX,n-1
048 072
C → AX,TZ
125 293
0 → AX,BUS → A,DATA
113 275
Code EF → BUS
Bus 8 9 10 11 12 13 14 15
1 1 1 0 1 1 1 1
205 517
RX → SP,TOR1
235 565
AX → RX
116 278
BUS → R1,DATA
112 274
2 → R1
135 309
RX R→ SP
2F2 754
RX → SP,TOR1
234 564
0 → AX,BUS → A,DATA,n → BUS
034 052
CodeE0 → BUS
1A6 422
BUS → R1,DATA
1B0 432
C → R1
33B 827
SPA → RX
305 773
RX → BUS,DATA,BUS → KR
100 256
178 376
BUS → R1,DATA,R2 → R1
141 321
07C 124
0⊕AX → RX
005 005
RX → SP
00C 012
0,SPA+0,A → RX,SCEI,INTVAL,IE
1/210

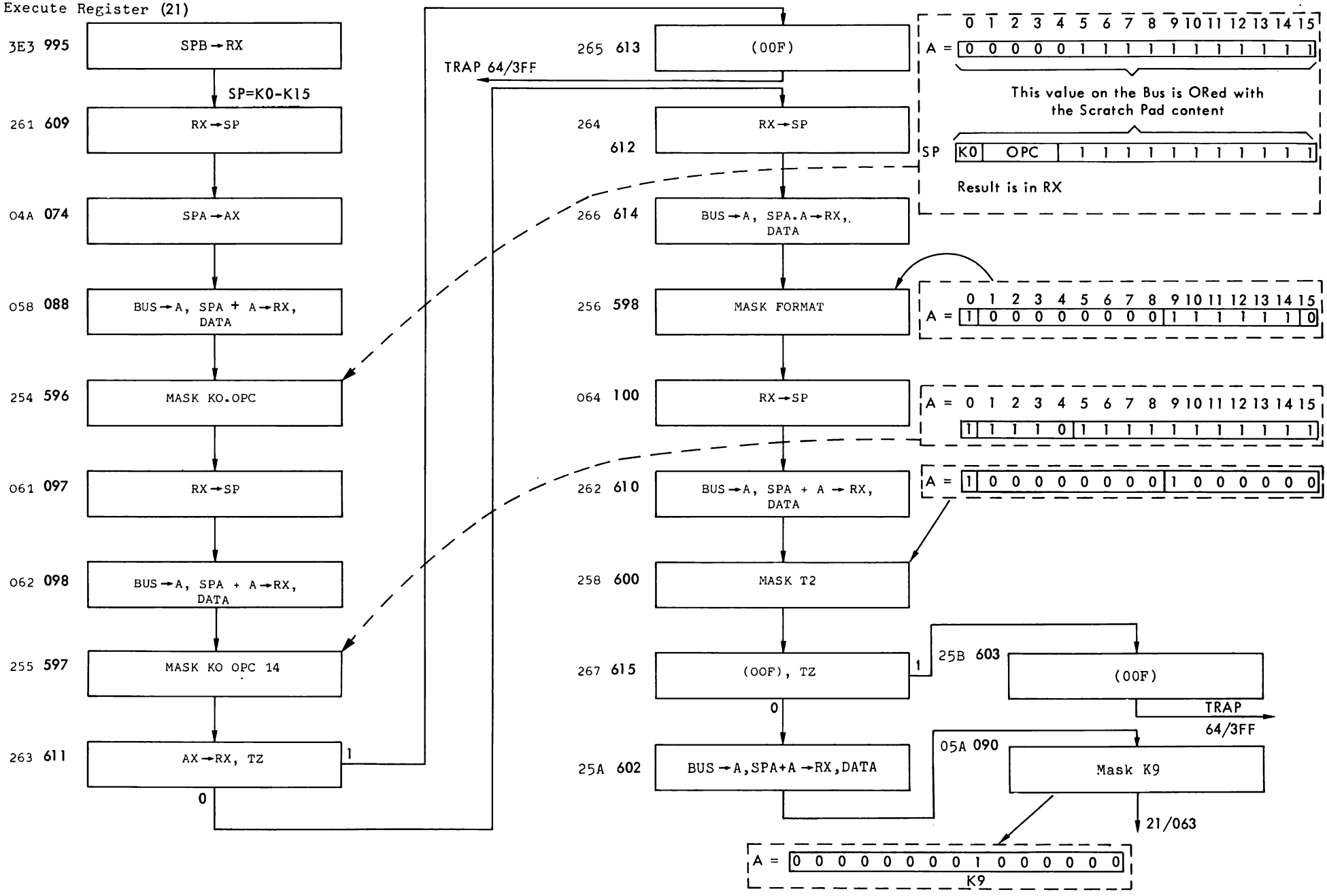

DRN,SRN
(15)
355 853
SPA→AX,TOR1
047 71
SPA→RX,A→AX
16E 366
SRN
0F3 243
SPA→RX,A→AX
114 276
SRN,TCEZ
225 549
RX→SP
8/234
224 548
(00F),TS
073 115
RX→SP,TOR1
386 902
(00F)
8/234
072 114
SPA→RX,A→AX,n-1
046 070
SPA→RX,TZ
115 277
0→AX,BUS→A,DATA
8/113
DLA,SLA
304 772
SLA,TCEZ
22A 554
SPA→RX,A→AX,n-1,TOV
19D 413
SLA,TCEZ
22B 555
CADR
2D5 725
RX→SP,SCEI
145 325
SP→RX,AR,SCEI,TOR1
184 338
RX→SP,P→AX,INTVAL,SCEI,IR
2/203
19C 412
SLA,CR3,TCEZ
03B 059
CADR,TOR1
384 900
RX→SP,F→AX,INTVAL,SCEI,IR
2/203
387 903
SLN
31A 794
A→RX
8/116
SRA,DRA
314 788
SRA,SetCR,TCEZ
23A 570
SPA→RX,A→AX,n-1
23B 571
CADR,SCEI,SetCR,TOR1
185 389
RX→SP,A→AX,INTVAL,SCEI,IR
7/215
03A 058
SPA→RX,A→AX,n-1
385 901
RX→SP,A→AX
015 021
BUS→R1R2,DATA
226 550
2→R1
01C 028
AX→RX
1/210

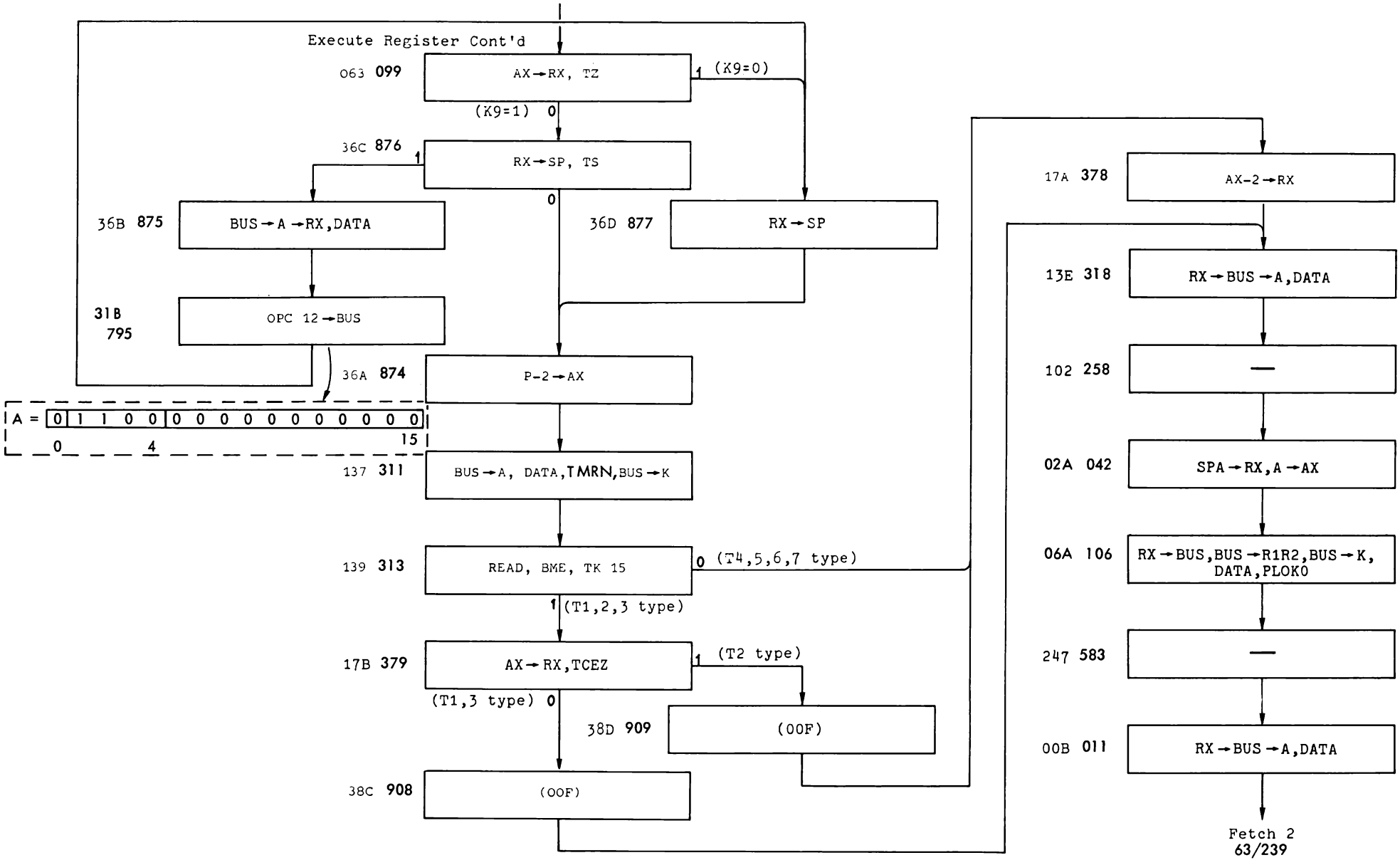

DLL,DRL,DLC,DRC (7)
373 883
BUS→R1,R2,PLOK1,DATA
160 352
A→R1, 2→R2
183 387
SPB→AX
10E 270
SPA→RX, AR, PLA 3
DLL,SLL
324 804
(199),SetCR,TCEZ
1
0
272 626
SPA→RX,A→AX,n-1,PLA3
013 019
P→AX,LoadIR,INTVAL,SCEI
2/203
231 561
P→AX,SetCR,PLA2
SLL,SRL,SLC,SRC (7)
372 882
SPA→AX,PLOK1
273 627
RX→SP,SetCR,SCEI,TOR1
214 532
Right
AX→RX, AR, PLOK 2, SCEI, TPLA1
Left
012 018
RX→SP, P→AX, SET CR, IR, INTVAL, SCEI
2/203
233 563
RX→SP, SET CR, TZ
230 560
0,A+1→RX, LOG, SCEI
DRL,SRL
335 821
(0A9),SetCR,TCEZ
DLC,SLC
365 869
(099),SetCR,TCEZ
DRC,SRC
375 885
(4A9),SetCR,TCEZ
215 533
BUS→R1,R2,SetCR,SCEI,DATA
20C 524
Code 2→R1,SetCR,INTVAL,IR
1→R2 SCEI
ODF 223
AX→RX,PLOK2,AR,TZ
232 562
RX→SP,P→AX,PLA2

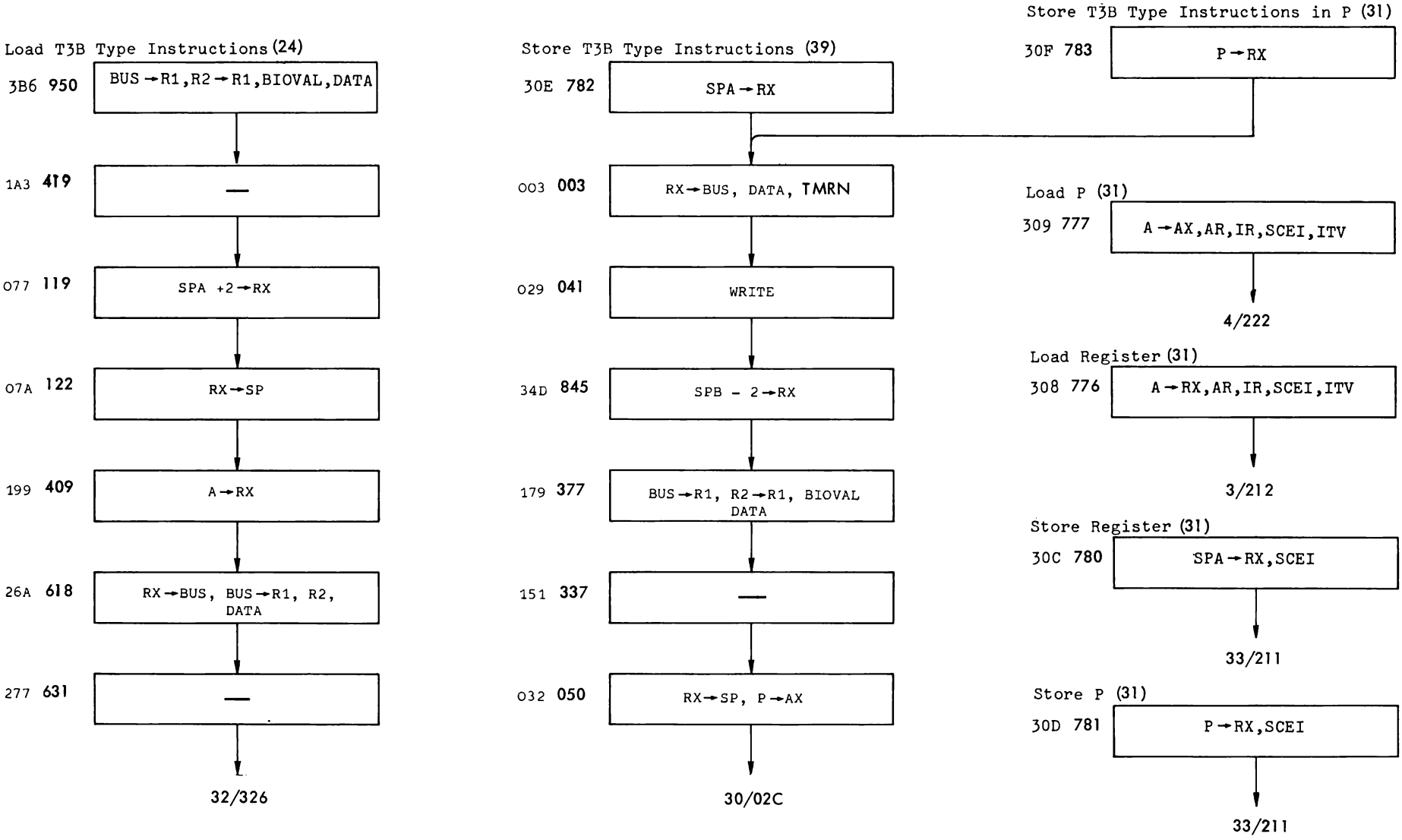

For T8 type instructions, if R1 = 0 then PLA1 = 1
if R1 ≠ 0 then PLA1 = 0
Output From Register (10)
380 896
SPA→RX,A→AX,SCEI
280 640
RX→BUS, AR, DATA, TMPN SCEI
28A 650
WRITE, BME, SET CR, IR, SCEI, ITV
Write External Register (12)
3E0 992
SPA→RX,A→AX,SCEI
080 128
RX→BUS, DATA, TME, SCEI
08C 140
WRITE, BME, IR, INTVAL SCEI
Absolute Conditional Branch to Register (28)
PLA1=0 → 312 786
PLA1=1 → 313 787
SPB→AX,SCEI,TCOK
1
0
2/201
2/020
Addition Register With Register (28)
322 802
SPA+SPB→RX,AR,IR,SCEI,ITV
2/202
Input to Register Send Status Test Status (11)
390 912
A→AX
290 656
BUS→A, AR, DATA, TMPN SCEI
29A 666
READ, SET CR, SCEI, BME
Read External Register (13)
3F0 1008
A→AX
090 144
BUS→A, DATA, TME, SCEI BUS→CR
09C 156
READ, BME, SCEI
Logical OR Register With Register (28)
352 850
SPA⊕SPB→RX,AR,IR,SCEI,ITV
Test Not Mask (28)
Exclusive OR Register With Register (28)
362 866
SPA⊕SPB→RX,LOG,SCEI,TK15
1
0
3/221
3/220
3EF 1007
A→RX, IR, SCEI, INTVAL
Subtract Register From Register(SUR) (6)
Negate Register(NGR) (6)
332 818
SPA-SPB→RX,AR,SCEI,TK15
1
0
1/210
Load Register With Register (28)
302 770
SPB→RX,AR,IR,SCEI,ITV
SUR
036 054
(00F),SetCR,IR,SCEI,ITV
NGR
037 055
0-SPB→RX,AR,IR,SCEI,ITV
3/212

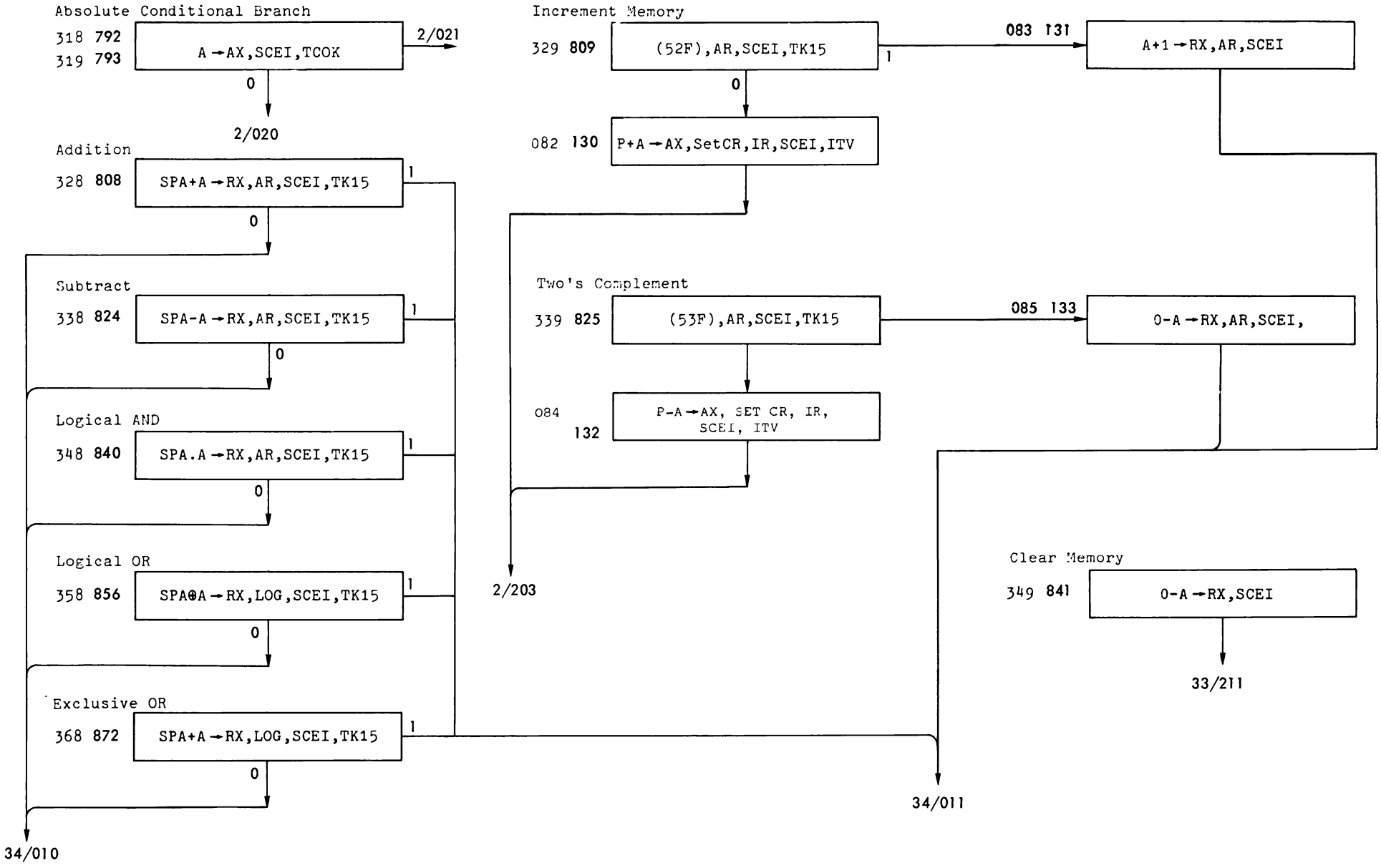

Load Register With P (29)
303 771
SPB→AX,AR,IR,SCEI,ITV
Addition Register With P (29)
323 803
P+SPB→AX,AR,IR,SCEI,ITV
Subtract Register From P (29)
333 819
P-SPB→AX,AR,IR,SCEI,ITV
Call Function With P (29)
3E7 999
SPB→RX
20/0DC
4/222
Compare Word-Register With P (29)
3D3 979
P-SPB→RX,CMP,SCEI
Test Mask(TM) (3)
Logical AND Register With Register(ANR)(3)
342 834
SPA.SPB→RX,LOG,SCEI,TK15
1
0
ANR
220 544
(00F),SetCR,IR,SCEI,ITV
3/212
Exchange Character-Register With Register (19)
3C2 962
SPB→AX
1B3 435
SPB→RX, BUS→KR, DATA
Bus
1 0 0 0
12 13 14 15
KR
1B1 433
Code 8→BUS
052 082
(099),Shift,n-1,SCEI
1E0 480
SPA→RX, A→AX, SCEI, TCEZ
1
0
Compare Word-Register With Register (29)
3D2 978
SPA,SPB→RX,CMP,SCEI
TM
221 545
(00F), SET CR, IR, SCEI, ITV
213 531
P→AX,SetCR,PLA2
Call Function With Register (20)
3E2 994
SPB→RX
0DC 220
RX→BUS, BUS→A, DATA
0E5 229
—
30/3E8
053 083
P→AX,LoadIR,SCEI
2/202
One's Complement-Register to Register (22)
3F2 1010
SPB→RX,SCEI
008 008
RX→SPA, IR, SCEI, ITV
051 081
SPA + 0→RX, LOG
3/212

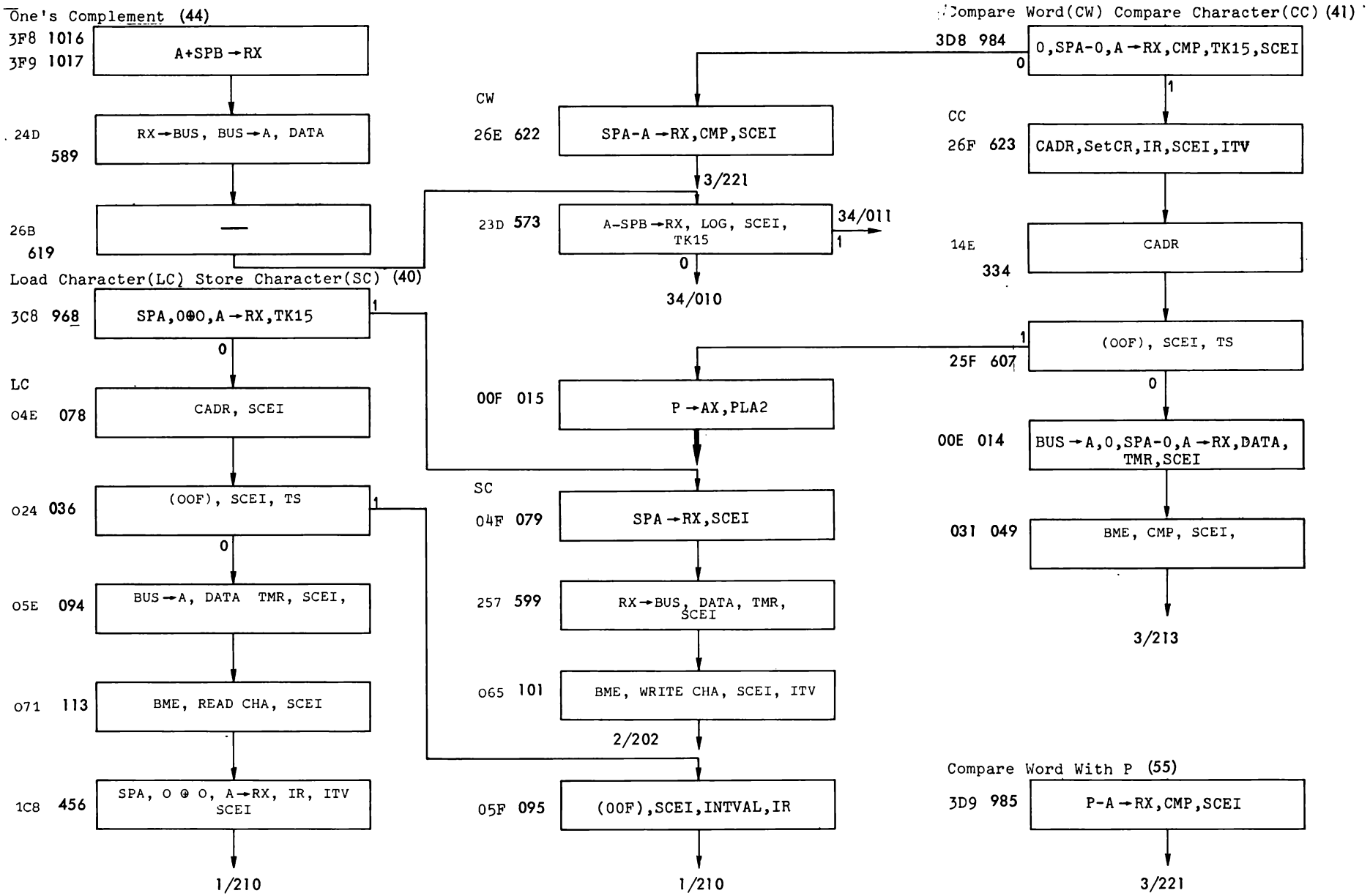

Call Function (30)
3E8 1000
SPA → AX
270 624
P → RX
Store P
236 566
RX → BUS, DATA TMR
229 553
WRITE, BME
20A 522
SPA-2 → AX, PLOK1
Store PSW
24C 588
AX-2 → RX, PSW → BUS, DATA, TMR, SCEI
269 617
WRITE, BME, SCEI, TPLA1
1
0
054 084
RX → SP, A → AX, IR, PLOK2, ITV, SCEI
2/203
Call Function With A15 (77)
3ED 1005
SPA → AX
055 085
RX → SP, A → AX
02C 044
SPA, 0+0 → RX, SCEI
017 023
0 → RX, SCEI, TZ
1
0
056 086
(OOF), IR, SCEI, ITV
2/203
057 087
RX → BUS, DATA, GFext
PLOK 2
Set P.I. F/F
06E 110
BIT 13 = 1
2/203
Return With A15 (43)
3EB 1003
SPB + 2 → AX
22F 559
BUS → A, R1, R2, AX+2 → RX, BUS → PSW, DATA, TMRN
Return (42)
3E9 1001
A → RX, TK15
1
21/261
0
260 608
SPB+2 → AX
02F 047
BUS → A, R1, R2, AX+2 → RX, DATA, BUS → CR, TMRN
01D 029
READ, BME, RX → P
1A7 423
BUS → R1, DATA, BIOVAL, SCEI, R2 → R1
193 403
R2 → R1, SCEI
13C 316
RX → SP, P → AX, IR, SCEI ITV
271 625
BUS → A, DATA, TMRN
07D 125
Read, BME
391 913
A → AX, PLA2

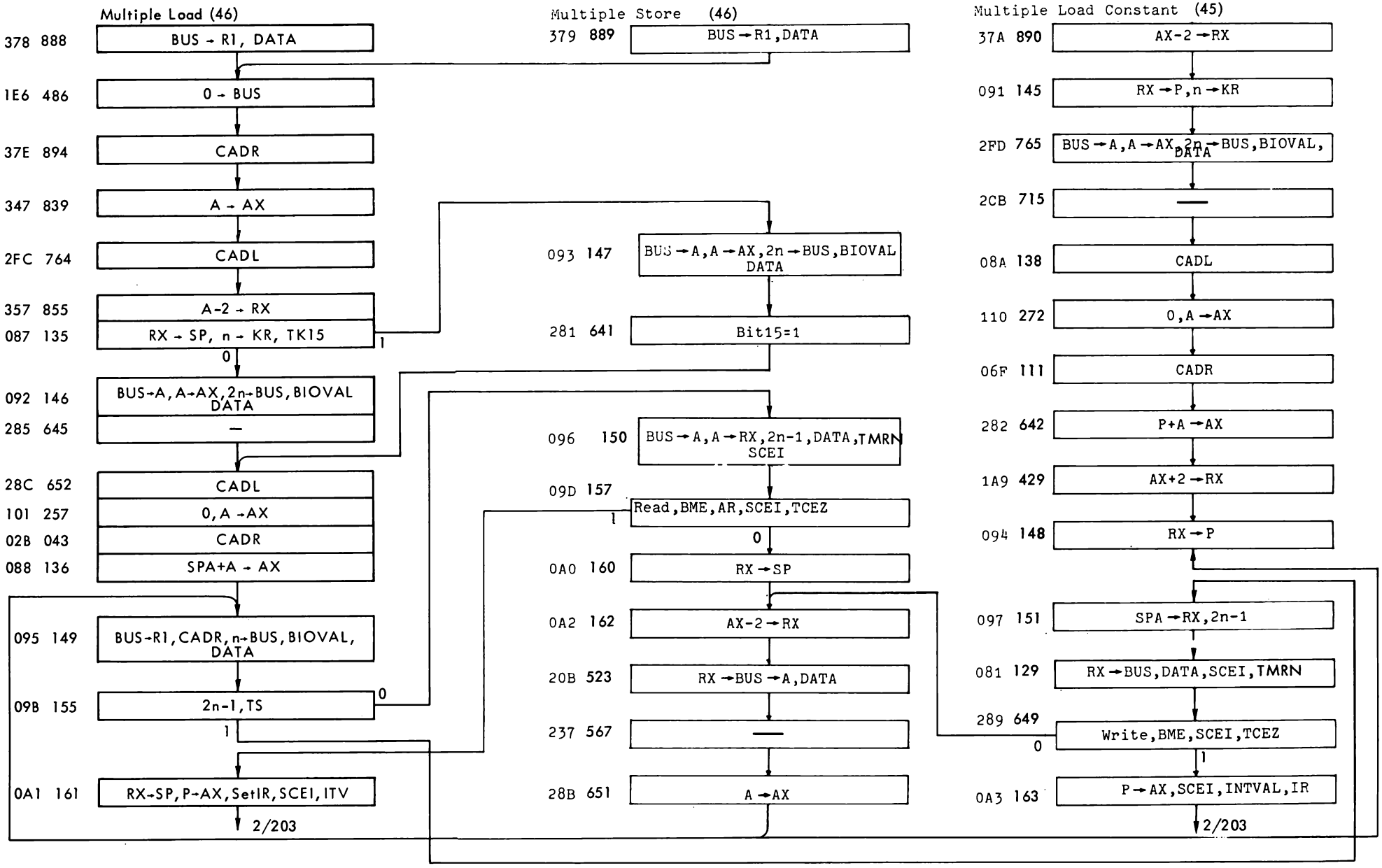

Execute Register (21)
3E3 995
SPB→RX
SP=K0-K15
261 609
RX→SP
04A 074
SPA→AX
058 088
BUS→A, SPA + A→RX, DATA
254 596
MASK KO.OPC
061 097
RX→SP
062 098
BUS→A, SPA + A→RX, DATA
255 597
MASK KO OPC 14
263 611
AX→RX, TZ
1
0
TRAP 64/3FF
265 613
(00F)
264 612
RX→SP
266 614
BUS→A, SPA.A→RX, DATA
256 598
MASK FORMAT
064 100
RX→SP
262 610
BUS→A, SPA + A→RX, DATA
258 600
MASK T2
267 615
(00F), TZ
1
0
25B 603
(00F)
TRAP 64/3FF
05A 090
Mask K9
21/063
25A 602
BUS→A,SPA+A→RX,DATA
A = 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1
This value on the Bus is ORed with the Scratch Pad content
SP K0 OPC 1 1 1 1 1 1 1 1 1 1
Result is in RX
A = 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0
A = 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
A = 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
A = 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
K9

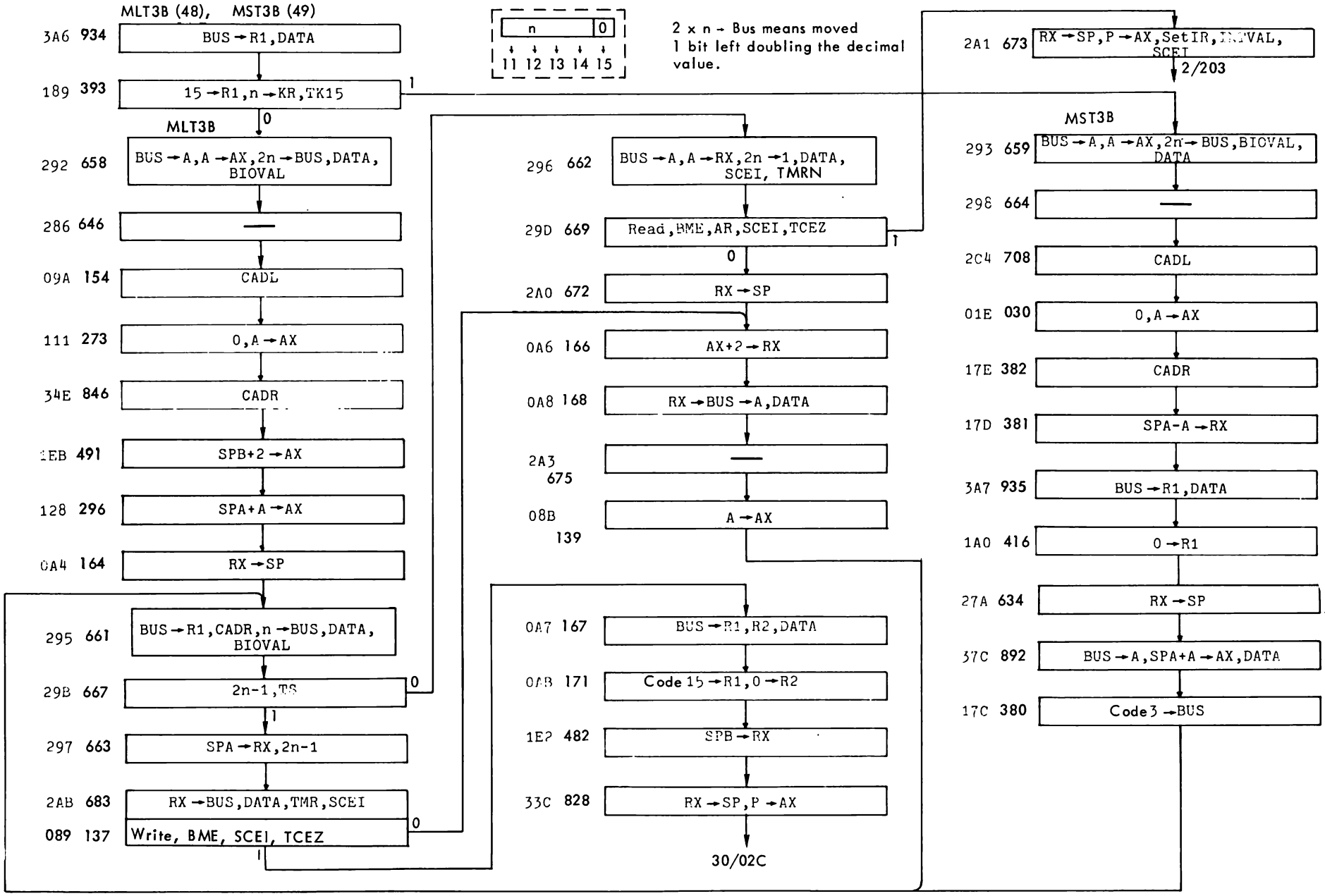

Execute Register Cont'd
063 099
AX→RX, TZ
1 (K9=0)
(K9=1) 0
36C 876
RX→SP, TS
1
0
36B 875
BUS→A→RX,DATA
36D 877
RX→SP
31B 795
OPC 12→BUS
36A 874
P-2→AX
A = 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 4 15
137 311
BUS→A, DATA,TMRN,BUS→K
139 313
READ, BME, TK 15
0 (T4,5,6,7 type)
1 (T1,2,3 type)
17B 379
AX→RX,TCEZ
1 (T2 type)
(T1,3 type) 0
38D 909
(00F)
38C 908
(00F)
17A 378
AX-2→RX
13E 318
RX→BUS→A,DATA
102 258
—
02A 042
SPA→RX,A→AX
06A 106
RX→BUS,BUS→R1R2,BUS→K,
DATA,PLOK0
247 583
—
00B 011
RX→BUS→A,DATA
Fetch 2
63/239

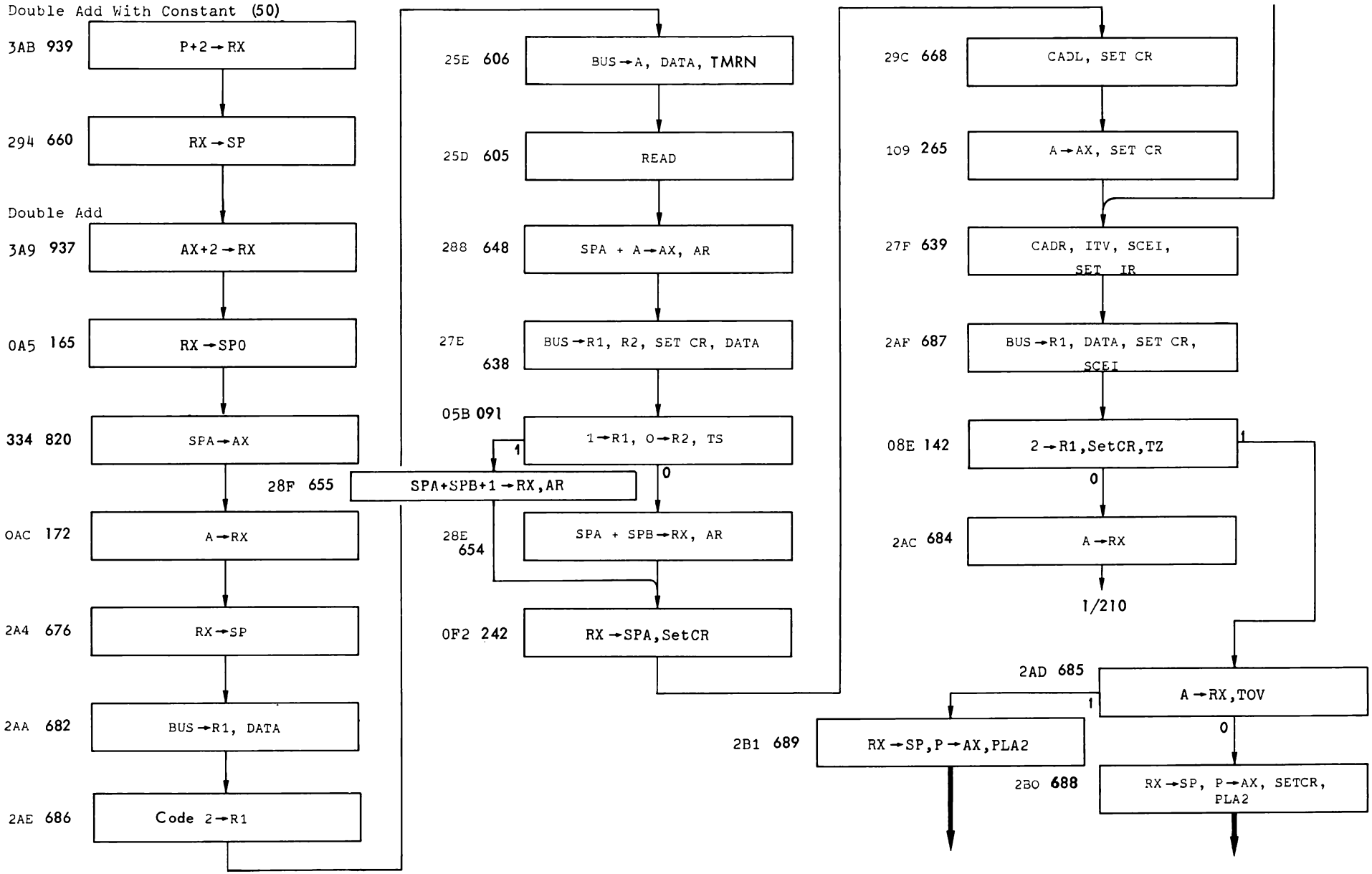

Load T3B Type Instructions (24)
3B6 950
BUS → R1, R2 → R1, BIOVAL, DATA
1A3 419
—
077 119
SPA +2 → RX
07A 122
RX → SP
199 409
A → RX
26A 618
RX → BUS, BUS → R1, R2, DATA
277 631
—
32/326
Store T3B Type Instructions (39)
30E 782
SPA → RX
003 003
RX → BUS, DATA, TMRN
029 041
WRITE
34D 845
SPB - 2 → RX
179 377
BUS → R1, R2 → R1, BIOVAL DATA
151 337
—
032 050
RX → SP, P → AX
30/02C
Store T3B Type Instructions in P (31)
30F 783
P → RX
Load P (31)
309 777
A → AX, AR, IR, SCEI, ITV
4/222
Load Register (31)
308 776
A → RX, AR, IR, SCEI, ITV
3/212
Store Register (31)
30C 780
SPA → RX, SCEI
33/211
Store P (31)
30D 781
P → RX, SCEI
33/211

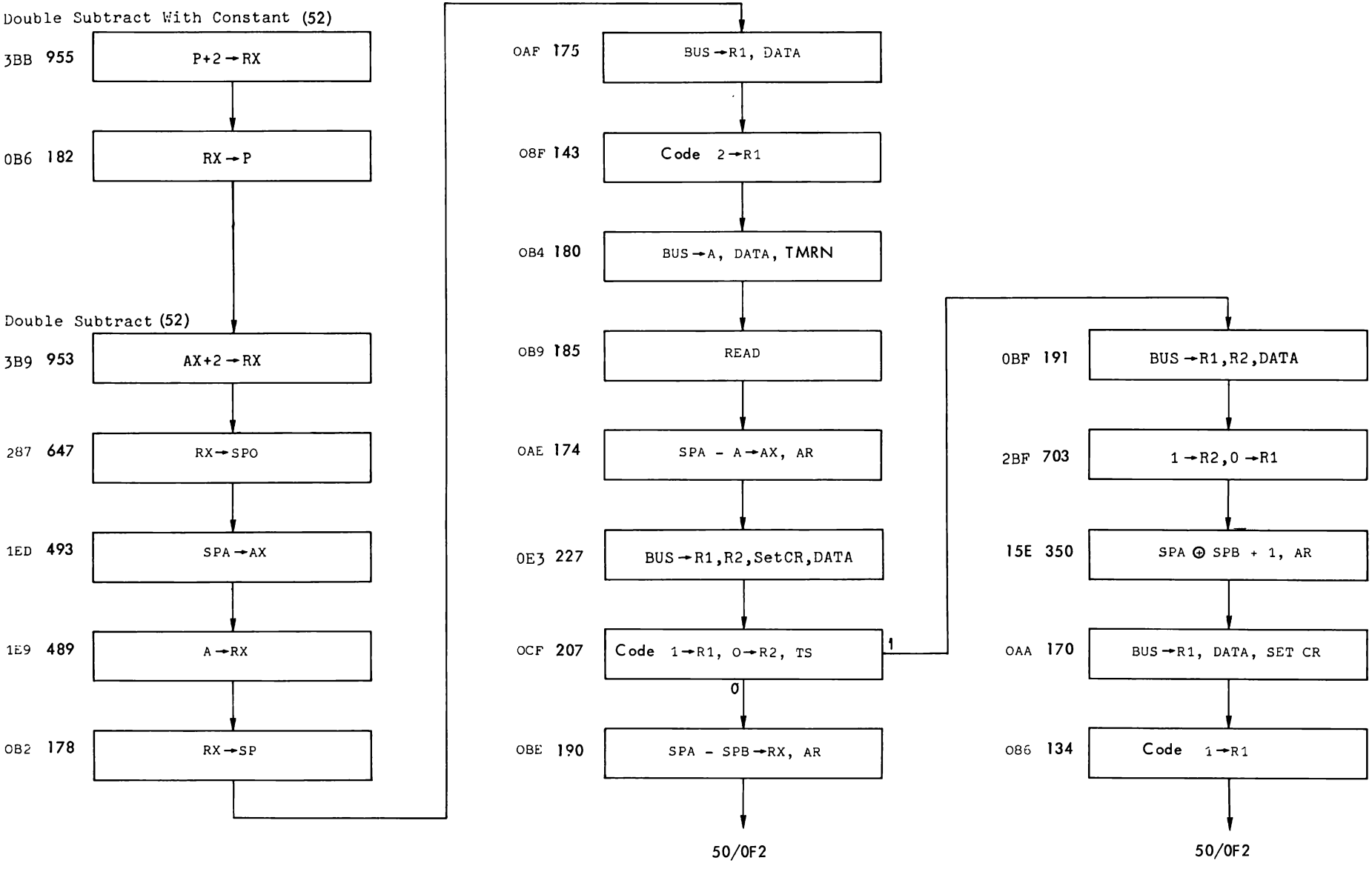

Absolute Conditional Branch
318 792
319 793
A → AX,SCEI,TCOK
2/021
0
2/020
Addition
328 808
SPA+A → RX,AR,SCEI,TK15
1
0
Subtract
338 824
SPA-A → RX,AR,SCEI,TK15
1
0
Logical AND
348 840
SPA.A → RX,AR,SCEI,TK15
1
0
Logical OR
358 856
SPA⊕A → RX,LOG,SCEI,TK15
1
0
Exclusive OR
368 872
SPA+A → RX,LOG,SCEI,TK15
1
0
34/010
Increment Memory
329 809
(52F),AR,SCEI,TK15
1
083 131
A+1 → RX,AR,SCEI
0
082 130
P+A → AX,SetCR,IR,SCEI,ITV
Two's Complement
339 825
(53F),AR,SCEI,TK15
085 133
0-A → RX,AR,SCEI,
084
132
P-A → AX, SET CR, IR, SCEI, ITV
2/203
34/011
Clear Memory
349 841
0-A → RX,SCEI
33/211


One's Complement (44)
3F8 1016
3F9 1017
A+SPB→RX
24D 589
RX→BUS, BUS→A, DATA
26B 619
—
Load Character(LC) Store Character(SC) (40)
3C8 968
SPA,0⊕O,A→RX,TK15
1
0
LC
04E 078
CADR, SCEI
024 036
(OOF), SCEI, TS
1
0
05E 094
BUS→A, DATA TMR, SCEI,
071 113
BME, READ CHA, SCEI
1C8 456
SPA, O ⊕ O, A→RX, IR, ITV SCEI
1/210
CW
26E 622
SPA-A→RX,CMP,SCEI
3/221
23D 573
A-SPB→RX, LOG, SCEI, TK15
34/011
1
0
34/010
00F 015
P→AX,PLA2
SC
04F 079
SPA→RX,SCEI
257 599
RX→BUS, DATA, TMR, SCEI
065 101
BME, WRITE CHA, SCEI, ITV
2/202
05F 095
(OOF),SCEI,INTVAL,IR
1/210
Compare Word(CW) Compare Character(CC) (41)
3D8 984
0,SPA-0,A→RX,CMP,TK15,SCEI
0
1
CC
26F 623
CADR,SetCR,IR,SCEI,ITV
14E 334
CADR
1
25F 607
(OOF), SCEI, TS
0
00E 014
BUS→A,0,SPA-0,A→RX,DATA, TMR,SCEI
031 049
BME, CMP, SCEI,
3/213
Compare Word With P (55)
3D9 985
P-A→RX,CMP,SCEI
3/221

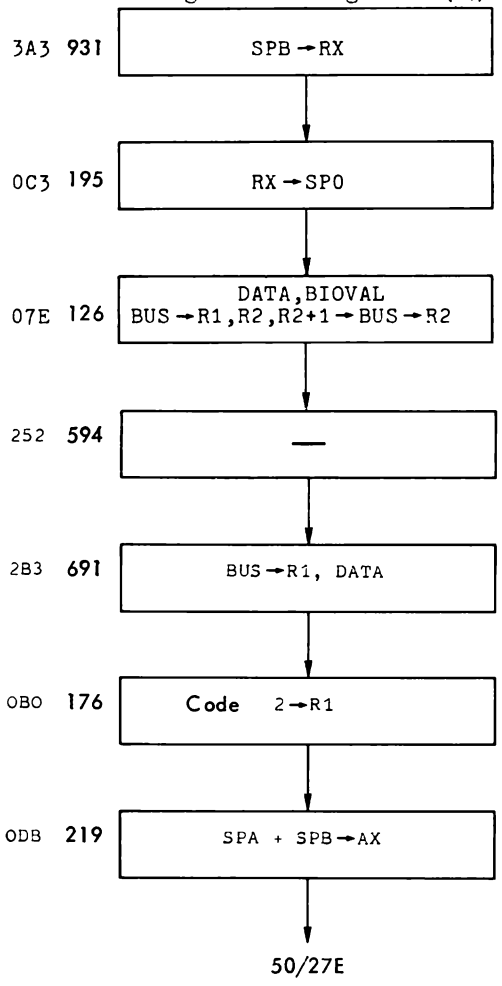

Multiple Load (46)
378 888
BUS → R1, DATA
1E6 486
0 → BUS
37E 894
CADR
347 839
A → AX
2FC 764
CADL
357 855
A-2 → RX
087 135
RX → SP, n → KR, TK15
1
0
092 146
BUS→A, A→AX, 2n→BUS, BIOVAL DATA
285 645
—
28C 652
CADL
101 257
0, A → AX
02B 043
CADR
088 136
SPA+A → AX
095 149
BUS→R1, CADR, n→BUS, BIOVAL, DATA
09B 155
2n-1, TS
0
1
0A1 161
RX→SP, P→AX, SetIR, SCEI, ITV
2/203
Multiple Store (46)
379 889
BUS→R1, DATA
093 147
BUS→A, A→AX, 2n→BUS, BIOVAL DATA
281 641
Bit15=1
096 150
BUS→A, A→RX, 2n-1, DATA, TMRN SCEI
09D 157
Read, BME, AR, SCEI, TCEZ
1
0
0A0 160
RX→SP
0A2 162
AX-2→RX
20B 523
RX→BUS→A, DATA
237 567
—
28B 651
A→AX
Multiple Load Constant (45)
37A 890
AX-2→RX
091 145
RX→P, n→KR
2FD 765
BUS→A, A→AX, 2n→BUS, BIOVAL, DATA
2CB 715
—
08A 138
CADL
110 272
0, A→AX
06F 111
CADR
282 642
P+A→AX
1A9 429
AX+2→RX
094 148
RX→P
097 151
SPA→RX, 2n-1
081 129
RX→BUS, DATA, SCEI, TMRN
289 649
Write, BME, SCEI, TCEZ
0
1
0A3 163
P→AX, SCEI, INTVAL, IR
2/203


MLT3B (48), MST3B (49)
3A6 934
BUS → R1, DATA
189 393
15 → R1, n → KR, TK15
1
0
MLT3B
292 658
BUS → A, A → AX, 2n → BUS, DATA, BIOVAL
286 646
—
09A 154
CADL
111 273
0, A → AX
34E 846
CADR
1EB 491
SPB+2 → AX
128 296
SPA+A → AX
0A4 164
RX → SP
295 661
BUS → R1, CADR, n → BUS, DATA, BIOVAL
29B 667
2n-1, TS
0
1
297 663
SPA → RX, 2n-1
2AB 683
RX → BUS, DATA, TMR, SCEI
089 137
Write, BME, SCEI, TCEZ
0
1
n
0
11 12 13 14 15
2 x n → Bus means moved 1 bit left doubling the decimal value.
296 662
BUS → A, A → RX, 2n → 1, DATA, SCEI, TMRN
29D 669
Read, BME, AR, SCEI, TCEZ
1
0
2A0 672
RX → SP
0A6 166
AX+2 → RX
0A8 168
RX → BUS → A, DATA
2A3 675
—
08B 139
A → AX
0A7 167
BUS → R1, R2, DATA
0AB 171
Code 15 → R1, 0 → R2
1E2 482
SPB → RX
33C 828
RX → SP, P → AX
30/02C
2A1 673
RX → SP, P → AX, SetIR, BIOVAL, SCEI
2/203
MST3B
293 659
BUS → A, A → AX, 2n → BUS, BIOVAL, DATA
298 664
—
2C4 708
CADL
01E 030
0, A → AX
17E 382
CADR
17D 381
SPA-A → RX
3A7 935
BUS → R1, DATA
1A0 416
0 → R1
27A 634
RX → SP
37C 892
BUS → A, SPA+A → AX, DATA
17C 380
Code 3 → BUS

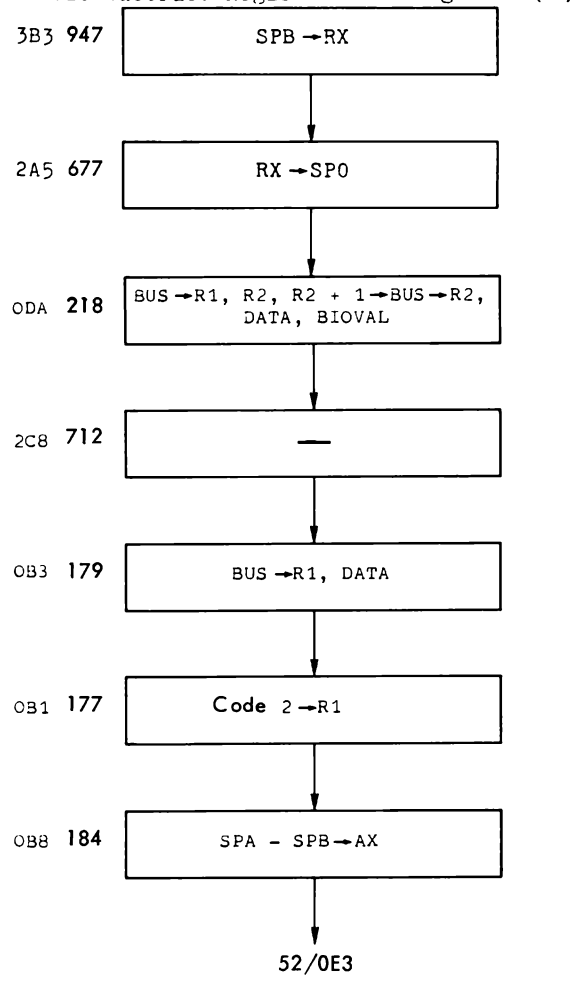

Double Add With Constant (50)
3AB 939
P+2→RX
294 660
RX→SP
Double Add
3A9 937
AX+2→RX
0A5 165
RX→SP0
334 820
SPA→AX
0AC 172
A→RX
2A4 676
RX→SP
2AA 682
BUS→R1, DATA
2AE 686
Code 2→R1
25E 606
BUS→A, DATA, TMRN
25D 605
READ
288 648
SPA + A→AX, AR
27E 638
BUS→R1, R2, SET CR, DATA
05B 091
1→R1, 0→R2, TS
1
0
28F 655
SPA+SPB+1→RX,AR
28E 654
SPA + SPB→RX, AR
0F2 242
RX→SPA,SetCR
29C 668
CADL, SET CR
109 265
A→AX, SET CR
27F 639
CADR, ITV, SCEI, SET IR
2AF 687
BUS→R1, DATA, SET CR, SCEI
08E 142
2→R1,SetCR,TZ
1
0
2AC 684
A→RX
1/210
2AD 685
A→RX,TOV
1
0
2B1 689
RX→SP,P→AX,PLA2
2B0 688
RX→SP, P→AX, SETCR, PLA2

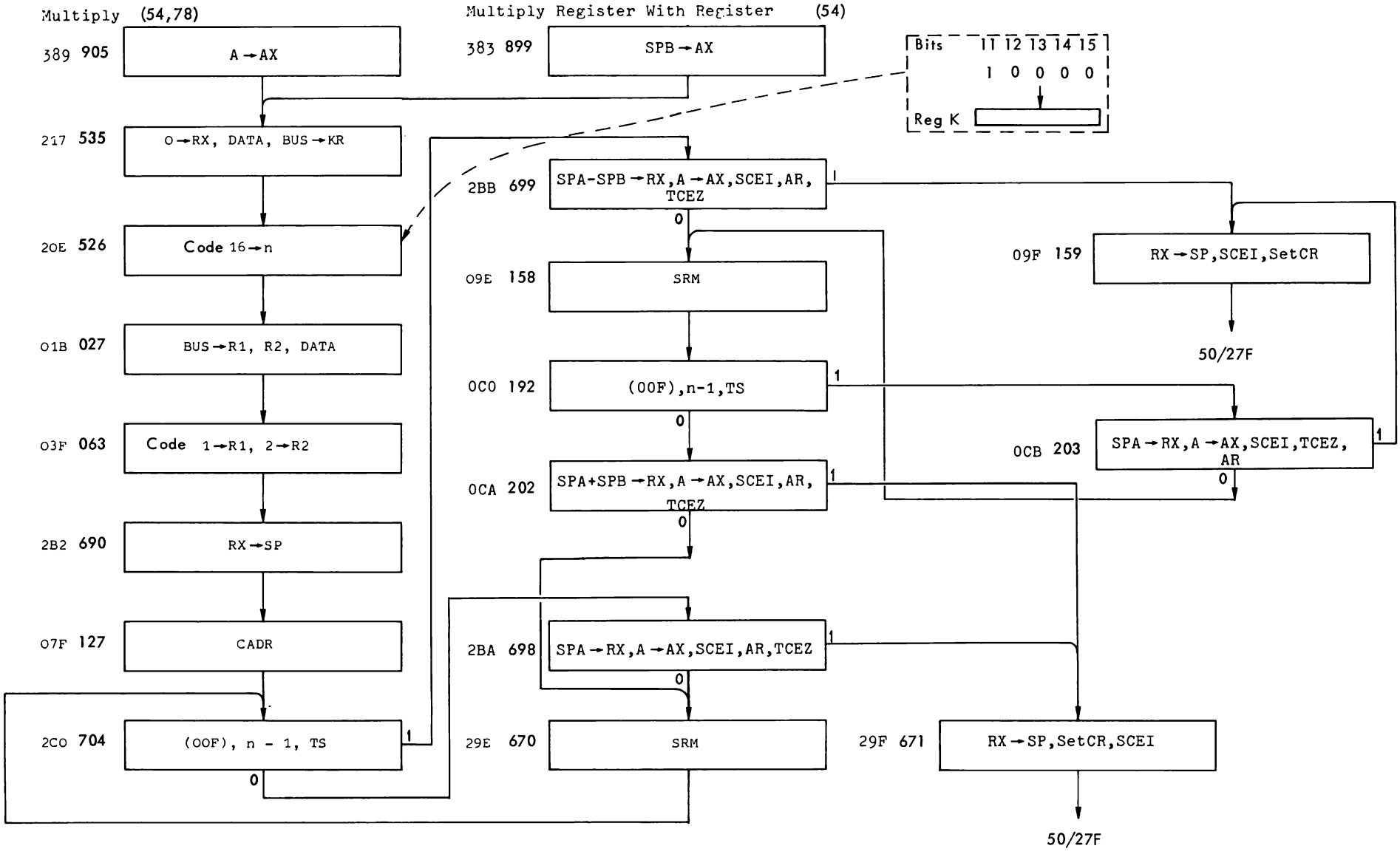

Double Subtract With Constant (52)
3BB 955
P+2→RX
0B6 182
RX→P
Double Subtract (52)
3B9 953
AX+2→RX
287 647
RX→SPO
1ED 493
SPA→AX
1E9 489
A→RX
0B2 178
RX→SP
0AF 175
BUS→R1, DATA
08F 143
Code 2→R1
0B4 180
BUS→A, DATA, TMRN
0B9 185
READ
0AE 174
SPA - A→AX, AR
0E3 227
BUS→R1,R2,SetCR,DATA
0CF 207
Code 1→R1, 0→R2, TS
1
0
0BE 190
SPA - SPB→RX, AR
50/0F2
0BF 191
BUS→R1,R2,DATA
2BF 703
1→R2,0→R1
15E 350
SPA ⊕ SPB + 1, AR
0AA 170
BUS→R1, DATA, SET CR
086 134
Code 1→R1
50/0F2

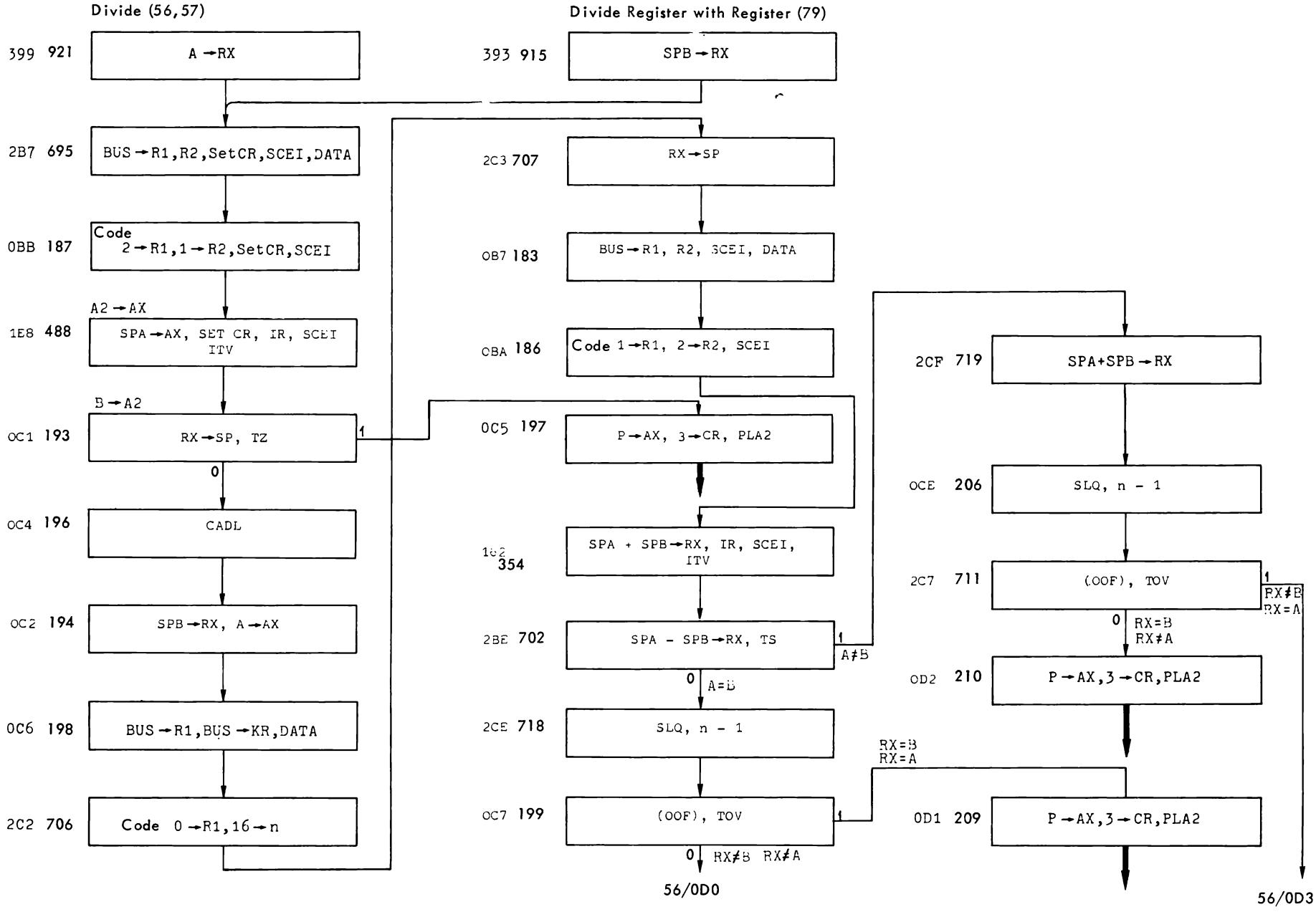
Double Add Register to Register (51)

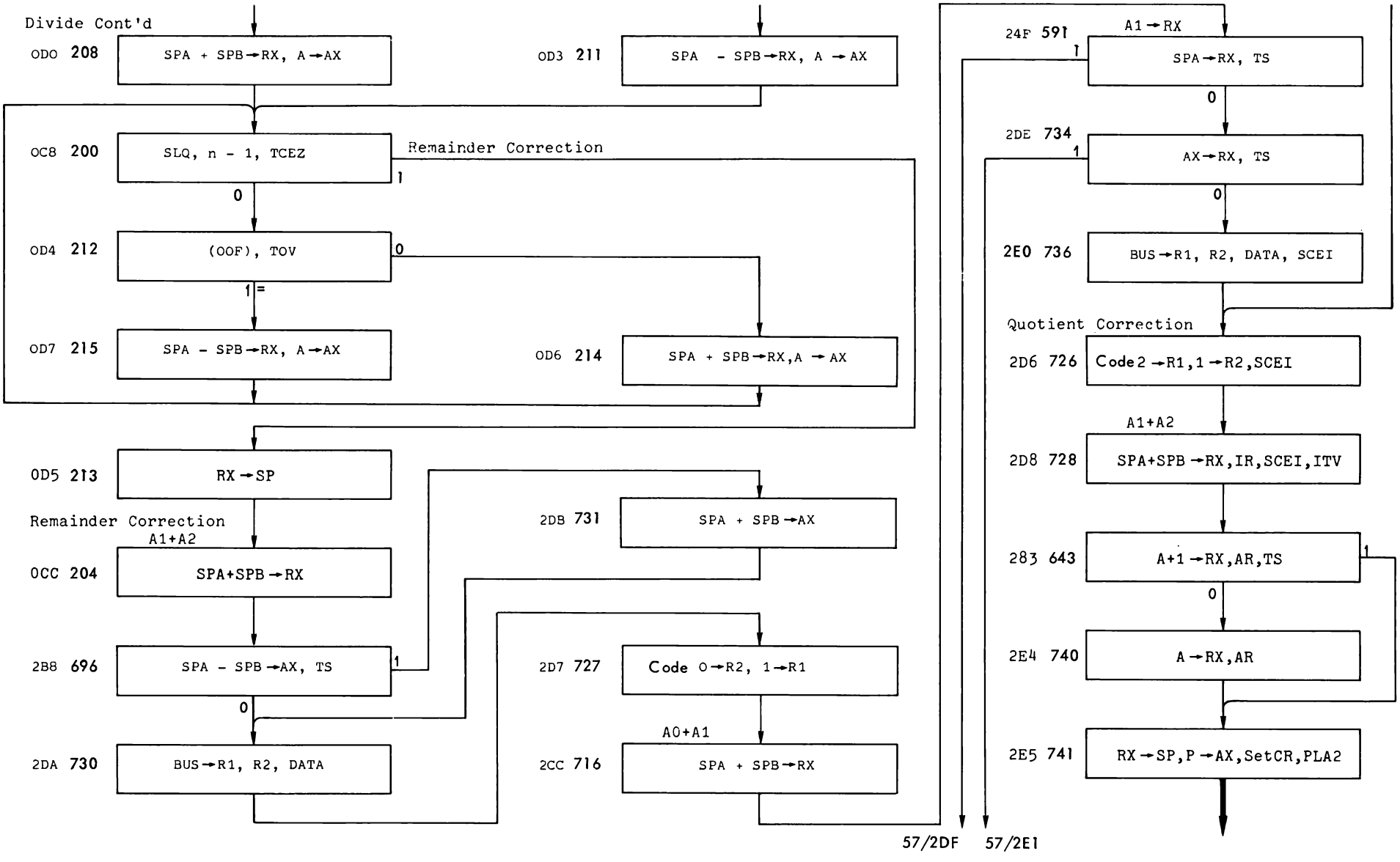


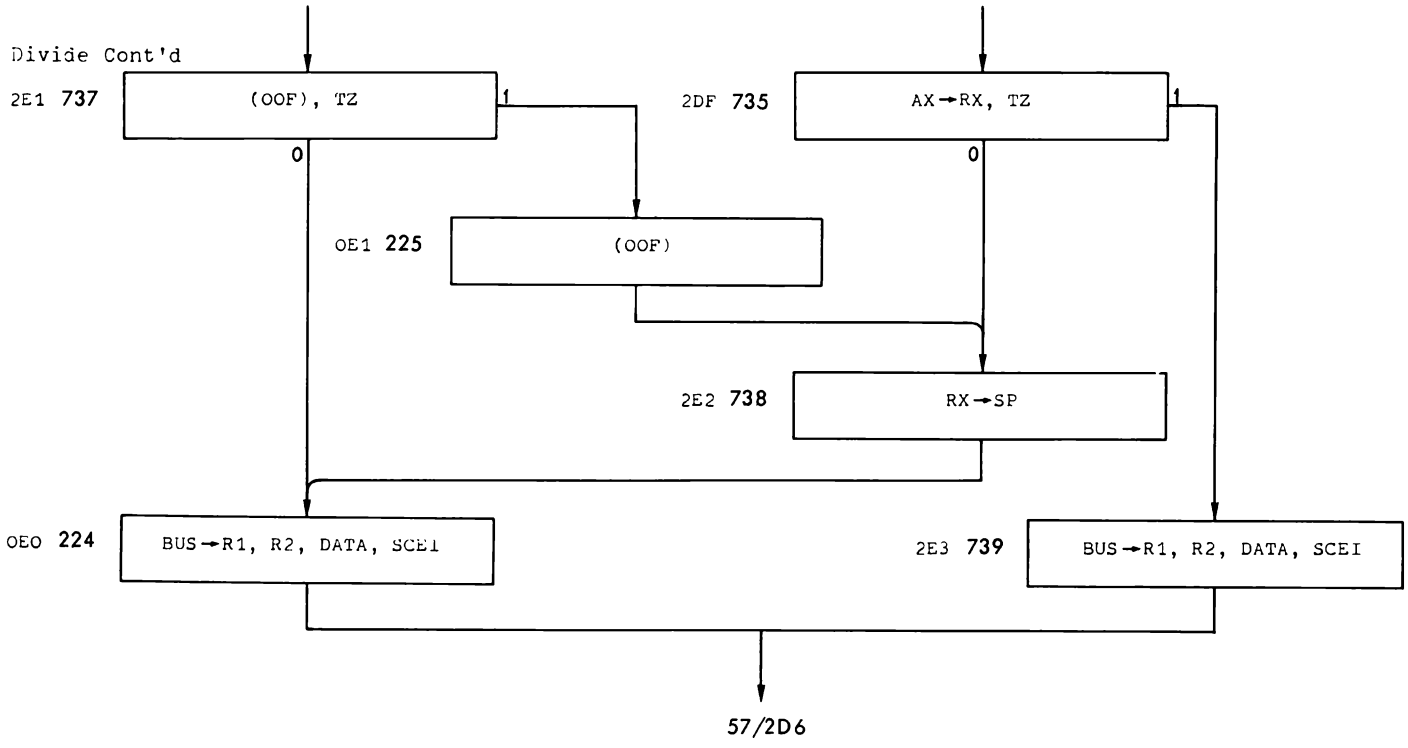
Double Subtract Register With Register (53)



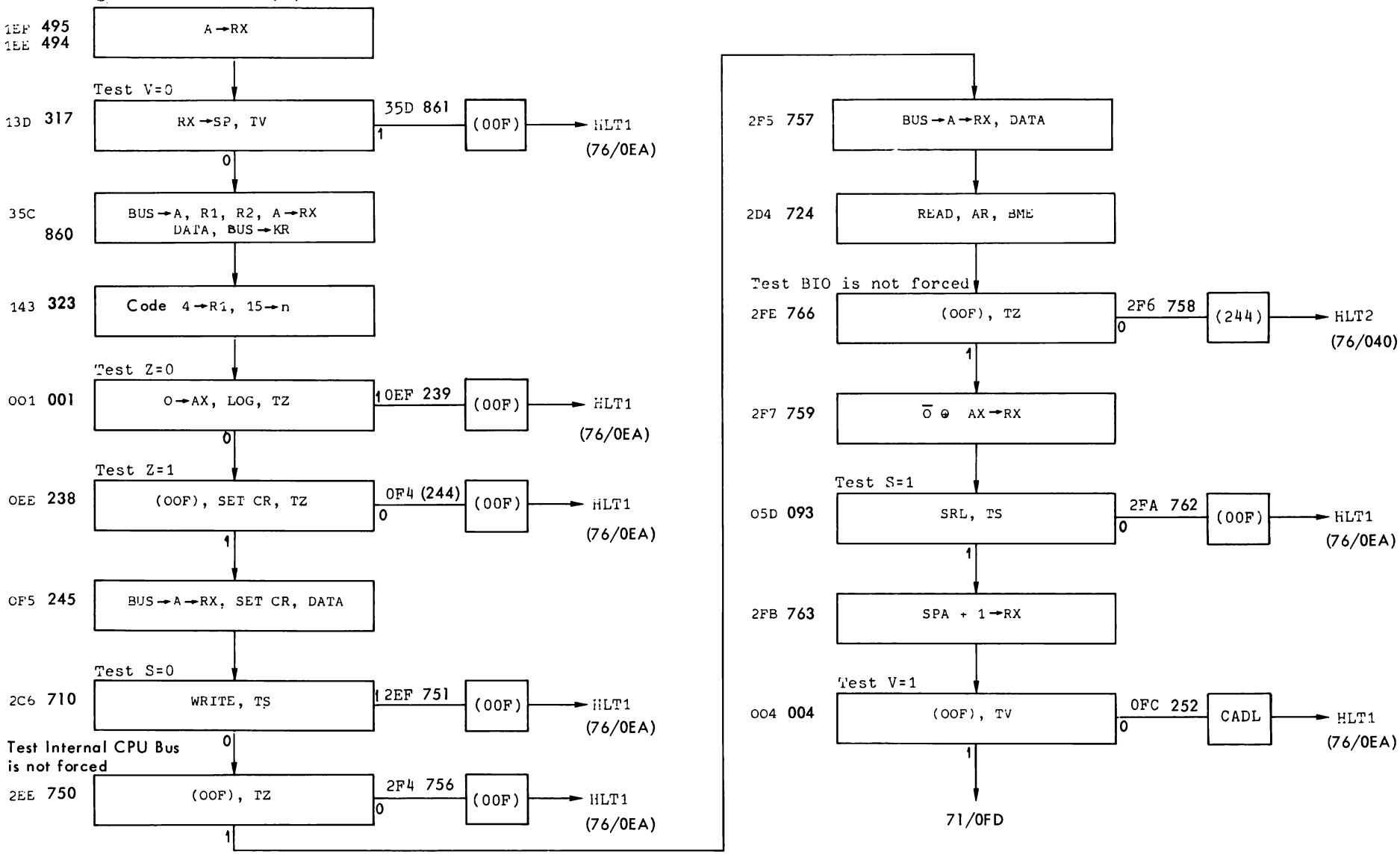

Multiply (54,78)
389 905
A → AX
217 535
0 → RX, DATA, BUS → KR
20E 526
Code 16 → n
01B 027
BUS → R1, R2, DATA
03F 063
Code 1 → R1, 2 → R2
2B2 690
RX → SP
07F 127
CADR
2C0 704
(00F), n - 1, TS
1
0
Multiply Register With Register (54)
383 899
SPB → AX
Bits 11 12 13 14 15
1 0 0 0 0
Reg K
2BB 699
SPA-SPB → RX, A → AX, SCEI, AR, TCEZ
1
0
09E 158
SRM
0C0 192
(00F), n-1, TS
1
0
0CA 202
SPA+SPB → RX, A → AX, SCEI, AR, TCEZ
1
0
2BA 698
SPA → RX, A → AX, SCEI, AR, TCEZ
1
0
29E 670
SRM
09F 159
RX → SP, SCEI, SetCR
50/27F
0CB 203
SPA → RX, A → AX, SCEI, TCEZ, AR
1
0
29F 671
RX → SP, SetCR, SCEI
50/27F

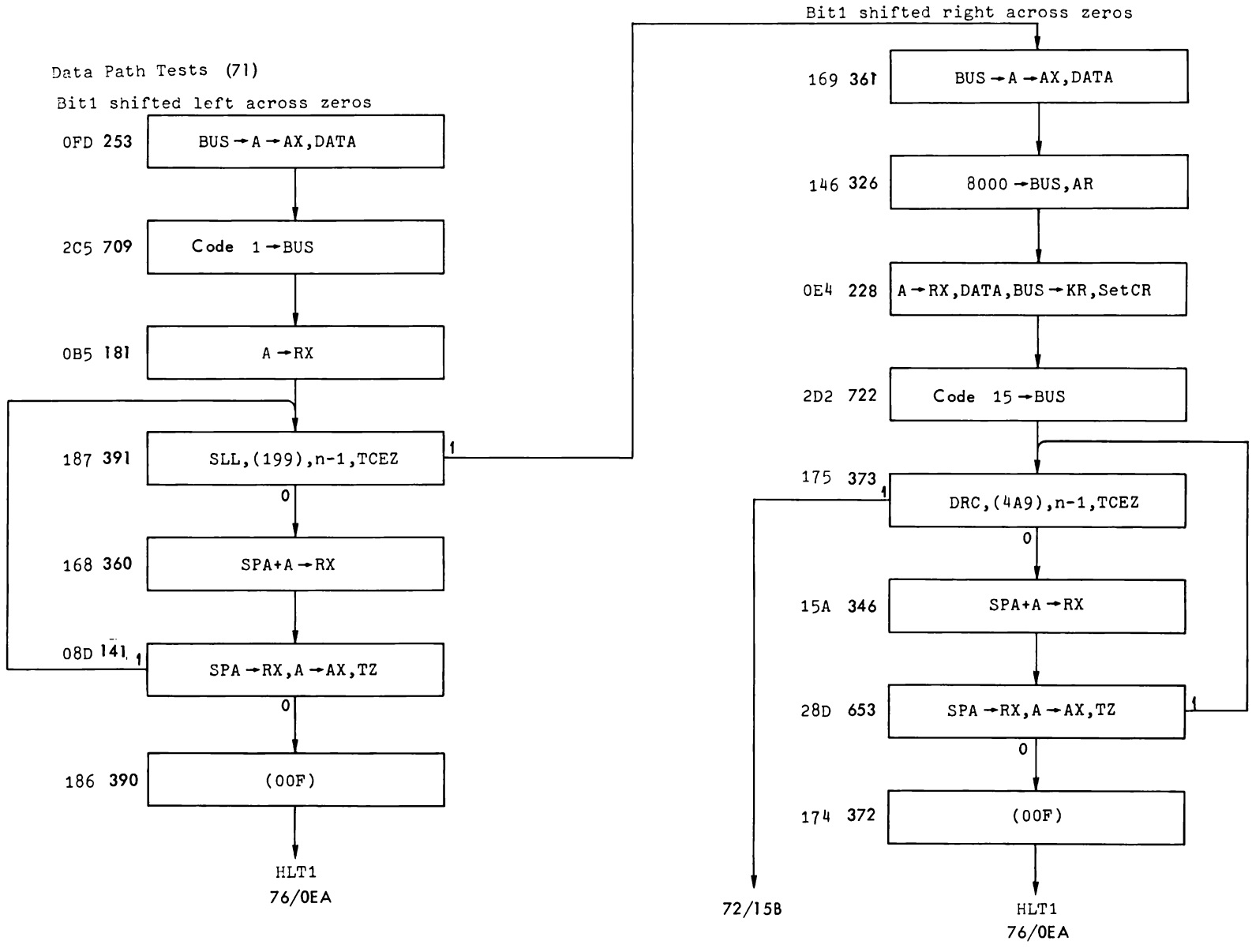

Divide (56,57)
399 921
A → RX
2B7 695
BUS → R1,R2,SetCR,SCEI,DATA
0BB 187
Code
2 → R1,1 → R2,SetCR,SCEI
A2 → AX
1E8 488
SPA → AX, SET CR, IR, SCEI
ITV
B → A2
0C1 193
RX → SP, TZ
1
0
0C4 196
CADL
0C2 194
SPB → RX, A → AX
0C6 198
BUS → R1,BUS → KR,DATA
2C2 706
Code 0 → R1,16 → n
Divide Register with Register (79)
393 915
SPB → RX
2C3 707
RX → SP
0B7 183
BUS → R1, R2, SCEI, DATA
0BA 186
Code 1 → R1, 2 → R2, SCEI
0C5 197
P → AX, 3 → CR, PLA2
SPA + SPB → RX, IR, SCEI,
ITV
354
2BE 702
SPA - SPB → RX, TS
1
A≠B
0
A=B
2CE 718
SLQ, n - 1
0C7 199
(OOF), TOV
1
0
RX≠B RX≠A
56/0D0
2CF 719
SPA+SPB → RX
0CE 206
SLQ, n - 1
2C7 711
(OOF), TOV
1
RX≠B
RX=A
0
RX=B
RX≠A
0D2 210
P → AX,3 → CR,PLA2
RX=B
RX=A
0D1 209
P → AX,3 → CR,PLA2
56/0D3

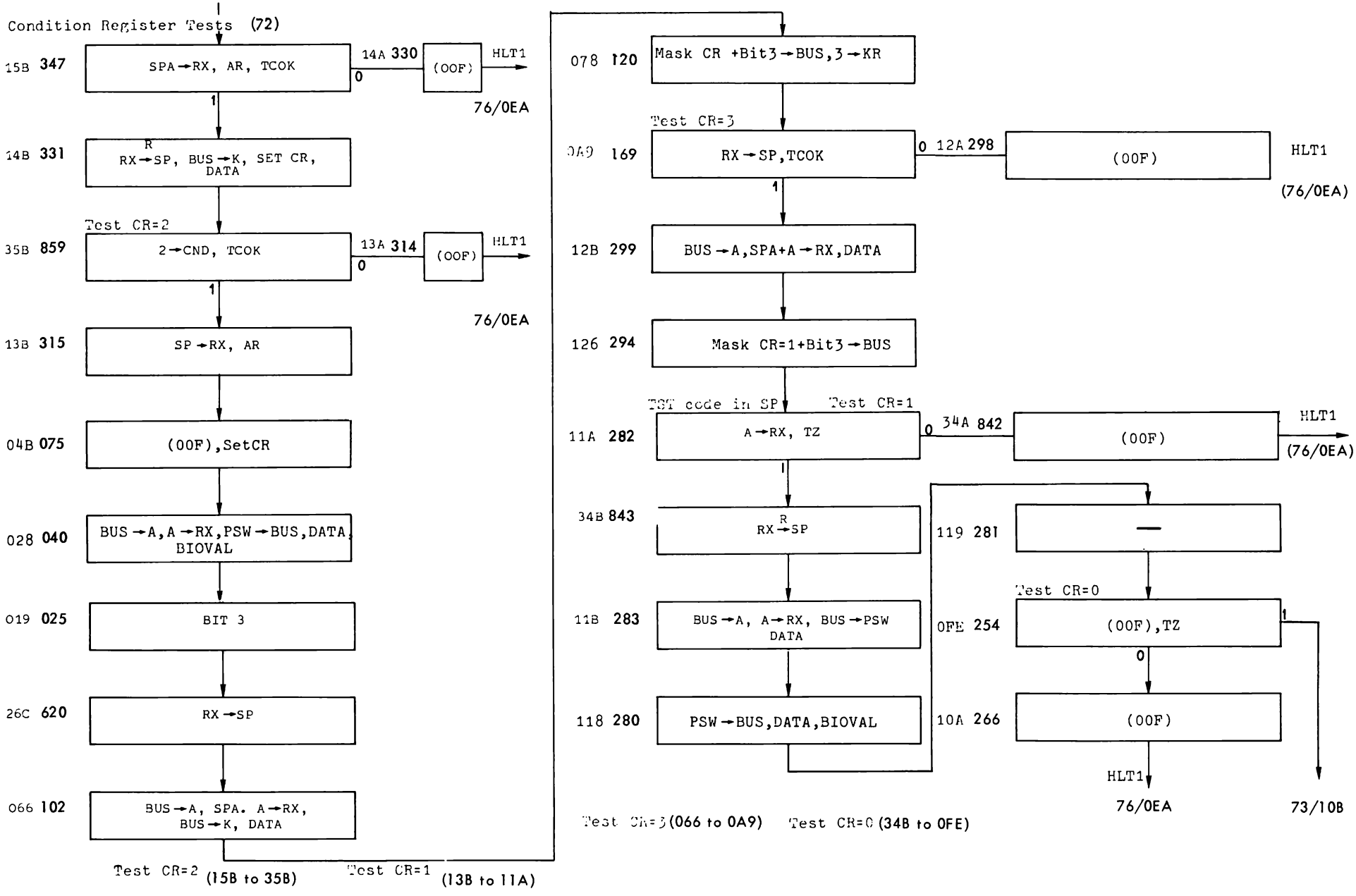

Divide Cont'd
0D0 208
SPA + SPB→RX, A→AX
0D3 211
SPA - SPB→RX, A→AX
0C8 200
SLQ, n - 1, TCEZ
Remainder Correction
1
0
0D4 212
(OOF), TOV
0
1=
0D7 215
SPA - SPB→RX, A→AX
0D6 214
SPA + SPB→RX,A→AX
0D5 213
RX→SP
Remainder Correction
A1+A2
0CC 204
SPA+SPB→RX
2DB 731
SPA + SPB→AX
2B8 696
SPA - SPB→AX, TS
1
0
2D7 727
Code 0→R2, 1→R1
A0+A1
2DA 730
BUS→R1, R2, DATA
2CC 716
SPA + SPB→RX
24F 591
A1→RX
1
SPA→RX, TS
0
2DE 734
1
AX→RX, TS
0
2E0 736
BUS→R1, R2, DATA, SCEI
Quotient Correction
2D6 726
Code2→R1,1→R2,SCEI
A1+A2
2D8 728
SPA+SPB→RX,IR,SCEI,ITV
283 643
A+1→RX,AR,TS
1
0
2E4 740
A→RX,AR
2E5 741
RX→SP,P→AX,SetCR,PLA2
57/2DF
57/2E1

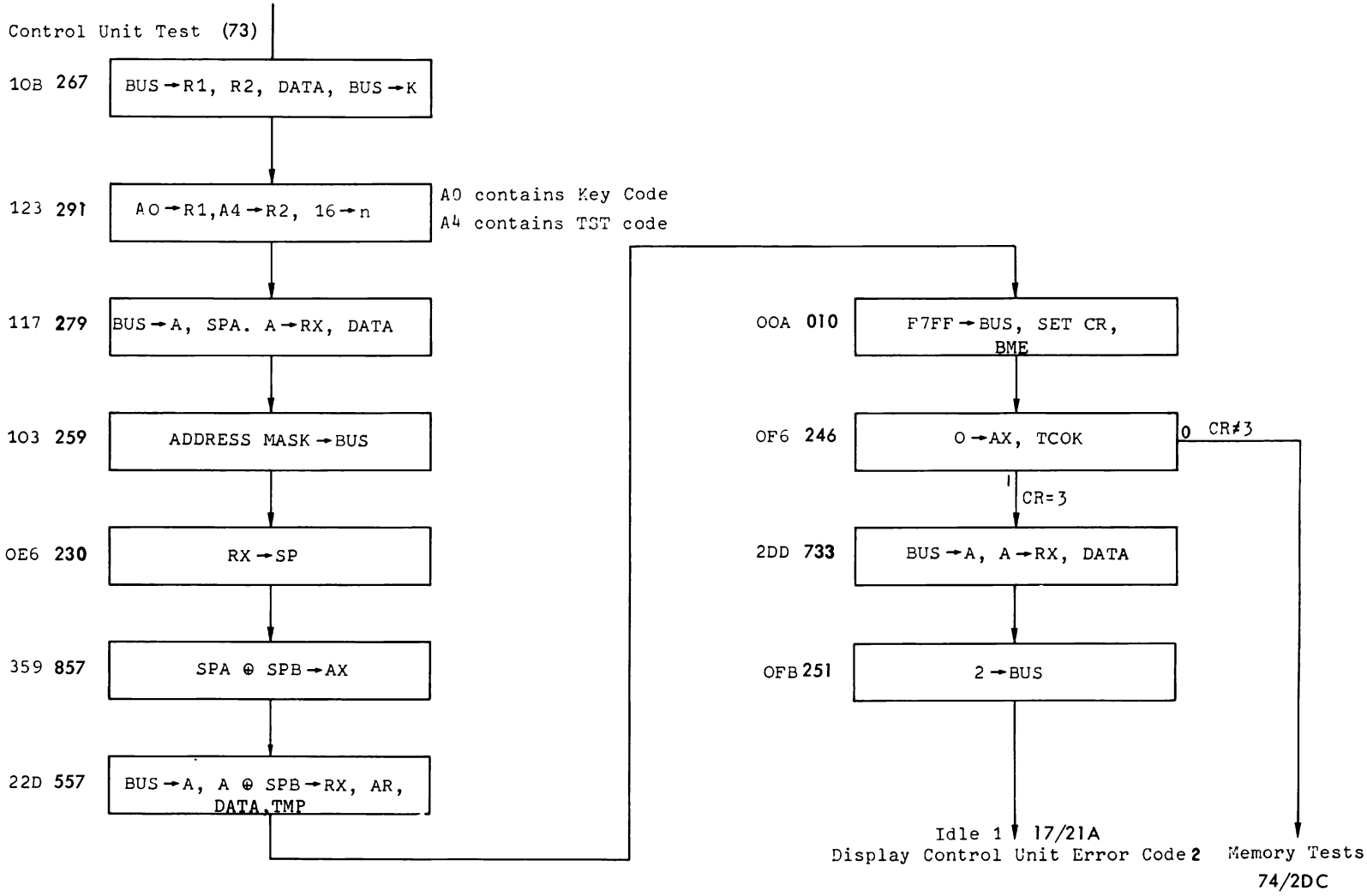

Divide Cont'd
2E1 737
(OOF), TZ
1
0
2DF 735
AX→RX, TZ
1
0
OE1 225
(OOF)
2E2 738
RX→SP
OE0 224
BUS→R1, R2, DATA, SCE1
2E3 739
BUS→R1, R2, DATA, SCEI
57/2D6

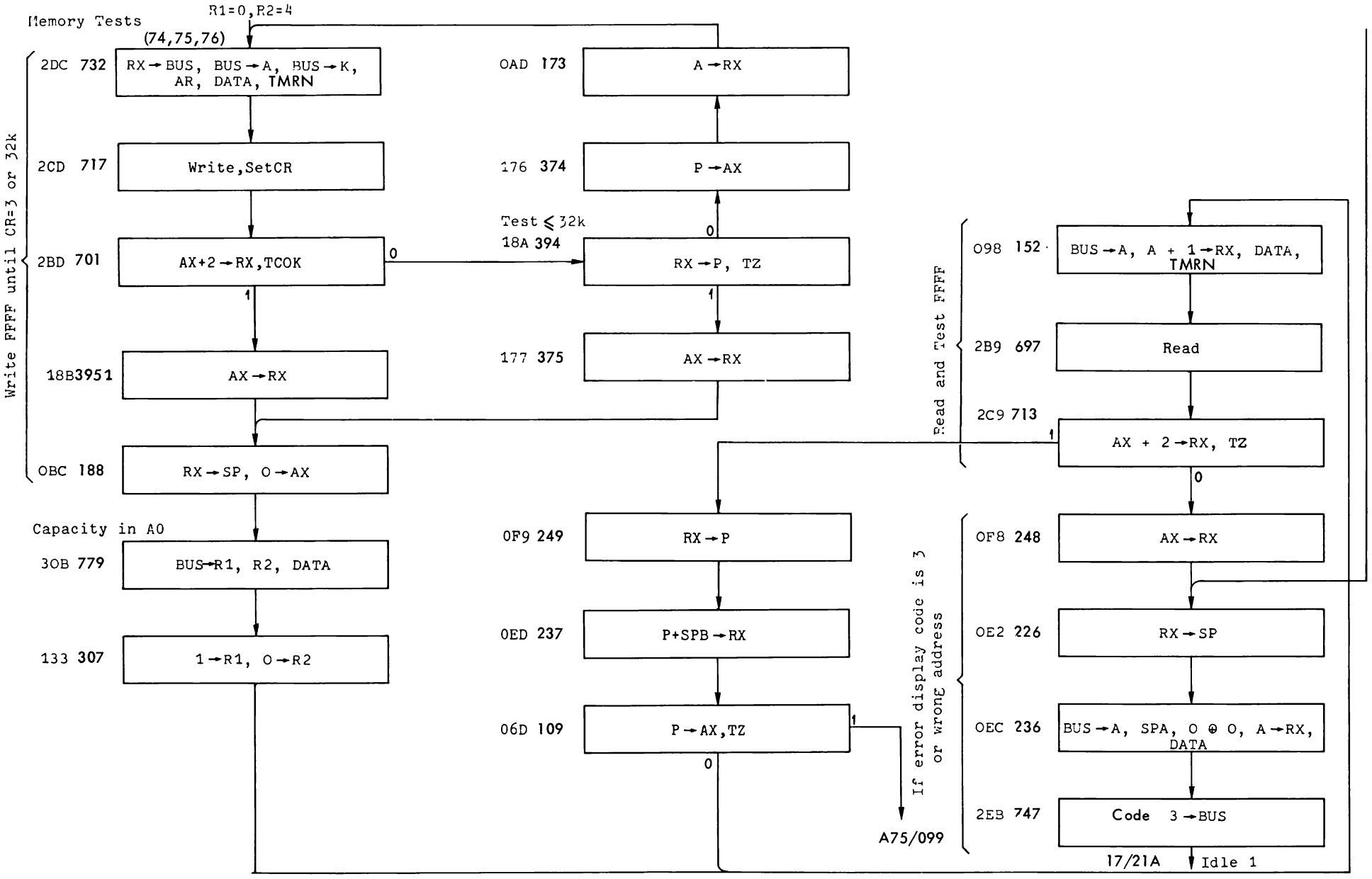
SPALU Flags and Bus Tests (71)

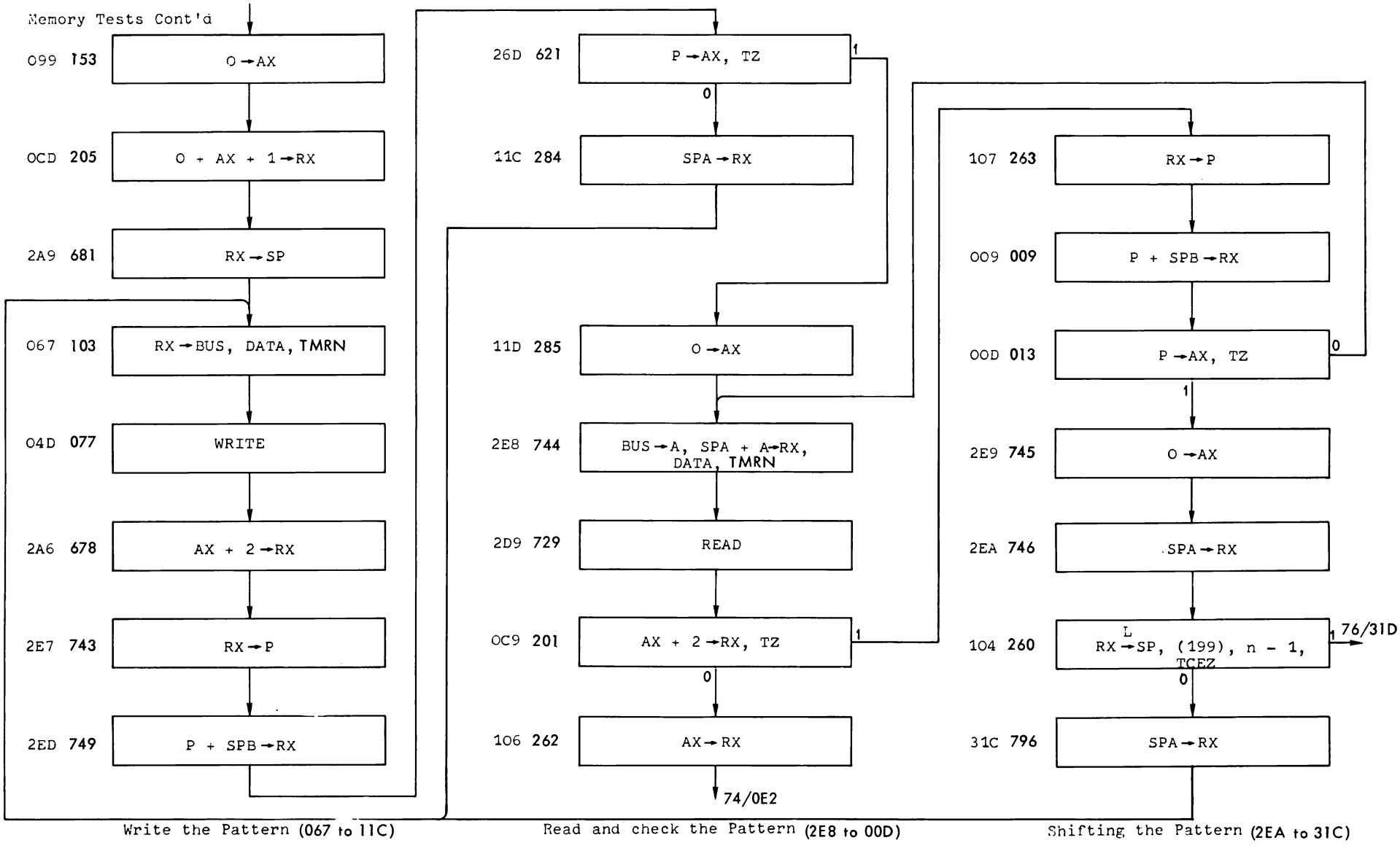



Data Path Tests (71)
Bit1 shifted left across zeros
0FD 253
BUS→A→AX,DATA
2C5 709
Code 1→BUS
0B5 181
A→RX
187 391
SLL,(199),n-1,TCEZ
1
0
168 360
SPA+A→RX
08D 141
1
SPA→RX,A→AX,TZ
0
186 390
(00F)
HLT1
76/0EA
Bit1 shifted right across zeros
169 361
BUS→A→AX,DATA
146 326
8000→BUS,AR
0E4 228
A→RX,DATA,BUS→KR,SetCR
2D2 722
Code 15→BUS
175 373
1
DRC,(4A9),n-1,TCEZ
0
15A 346
SPA+A→RX
28D 653
SPA→RX,A→AX,TZ
1
0
174 372
(00F)
72/15B
HLT1
76/0EA


Condition Register Tests (72)
15B 347
SPA→RX, AR, TCOK
14A 330
(00F)
HLT1
76/0EA
14B 331
R
RX→SP, BUS→K, SET CR, DATA
Test CR=2
35B 859
2→CND, TCOK
13A 314
(00F)
HLT1
76/0EA
13B 315
SP→RX, AR
04B 075
(00F),SetCR
028 040
BUS→A,A→RX,PSW→BUS,DATA, BIOVAL
019 025
BIT 3
26C 620
RX→SP
066 102
BUS→A, SPA. A→RX, BUS→K, DATA
Test CR=2 (15B to 35B)
Test CR=1 (13B to 11A)
078 120
Mask CR +Bit3→BUS,3→KR
Test CR=3
0A9 169
RX→SP,TCOK
12A 298
(00F)
HLT1
(76/0EA)
12B 299
BUS→A,SPA+A→RX,DATA
126 294
Mask CR=1+Bit3→BUS
TST code in SP
Test CR=1
11A 282
A→RX, TZ
34A 842
(00F)
HLT1
(76/0EA)
34B 843
R
RX→SP
11B 283
BUS→A, A→RX, BUS→PSW DATA
118 280
PSW→BUS,DATA,BIOVAL
119 281
—
Test CR=0
0FE 254
(00F),TZ
10A 266
(00F)
HLT1
76/0EA
73/10B
Test CR=3 (066 to 0A9)
Test CR=0 (34B to 0FE)


Control Unit Test (73)
10B 267
BUS → R1, R2, DATA, BUS → K
123 291
A0 → R1, A4 → R2, 16 → n
A0 contains Key Code
A4 contains TST code
117 279
BUS → A, SPA. A → RX, DATA
103 259
ADDRESS MASK → BUS
0E6 230
RX → SP
359 857
SPA ⊕ SPB → AX
22D 557
BUS → A, A ⊕ SPB → RX, AR, DATA, TMP
00A 010
F7FF → BUS, SET CR, BME
0F6 246
0 → AX, TCOK
0 CR≠3
1 CR=3
2DD 733
BUS → A, A → RX, DATA
0FB 251
2 → BUS
Idle 1 17/21A
Display Control Unit Error Code 2
Memory Tests 74/2DC


Memory Tests
(74,75,76)
R1=0,R2=4
2DC 732
RX→BUS, BUS→A, BUS→K, AR, DATA, TMRN
2CD 717
Write,SetCR
2BD 701
AX+2→RX,TCOK
0
1
18B3951
AX→RX
OBC 188
RX→SP, 0→AX
Capacity in A0
3OB 779
BUS→R1, R2, DATA
133 307
1→R1, 0→R2
Write FFFF until CR=3 or 32k
OAD 173
A→RX
176 374
P→AX
Test ≤ 32k
18A 394
0
RX→P, TZ
1
177 375
AX→RX
0F9 249
RX→P
0ED 237
P+SPB→RX
06D 109
P→AX,TZ
1
0
A75/099
If error display code is 3 or wrong address
Read and Test FFFF
098 152
BUS→A, A + 1→RX, DATA, TMRN
2B9 697
Read
2C9 713
1
AX + 2→RX, TZ
0
OF8 248
AX→RX
OE2 226
RX→SP
OEC 236
BUS→A, SPA, 0 ⊕ 0, A→RX, DATA
2EB 747
Code 3→BUS
17/21A
Idle 1


Memory Tests Cont'd
099 153
0 → AX
0CD 205
0 + AX + 1 → RX
2A9 681
RX → SP
067 103
RX → BUS, DATA, TMRN
04D 077
WRITE
2A6 678
AX + 2 → RX
2E7 743
RX → P
2ED 749
P + SPB → RX
26D 621
P → AX, TZ
1
0
11C 284
SPA → RX
11D 285
0 → AX
2E8 744
BUS → A, SPA + A → RX, DATA, TMRN
2D9 729
READ
0C9 201
AX + 2 → RX, TZ
1
0
106 262
AX → RX
74/0E2
107 263
RX → P
009 009
P + SPB → RX
00D 013
P → AX, TZ
0
1
2E9 745
0 → AX
2EA 746
SPA → RX
104 260
L
RX → SP, (199), n - 1, TCEZ
1
76/31D
0
31C 796
SPA → RX
Write the Pattern (067 to 11C)
Read and check the Pattern (2E8 to 00D)
Shifting the Pattern (2EA to 31C)

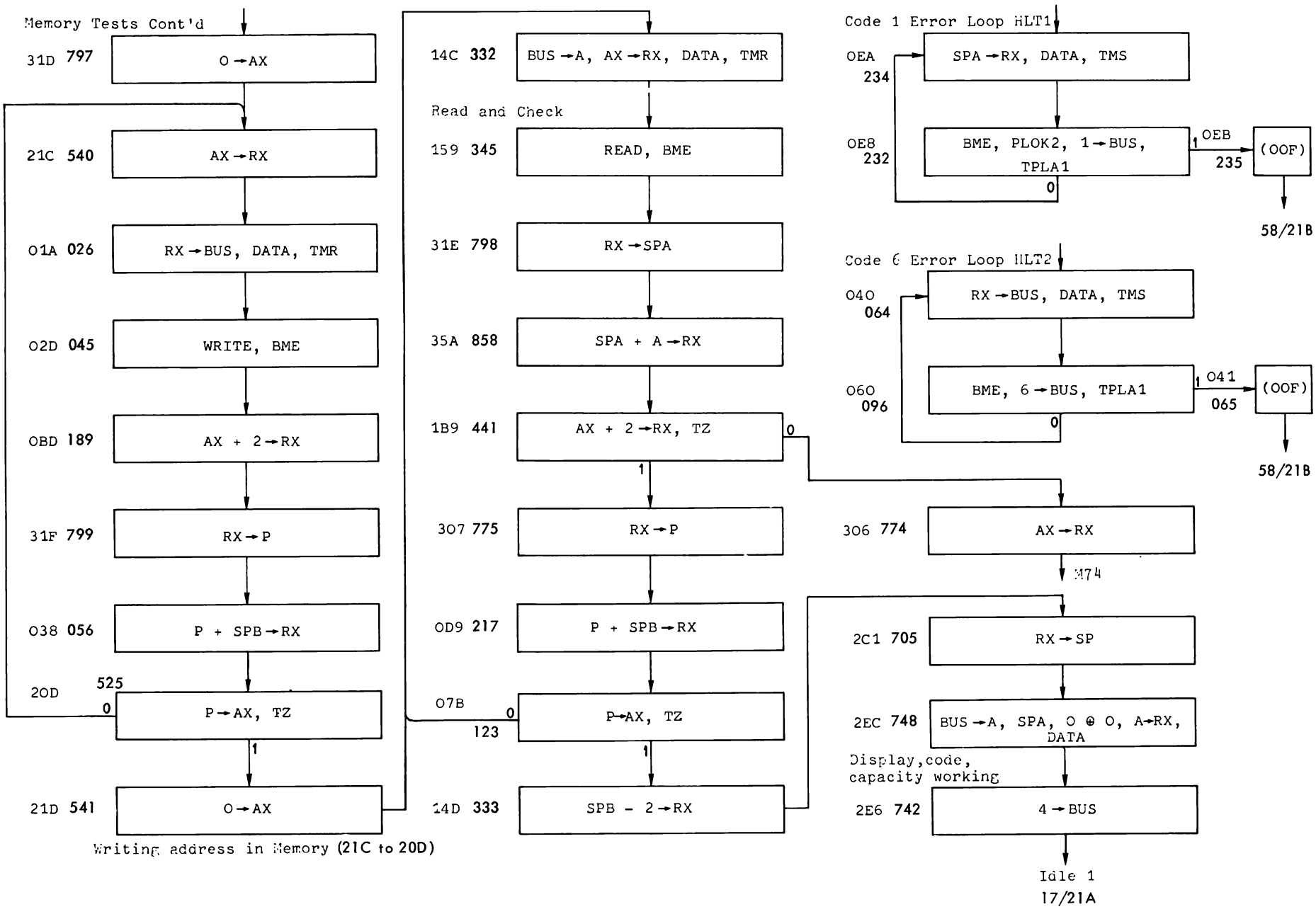

Memory Tests Cont'd
31D 797
0 → AX
21C 540
AX → RX
01A 026
RX → BUS, DATA, TMR
02D 045
WRITE, BME
0BD 189
AX + 2 → RX
31F 799
RX → P
038 056
P + SPB → RX
20D
525
0
P → AX, TZ
1
21D 541
0 → AX
Writing address in Memory (21C to 20D)
14C 332
BUS → A, AX → RX, DATA, TMR
Read and Check
159 345
READ, BME
31E 798
RX → SPA
35A 858
SPA + A → RX
1B9 441
AX + 2 → RX, TZ
0
1
307 775
RX → P
0D9 217
P + SPB → RX
07B
0
123
P → AX, TZ
1
14D 333
SPB - 2 → RX
Code 1 Error Loop HLT1
0EA
234
SPA → RX, DATA, TMS
0E8
232
BME, PLOK2, 1 → BUS, TPLA1
1
0
0EB
235
(OOF)
58/21B
Code 6 Error Loop HLT2
040
064
RX → BUS, DATA, TMS
060
096
BME, 6 → BUS, TPLA1
1
0
041
065
(OOF)
58/21B
306 774
AX → RX
M74
2C1 705
RX → SP
2EC 748
BUS → A, SPA, 0 ⊕ 0, A → RX, DATA
Display, code, capacity working
2E6 742
4 → BUS
Idle 1
17/21A

Table 5-7 Meanings of Terms and Signal Names Used in the Text

ACN	Acknowledge
ALU	Arithmetic Logic Unit
ALIMOS	Memory Power Supply
ANSF	Answer Flip-Flop
AR	CPU Condition Register set to Arithmetic Mode
BIO	Bus Input Output
BIOEKEY	Enable Bus Input Output lines at Control Panel KEYs
BIEC	Binary Encodes
BMTT	Bi-directional MOS/TTL Translator
BME	Bus to Memory Transfer
CADL	Left Single Shift
CADR	Right Single Shift
CGFN	Enable or Inhibit Internal Interrupts
CR	Condition Register (Planet)
CPU	Central Processor Unit
CPFN	Control Panel Interrupt
CPZON	Used for resetting the Break Logic of SCUZ
CU	Control Unit
CPT	Transmitter Clock Pulse
CPR	Receiver Clock Pulse
CMP	CPU Condition Register set to compare mode
DA	Device Address
DMA CU	Direct Memory Access Control Unit
DCTE	Data Circuit Termination Equipment
ENB	Enable Flip-Flop (Planet
FU	User Flip-Flop (Planet)
F/F	Flip-Flop
GF	General Flip-Flops
GPBS	General Purpose Bus S
GM	Microinstruction bits 17,18,19
GF int	General F/Fs of Planet User (FU), Enable (ENB), Interrupt (IR)
GF ext	General F/Fs for internal interrupts RTC, PFF, CPF, PIF
INTAD	Interrupt Address
INTVAL	Interrupt Validated
ITV	Interrupt Validated
IR	Interrupt Flip-Flop (Planet)
IOP	Input Output Processor
IOCU	Input Output Control Unit
IPL	Initial Program Loader
IS	Interrupt Signal (CPU)
IPS	Initial Program Loader Data Transfer Signal
I/O	Input/Output

Table 5-7 contd.

K	K Register of Planet
KL or KR	Right or left part of K Register
LOCMOS	Local Oxidation of Complementary Metal Oxide Semiconductor
LOG	CPU Condition Register set to logic mode
MAD	Memory Address Lines
MAX	Bootstrap ROM Address Lines
NA	Microinstruction bits 0-9 used to prepare next address
NAR	Microinstruction bits 10,11 used to prepare next address
P	Program Counter or Register (Spalu)
PIF	Program Interrupt Flip-Flop
PROM	Programmable Read Only Memory
Planet	Large Scale Integrated Circuit with Programmable Logic Arrays
PSW	Program Status Word
PLR	Program Level Register PLA Programmable Logic Array
PFF	Power Failure Interrupt
PUP	Control Panel
Plok	Planet Internal Signal
RAM	Random Access Memory
RAZ	Reset to Zero (mini-panel)
ROM	Read Only Memory
Reg	Register
RA	ROM Address Lines (bits 0-23)
RADA	ROM Address Lines (bits 24-39)
RAR	ROM Address Register
REPROM	Reprogrammable Read Only Memory
RTC	Real Time Clock
RCONT	ROM Contents bits 24-35. Input to Spalu to control operations
RSR	Receiver Shift Register (SCUZ)
RBR	Receiver Buffer Register (SCUZ)
RS	Microinstruction bits 12-16 used to generate internal Planet control signals.
Spalu	Large Scale Integrated Circuit with a Scratch Pad and Arithmetic Logic Unit.
SP	Scratch Pad
Stack	Part of Memory
SCEI	Scanning Interrupts
S-flag	Sign bit
SCUZ	Serial Control Unit type Z
SLL	Left Logical Shift
SLC	Left Circular Shift
SLQ	Left Division Shift
SRA	Right Arithmetic Shift

Table 5-7 contd.

SRC	Right Circular Shift
SRL	Right Logical Shift
SRM	Right Multiplication Shift
TCOK	Branch Condition Register
TS	Test Sign Bit TV Test Overflow TZ Test Zero
TMRN	Transfer Master to Memory
TMPN	Transfer Master to Peripheral Control Unit
TMEN	Transfer Master to External Register
TMSN	Transfer Master to Slave. Used to represent TMRN, TMPN, TMEN
TSMN	Transfer Slave to Master
TSR	Transmitter Shift Register (SCUZ)
TBR	Transmitter Buffer Register (SCUZ)
UART	Universal Asynchronous Receiver Transmitter (SCUZ)
V-flag	Overflow
Z-flag	Zero

5.93 Using the Test Tool

Plug CPSZ into extender card and connect the spring loaded connectors to the address register devices. The following tests may now be carried out.

- Checking Address Sequence — Select switch position PR; Connect connect clock inhibit; Push RUN (control panel); Push CL button to increment address register, each hex. address is displayed and the stop lamp remains on.
- Halt at Selected Address — Select switch position CYC; Connect clock inhibit; Select ROM address; Push RUN; Halts at selected address, stop lamp is on; Push CL button and the register runs up to selected address again.
- Displaying a Waveform — Select switch position CYC; Disconnect clock inhibit; Connect oscilloscope to SYNC position; Select ROM address; Push RUN; Each time that the selected address is passed, the waveform is displayed.
- Display Hexadecimal Address Only — Select switch position CYC; Disconnect clock inhibit; Push RUN; Address Register continues running, Test Tool does not affect operation, only displays the addresses.

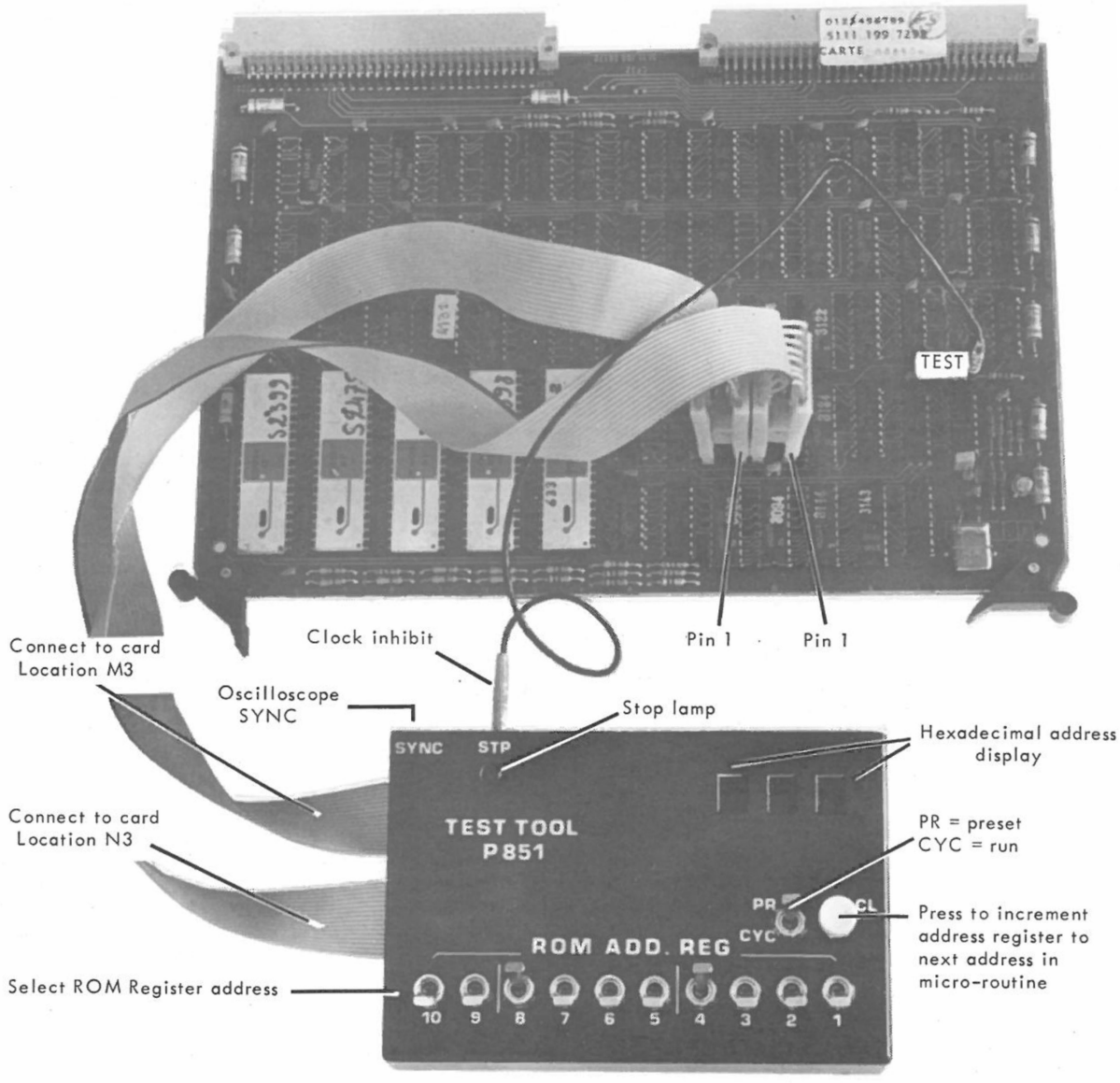


Figure 5-26 P851 TEST TOOL

USING THE TEST TOOL

Plug card CPSZ into the extender card and clip the tool connectors on the address register chips.

ADDRESS SEQUENCE TEST
Select switch position PR
Connect clock inhibit
Press RUN on the control panel
Press CL button to increment the address
Hexadecimal address is displayed and the stop lamp remains on.

HALT AT SELECTED ADDRESS TEST
Select switch position CYC
Connect clock inhibit
Select ROM address
Press RUN on the control panel
Halts at selected address - stop lamp on.
Press CL button to run up to selected address again.

DISPLAYING A WAVEFORM
Select switch position CYC
Disconnect clock inhibit
Connect oscilloscope sync to SYNC
Select ROM address
Press RUN on the control panel
Place oscilloscope probe on desired circuit pin
Waveform is displayed each time the selected address is passed.

DISPLAYING HEXADECIMAL ADDRESS
Select switch position CYC
Disconnect clock inhibit
Press RUN on the control panel
Address is displayed continuously.
CPU operation is not affected.
Address Register continues to run.

SECTION VI

MICRODIAGNOSTICS AND TROUBLESHOOTING

6.1 This section describes the tests performed by the CPU microdiagnostic routines and the error codes displayed by both the Mini-Panel and the Full Control Panel. If the Standard Test Programs (see Table 6-1 at the end of this section) either cannot be loaded or will not run, there are a number of Simple Test Programs which can be loaded direct into memory using the control panel switches. For systems that normally operate with a Mini-Panel there are a number of more sophisticated test programs, together with suggested methods of producing the program tape.

6.2 FAULT FINDING PROCEDURE

Figure 6-1 shows a flowchart of a fault finding procedure with references to the relevant pages of this service manual. It covers the most probable areas of a system breakdown and assumes that the fault is continuous. When a decision point is reached on the flowchart the engineer must choose the path most appropriate to the fault symptoms. Where the choice is not clear, first take the path that most easily isolates and tests one part of the system and then gradually include other parts of the system until the fault is isolated. For intermittant faults the same procedure can be used but must be repeated each time the symptoms occur (if it is impossible to simulate the fault). It should also be remembered that software not debugged properly can cause faults, which to the programmer appear to be hardware oriented. Therefore, before starting on the hardware, check with the operator if the fault occurs with all programs, with only certain programs, and if these programs are new to the system. If the standard test programs run correctly but the customer's programs do not run correctly, the Software Product Support group should be contacted.

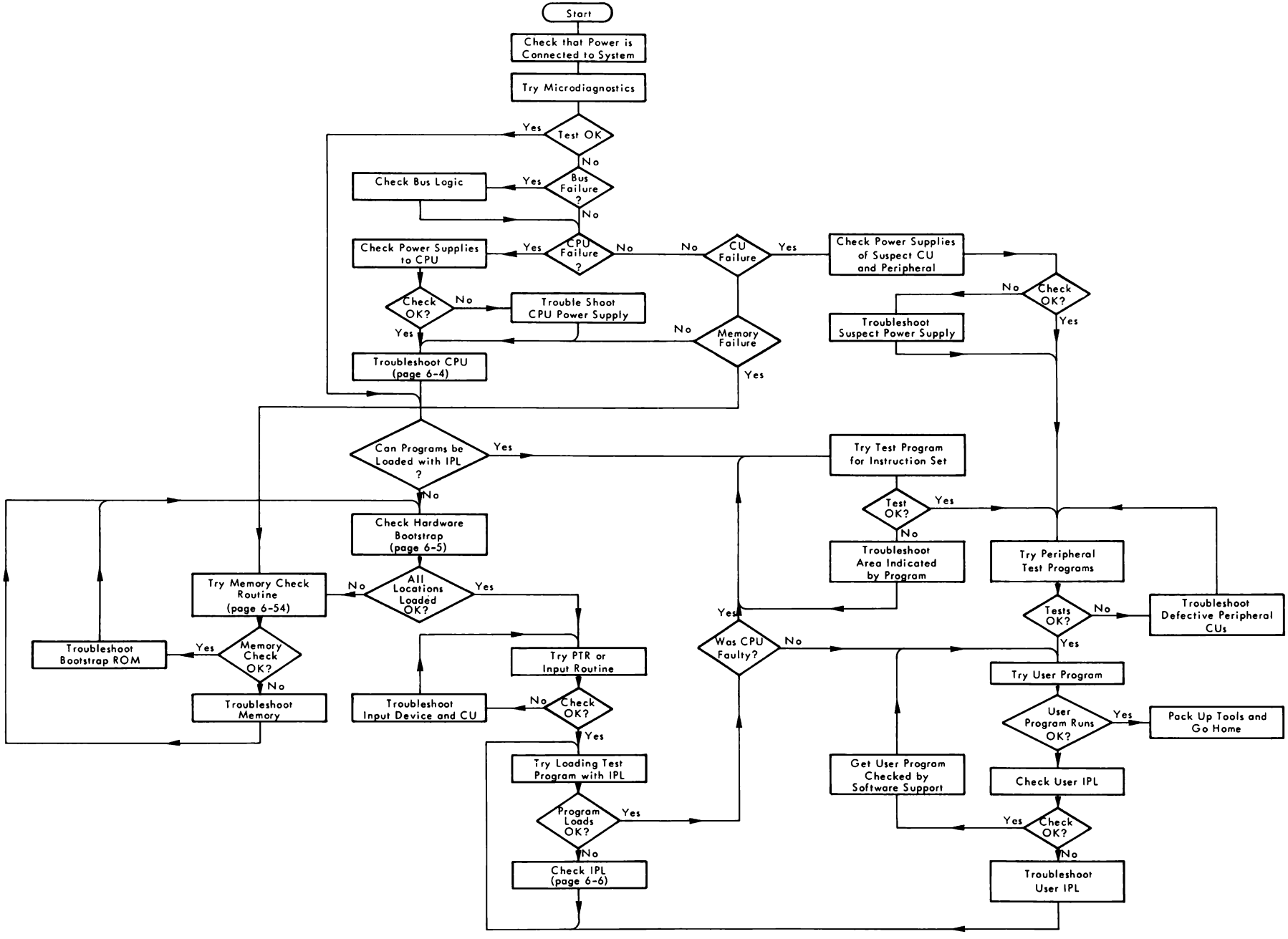


Figure 6-1 Fault Finding Flow Chart

6.3 MICRODIAGNOSTIC ROUTINES

These routines are run by a microprogram contained in the CPU control ROMs. The microprogram is initiated when the Safety Key (on either the Inter-Panel or Mini-Panel cards) is switched to the TEST position. The routines check the CPU, the GPBS, the data path of the CU that is used by the IPL, and the Memory. If an error is detected during a routine the microprogram jumps to a loop which displays an error code on DATA lamps 13 to 15 of the Control Panel or on the three Test Indicator LEDs of the Mini-Panel. When the routines have been performed without error the microprogram stops and displays hexadecimal 4 on the DATA lamps or Test Indicators; with the full panel the other lamps are used to display the memory size. NOTE: For systems fitted with a Full Control Panel the DATA switches must be set to address the CU used by the IPL.

6.4 CPU Tests

These tests are shown in Microinstruction Routines 71 and 72 in section V of this volume. The checks carried out by the routine are:

- The state of the Planet R1Z decoding.
- The correct operation of the Spalu flags Zero, Overflow, and Sign.
- That none of the Internal Bus lines are blocked.
- That none of the GPBS lines are forced.
- That operations by the Spalu will set the Condition Register into states 0, 2, and 1.
- That the Condition Register can be set to state 3 directly by Planet.
- That the Condition Register can be set to 0 from the Internal Bus and transferred to the Spalu RX register and is still equal to 0.
- That the operation of the Spalu data is correct by shifting a single one bit left and right across zeros.

6.5 If the GPBS test fails the microprogram jumps to a loop that displays hexadecimal 6 on the DATA lamps or the Test Indicators.

6.6 If an error is found in either the Spalu, Planet or Internal Bus the microprogram jumps to a loop that displays hexadecimal 1 on the DATA lamps or Test Indicators.

6.7 CU Test

During the condition register tests, the microprogram loads the TST instruction into register A0. This instruction is sent to CU whose address is indicated by the DATA switches of either the Full or Mini Control Panel. Normally the address on the switches should be the one used by the General Purpose Bootstrap Rom, but for systems fitted with a Dedicated Bootstrap Rom the address of any CU may be used. The routine for this test is shown in Microinstruction Routine 73 in section V of this volume. The TST instruction is sent to the addressed CU and the CPU checks the state of the Condition Register.

6.8 If the condition register is 3 an error condition has been found and the microprogram jumps to a loop that displays hexadecimal 2 on the DATA lamps or Test Indicators. For all other states of the condition register the microprogram starts the Memory Tests.

6.9 Memory Test

The routine for this test is shown in Microinstruction Routine 74 in section V of this volume. The checks carried out by the routine are:

- The memory size is calculated by writing the hexadecimal pattern FFFF into every memory location starting at 0 and continuing until either the Time OUT signal sets the condition register to 3 (less than 32K) or the memory address reaches 32K.
- The memory size is then stored in register A0.
- Then all memory locations are read and checked to see that the FFFF pattern has been correctly stored.
- Then all memory locations have a single one bit shifted left across zeros sixteen times; at the end of this test all locations contain zero.
- Then each location has its own address written into it and is check read.

6.10 If an error is found the microprogram jumps to a loop that displays hexadecimal 3 on the DATA lamps or Test Indicators; for systems fitted with a Full Control Panel, the address of the memory location in error is also displayed.

6.11 TROUBLESHOOTING THE CPU, CU, AND MEMORY CARDS

When the error code hexadecimal 1 is displayed a foult has been found in one of the tests on either the Spalu, Planet or Internal Bus. When the error code hexadecimal 6 is displayed a fault has been found on the GPBS. The following routines should enable the faulty element to be isolated.

6.12 Internal Bus Fault

If one of the Internal Bus lines is forced it should be possible to find the fault using the load register routine. First load the pattern FFFF — if any lamp is not lit the appropriate line is being forced high. If all the lamps are lit try the pattern 0000 — any lamp lit shows that the appropriate line is being forced low. If the fault does not show up on the control panel lights try the same thing using a data probe on the actual card tracks.

6.13 Spalu Flags and Planet Faults

The Spalu Flags V (Overflow), Z (Zero Result), and S (Sign Bit) are used to set the Planet Condition Register as the result of certain instructions, e.g. Arithmetic Instructions. If after an error code hexadecimal 1 has been displayed, and the result of the Internal Bus tests have shown no forced lines, try the following routine to find out if the flags and CR bits are being activated correctly:

- Load any memory address in register A0 (Program Counter)
- Set the hexadecimal pattern 1101 on the DATA switches and push the LM button to load it into memory (this loads the instruction ADK A1,1 into the selected memory location).
- Set the hexadecimal pattern 207F on the DATA switches and push the LM button (this loads a Halt Instruction into memory).
- Set the hexadecimal pattern FFFF on the DATA switches, set the 8 4 2 1 switches to select register A1, then push the LR to load the pattern into the register.
- Load the memory address that contains the ADK instruction into register A0.
- Push the RUN button. The routine will execute the instruction and stop at the halt instruction.
- Push the RST button and read the state of the Condition Register (bits 6 and 7), for the pattern above the state should be 00 indicating a Zero result in register A1.
- Repeat the routine using the hexadecimal pattern 80 — the result should be 01 indicating a positive Sign bit in register A1.
- Repeat the routine using the hexadecimal pattern FFFE — the result should be 10 indicating a negative Sign bit in register A1.
- Repeat the routine using the hexadecimal pattern 7FFF — the result should be 11 indicating an Overflow condition in register A1.

6.14 If the expected result is not obtained for any of the patterns repeat the routine using that pattern and use a Data Probe to check whether the erroneous result is caused by the Spalu Flags or Planet.

6.15 Data Path Test

If all the previous tests have been performed correctly try the following routines to check the Spalu data paths:

- Select a memory address on the DATA switches and load it into register A0.
- Set the DATA switches to hexadecimal pattern 3941 and push the LM button. This loads the instruction SLL A1,1 into memory.
- Set the DATA switches to hexadecimal pattern 207F and push the LM button to load the Halt instruction.
- Set the DATA switches to hexadecimal pattern 5F06 and push the LM button to load the RB instruction.
- Set the DATA switches to hexadecimal pattern 0001. Set the 8 4 2 1 switches to address register A1 and push the LR button to load the pattern into the register.
- Load the memory address that contains the SLL instruction into register A0.
- Set the 8 4 2 1 switches to address register A1.
- Push the RUN button. The routine will execute the instruction and stop at the Halt instruction.
- Push the RR button and check that the contents of register A1 have been shifted one place left.

- Alternately push the RUN and RR buttons and check that the contents of register A1 are shifted one place left for each operation of the buttons until the register contains zeros.
- Repeat the routine using an SRL instruction hexadecimal pattern 3961 and test pattern 1000 in register A1.
- Repeat the routine using a DLC instruction hexadecimal pattern 38C1 and test pattern 0001 loaded in register A2. For this test the contents of both register A1 and A2 should be checked for left shifts.
- Repeat the routine using a DRC instruction hexadecimal pattern 38E1 and test pattern 0001 loaded in register A2. For this test the contents of both register A1 and A2 should be checked for right shifts.

6.16 GPBS Fault

If the error code hexadecimal 6 is displayed it indicates that one or more of the BIO lines is being forced either high or low. To find the line or lines being forced use the same routines described for an Internal Bus fault.

6.17 Control Unit Fault

If the CU test fails the error code hexadecimal 2 is displayed and the contents of the Condition Register will be hexadecimal 3 indicating device address unknown or that timeout has occurred. Check that the CU address set on the DATA switches at the start of the microdiagnostic routine corresponds with the address set up by the U links on the CU card. If the two addresses correspond use the following routine and check the state of the MAD lines and timing signals:

- Load any memory address in register A0.
- Set the DATA switches to either of two hexadecimal patterns: 4980 plus the device address for a TST instruction, or 4180 plus the device address for a CIO Halt Instruction (these two instructions are always accepted by the the CU irrespective of the sequensor state); then push the LM button to load the instruction in memory.
- Set the DATA switches to hexadecimal pattern 5F04 then push the LM button to load the RB instruction into memory.
- Load the memory address that contains the I/O instruction into register A0.
- Push the RUN button. The routine will execute the instruction repeatedly (because of the RB instruction) until stopped by pushing the INST button.

Whilst the routine is running check the state of the MAD lines and timing signals both on the Backpanel and on the CU card using a Data Probe. If the fault cannot be found on the CU card addressed by the Bootstrap Rom, remove the other CU cards one at a time until the fault disappears then troubleshoot the last card removed. If the fault persists with all other CU cards removed check the printed circuit backpanel and the operation of the CPU Condition Register.

6.18 Memory Fault

If the microdiagnostic Memory Test fails the error code hexadecimal 3 is displayed with the current memory address. If the memory test program BZRAM can be loaded with the IPL routine, this should enable the fault to be diagnosed and repaired. For faults where the standard memory test program cannot be loaded try one of the programs suggested in this section.

6.19 BOOTSTRAP ROM FAULT

The Bootstrap is contained in a 256 4-bit word ROM on the CPU card and when the IPL button (on the Full Panel) or the START button (on the Mini-Panel) is pushed, the contents of the ROM are loaded into the first 64 words of memory (hexadecimal 0000 to 007E). Once the ROM contents are loaded, the Bootstrap starts to load the software Lowcore IPL into memory starting at hexadecimal location 80. If you are unable to load the IPL or one of your test programs using the IPL routine, check that the Bootstrap is being loaded correctly using the following routine:

- Put all the DATA switches to the zero position.
- Make sure that all the peripherals are switched OFF.
- Push the INST button, the MC button, and then the IPL button.
- Set the 8 4 2 1 switches to address register A0 and push the LR button.
- Push the RM button and compare displayed contents of the memory with either Table 5-1 or Table 5-2 of this volume. Repeat this operation until the first 64 locations have been checked.

6.20 If the contents of the Bootstrap ROM are correct try to verify the Software IPL.

6.21 SOFTWARE IPL

The type of IPL used by the customer will depend on the system configuration, but all IPLs perform a similar function in that they load programs from a chosen device. The three IPL listings given are all used by the Standard Test Programs: the IPL52S is a Stand Alone program loader which prints out the program IDENT, the IPL52H is similar except it has a Halt feature, and the IPL52N is the same as the IPL52S except it does not print out the program IDENT. Only the IPL52S will be described here and like all IPLs it is in two parts, the Low Core IPL and the High Core IPL.

6.22 Loading the IPL

The input device used to load the IPL will depend on the type of media used to hold the IPL data i.e. for Paper Tape either the PTR or ASR reader. First the parameters for the type of transfer to be handled by the Bootstrap must be loaded on the DATA switches for use by the Bootstrap and the Low Core and High Core IPLs. The switches have the following meaning:

Bit	Meaning
0 = 1	IPL is loaded in 4x4 format
1 = 1	IPL is loaded from Disc. If the Disc is selected then bit 2 has the following meaning:
2 = 0	Fixed Head Disc
2 = 1	Moving Head Disc
3 = 0	The input device is connected to the IOP Channel
3 = 1	The device is connected to the Program Channel
If the input device is Disc, Magnetic Tape, or Cassette Tape bits 4 to 7 are used by the Read Command which is sent to the device. For Magnetic and Cassette Tape these bits are sent out on BIO12 to 15, and for the Disc these bits are sent out on BIO10 to 13. The codes for the three devices are:	
4 to 7 = 0010	The Read Command for the Magnetic Tape
4 to 7 = 0011	The Sector Address for the Disc Read Command
4 to 7 = 0111	The Read a Block Command for the Cassette Tape
8 = 0	Single Device Controller
8 = 1	Multi Device Controller
9 = 1	The CDD Disc is the input device
10 to 15	The device address

For example with the DATA switches set for hexadecimal 63C2 the IPL will be loaded from the Moving Head CDD Disc whose address is 2 starting to read at sector 3. Once the parameters have been set on the DATA switches push the RST, MC, and IPL buttons. The contents of the Hardware Bootstrap ROM will be loaded into the first 64 locations of memory and the Bootstrap will then load the Low Core IPL.

6.23 Low Core IPL

The Bootstrap loads 80 characters then jumps to location hexadecimal 84 which is the start address of the Low Core IPL. The Low Core IPL has three functions:

- It computes the size of memory and subtracts 400 words which gives the first address of the High Core IPL.
- It modifies the contents of some of the Bootstrap instructions then jumps back to Bootstrap (as necessary) to load the High Core IPL.
- When loading is complete it jumps to the start address of the High Core IPL.

6.24 High Core IPL

Once loaded by the Low Core IPL and Bootstrap the High Core IPL loads the Object Program and enables:

- The peripheral, whose address is on the DATA switches (and has been stored in register A15 by the Bootstrap), to be used as the input device.
- The IDENT of the program to be printed out on Operator's peripheral.
- The program to be loaded into its designated place in memory (overwriting the Bootstrap and Low Core IPL).
- The program to be started either automatically or manually.

6.25 Verification of the Low Core IPL

This can only be carried out if loading stops before the message IDENT 'Name' has been typed out on the Operator's peripheral. To verify that loading is taking place (even if the input device is apparently operating, the Bootstrap may not actually be loading the IPL data) try the following routine:

- Stop the CPU by pushing the INST button.
- Load the hexadecimal value 84 into location 84 (the start address of the Low Core IPL) and try to load the IPL again. If the CPU loops on location 84, the Bootstrap is not loading the IPL and the status of peripheral will be found in register A7.
- If loading takes place beyond location 84 but still stops before the Ident is printed, load hexadecimal 80 into register A0.
- Push the INST button and check the contents of memory by comparing the data displayed against the IPL listing. Repeat until each location of the Low Core IPL has been checked or the faulty location found.

6.26 Verification of the High Core IPL

This can only be carried out if the messages EOS, EOF have not been typed out on the Operator's peripheral. If the program loops in the High Core IPL the loading address +2 will be found in register A11, otherwise compute the loading address by subtracting hexadecimal 400 from the memory size, then use the following routine to find where the loading aborts:

- Stop the CPU by pushing the INST button.
- Load the start address of the High Core IPL into register A0.
- Push the INST button and compare the displayed memory contents with the IPL listing.

6.27 If both the Low Core and High Core IPLs and the program appear to be loading correctly but the program does not start, check the contents of register A0 to find out where the CPU has stopped. This check should indicate whether the program has tried to start or if the fault is due to the IPL. Another possible fault that points to a malfunction of the IPL is if the Operator's peripheral or its CU is malfunctioning.

IPL52S

```
00000                                  IDENT   IPL52S
00001                                  ENTRY   IPL88
00002                                  ENTRY   IPL44
00003                          *
00004                          *                       RELIER IPL GENERATION
00005          000C            NBREC   EQU     12
00006                          *
00007          0000            BIPLPR  EQU     *
00008          0000            IPL88   EQU     *
00009  0000    010C                    LDK     A1,NBREC
00010  0002    8220                    LDK,L   A2,BIPL-80
       0004    00B0  R
00011  0006    0785                    LDK     A7,/85                  BASIC WRITE
00012  0008    80A0                    LDKL    A8,DECB
       000A    001E  R
00013          000C            PUNCH   EQU     *
00014  000C    1250                    ADK     A2,80
00015  000E    8241                    ST      A2,DECBUF               SET BUFF ADDR
       0010    0020  R
00016  0012    2804                    LKM                             PUNCH ONE RECORD
00017  0014    0001                    DATA    1
00018  0016    1901                    SUK     A1,1                    COUNT DONE ?
00019  0018    590E                    RB(1)   PUNCH                   NO
00020  001A    2804                    LKM                             YES
00021  001C    0003                    DATA    3                       EXIT
00022                          *
00023          001E            DECB    EQU     *
00024  001E    0003                    DATA    3                       FILE CODE
00025  0020                    DECBUF  RES     1
00026  0022    0050                    DATA    80
00027  0024    0000                    DATA    0
00028  0026    0000                    DATA    0
00029  0028    0000                    DATA    0
00030                                  EJECT
00031                                  RORG    BIPLPR+/100
00032                          *
00033          0100            BIPL    EQU     *
00034                          *                       1ST RECORD = LOW CORE IPL
00035                          *                       LOW CORE IPL,1 RECORD,LOADED AT /80 AND
00036                          *                       STARTED AT /84
00037                          *
00038  0100    FFFF            LDFLG   DATA    /FFFF                   LDFLG = -1 IF JUST LOADED
00039  0102    0000                    DATA    0                       NOT USED
00040          0104            IPL     EQU     *
00041                          *                       ADDR OF IPL = BASE ADDR
00042  0104    92A0                    LDKL    A10,/84
       0106    0084
00043                          *                       CHECK IF JUST LOADED OR NOT
00044  0108    904B                    IM      LDFLG-IPL,A10
       010A    FFFC
00045  010C    511E                    RF(1)   IPL100                  NO,NOT THE 1ST TIME
```

```
00046                          *
00047                          *
00048                          *                                YES,COMPUTE HIGH CORE LIMIT
00049                          *
00050  010E  8720              LDK.L   A7,/5555                 PATTERN
       0110  5555
00051  0112  81A0              LDKL    A9,/FC00                 HIGH CORE -/400
       0114  FC00
00052        0116  TPL010      EQU     *
00053  0116  8727              STR     A7,A9
00054  0118  FF26              CWR*    A7,A9                    IF THE LOCATION (A9) EXISTS,(A7)=((A9))
00055  011A  5006              RF(0)   IPL020                   MATCH
00056  011C  99A0              SUKL    A9,/2000                 NO,NEXT LOWER BLOCK OF 4K
       011E  2000
00057  0120  5F0C              RB      IPL010
00058        0122  IPL020      EQU     *
00059  0122  8386              LDR     A11,A9                   SAVE BASE ADDR OF HIGH CORE IPL
00060  0124  93A0              ADKL    A11,2                    START ADD.
       0126  0002
00061  0128  8486              LDR     A12,A9                   SAVE LOAD ADD.
00062  012A  5704              RF      IPL110
00063        012C  IPL100      EQU     *
00064  012C  94A0              ADKL    A12,80
       012E  0050
00065                          * LOAD ADDRESS OF NEXT RECORD
00066        0130  IPL110      EQU     *
00067  0130  871F              LDR     A7,A15                   RESTORE CIO INST IN BOOT
00068  0132  273F              ANK     A7,/3F
00069  0134  9720              ADK.L   A7,/41C0                 CIO INSTRUCTION
       0136  41C0
00070  0138  8720              STR     A7,A3
00071                          *
00072  013A  0557              LDK     A5,/57                   CHANGE LOC. /5A OF BOOT IN ORDER TO
00073  013C  E541              SC      A5,/5A                   CANCEL LEADING CHARACTER FLAG
       013E  005A
00074                          *                               RE INITIALIZE A5,A6
00075  0140  0550              LDK     A5,80                    # OF CHARACTERS
00076  0142  8612              LDR     A6,A12                   LOAD ADDR.
00077                          *                               CHECK IF COUNT DONE
00078  0144  904B              IM      CNTFLG-IPL,A10
       0146  0048
00079  0148  890E              ABR(1)  A11
00080  014A  0F42              AB      /42                      START BOOT ABAIN
00081        014C  CNTFLG      EQU     *                        = NUMBER OF RECORDS OF HIGH CORE IPL
00082  014C  FFF5              DATA    1-NBREC
00083  014E  0000              DATA    0
00084  0150  0000              DATA    0
00085  0152  0000              DATA    0
00086  0154  0000              DATA    0
00087  0156  0000              DATA    0
00088  0158  0000              DATA    0
00089  015A  0000              DATA    0
00090  015C  0000              DATA    0
00091  015E  0000              DATA    0
00092  0160  0000              DATA    0
00093  0162  0000              DATA    0
00094  0164  0000              DATA    0
00095  0166  0000              DATA    0
```

```
00096  0168  0000          DATA    0
00097  016A  0000          DATA    0
00098  016C  0000          DATA    0
00099  016E  0000          DATA    0
00100  0170  0000          DATA    0
00101  0172  0000          DATA    0
00102  0174  0000          DATA    0
00103  0176  0000          DATA    0
00104  0178  0000          DATA    0
00105  017A  0000          DATA    0
00106  017C  0000          DATA    0
00107  017E  0000          DATA    0
00108  0180  0000          DATA    0
00109  0182  0000          DATA    0
00110  0184  0000          DATA    0
00111  0186  0000          DATA    0
00112  0188  0000          DATA    0
00113  018A  0000          DATA    0
00114  018C  0000          DATA    0
00115  018E  0000          DATA    0
00116  0190  0000          DATA    0
00117  0192  0000          DATA    0
00118  0194  0000          DATA    0
00119  0196  0000          DATA    0
00120  0198  0000          DATA    0
00121  019A  0000          DATA    0
00122  019C  0000          DATA    0
00123  019E  0000          DATA    0
00124  01A0  0000          DATA    0
00125  01A2  0000          DATA    0
00126  01A4  0000          DATA    0
00127  01A6  0000          DATA    0
00128  01A8  0000          DATA    0
00129  01AA  0000          DATA    0
00130  01AC  0000          DATA    0
00131  01AE  0000          DATA    0
00132  01B0  0000          DATA    0
00133  01B2  0000          DATA    0
00134  01B4  0000          DATA    0
00135  01B6  0000          DATA    0
00136  01B8  0000          DATA    0
00137  01BA  0000          DATA    0
00138  01BC  0000          DATA    0
00139  01BE  0000          DATA    0
00140  01C0  0000          DATA    0
00141  01C2  0000          DATA    0
00142  01C4  0000          DATA    0
00143  01C6  0000          DATA    0
00144  01C8  0000          DATA    0
00145  01CA  0000          DATA    0
00146  01CC  0000          DATA    0
00147  01CE  0000          DATA    0
00148  01D0  0000          DATA    0
00149  01D2  0000          DATA    0
00150  01D4  0000          DATA    0
00151  01D6  0000          DATA    0
00152  01D8  0000          DATA    0
00153  01DA  0000          DATA    0
```

```
00154  01DC  0000              DATA    0
00155  01DE  0000              DATA    0
00156  01E0  0000              DATA    0
00157  01E2  0000              DATA    0
00158  01E4  0000              DATA    0
00159  01E6  0000              DATA    0
00160  01E8  0000              DATA    0
00161  01EA  0000              DATA    0
00162  01EC  0000              DATA    0
00163  01EE  0000              DATA    0
00164  01F0  0000              DATA    0
00165  01F2  0000              DATA    0
00166  01F4  0000              DATA    0
00167  01F6  0000              DATA    0
00168  01F8  0000              DATA    0
00169  01FA  0000              DATA    0
00170  01FC  0000              DATA    0
00171  01FE  0000              DATA    0
00172  0200  0000              DATA    0
00173  0202  0000              DATA    0
00174  0204  0000              DATA    0
00175                          EJECT
00176                                                  HIGH CORE IPL
00177         0000     PTR     EQU     0              DEVICE ADDR WILL BE INITIALIZED BY HCIPL
00178         0010     ASR     EQU     /10
00179         0001     S       EQU     1
00180         0000     H       EQU     0
00181         0001     BOM     EQU     1
00182         0000     SA      EQU     0
00183                  *
00184         0000     MODE    EQU     SA             FOR MESSAGE  OUTPUT
00185                          RORG    BIPL+80        1ST RECORD OF HIGH CORE IPL
00186         0150     HCIPL1  EQU     *              1ST WORD OF THE CURRENT RECORD
00187  0150   FEFE             DATA    /FEFF
00188         0152     HCIPL   EQU     *
00189  0152   85A0             LDKL    A13,OUTMSG-HCIPL
       0154   00EA
00190  0156   958E             ADR     A13,A11        LOAD A13 WITH ADDR OF OUTMSG ROUTINE
00191  0158   868E             LDR     A14,A11
00192  015A   96A0             ADKL    A14,CFZON-HCIPL
       015C   00E8
00193                  *                ADDR OF STACK
00194  015E   871E             LDR     A7,A15         GET DEVICE ADDR
00195  0160   87CF             ST      A15,SAVA15-HCIPL,A11  SAVE  4*4  FLAG
       0162   0354
00196  0164   5606             RF(6)   *+8
00197  0166   904F             IM      COREND+2-HCIPL,A11
       0168   0358
00198  016A   5C06             RB(4)   *-4
00199  016C   273F             ANK     A7,/3F
00200                  *                        INIT IO INSTRUCTIONS
00201  016E   974F             ADS     A7,CIOPTR-HCIPL,A11
       0170   0142
00202  0172   974F             ADS     A7,INP2-HCIPL,A11
       0174   0188
00203  0176   974F             ADS     A7,END2-HCIPL,A11
       0178   01CE
00204  017A   974F             ADS     A7,END1-HCIPL,A11
```

```
        017C  0208
00205   017E  974F              ADS       A7,INR44-HCIPL,A11
        0180  01A2
00206   0182  811E              LDR       A1,A15
00207   0184  39C8              SLC       A1,8                    TEST MULTI-SINGLE DEVICE CONT,
00208   0186  5602              RF(6)     HCIPL2
00209   0188  270F              ANK       A7,/F                   MULTI DEVICE CONTROLLER
00210         018A      HCIPL2  EQU       *
00211   018A  974F              AD,S      A7,SSTMLX-HCIPL,A11
        018C  014E
00212   018E  974F              ADS       A7,SSTPR2-HCIPL,A11
        0190  018C
00213   0192  974F              ADS       A7,SSTPTR-HCIPL,A11
        0194  01D0
00214   0196  974F              ADS       A7,SSTPR1-HCIPL,A11
        0198  020A
00215   019A  3F41              SLL       A7,1
00216   019C  974F              AD,S      A7,WER1-HCIPL,A11       INIT WER/RER INST
        019E  013A
00217   01A0  974F              AD,S      A7,WER2-HCIPL,A11
        01A2  013C
00218   01A4  974F              AD,S      A7,RER-HCIPL,A11
        01A6  017C
00219   01A8  57AC              RF        CKOK
00220   01AA  4F42      SYSMSG  DATA      'OBJCT TAPE ON RE'
        01AC  4A43
        01AE  5420
        01B0  5441
        01B2  5045
        01B4  204F
        01B6  4E20
        01B8  5245
00221   01BA  4144              DATA      'ADER, THINK OF B'
        01BC  4552
        01BE  2E20
        01C0  5448
        01C2  494E
        01C4  4B20
        01C6  4F46
        01C8  2042
00222   01CA  4153              DATA      'ASE 1'
        01CC  4520
        01CE  2120
00223   01D0  0D0A              DATA      /0D0A
00224         01D2      STAD    EQU       *
00225   01D2  4543      ECMSG   DATA      'EC'
00226   01D4  0D0A              DATA      X'0D0A'
00227   01D6  4F56      OFLMSG  DATA      'OVFR'
        01D8  4652
00228   01DA  0D0A              DATA      /0D0A
00229   01DC          BUFF      RES       39
00230   022A  FFFF              DATA      /FFFF
00231   022C  0000      MASTFG  DATA      0
00232   022E  0000      SAVBAS  DATA      0
00233   0230                    RES       6                       * STACK  AREA
00234         023A      CFZON   EQU       *-2
00235         023C      OUTMSG  EQU       *
```

```
00236  023C  0304            LDK     A3,4
00237  023E  43D0            CIO     A3,S,ASR
00238  0240  E324            LCR     A3,A1
00239  0242  4310    OTR     OTR     A3,0,ASR
00240  0244  5C04            RB(4)   *-2
00241  0246  1101            ADK     A1,1
00242  0248  1A01            SUK     A2,1
00243  024A  5C0C            RB(4)   OTR-2
00244  024C  0300            LDK     A3,0
00245  024E  4390            CIO     A3,H,ASR
00246  0250  48D0            SST     A3,ASR
00247  0252  5C04            RB(4)   *-2
00248  0254  F03A            RTN     A14
00249                 *
00250                 *
00251         0256    CKOK    EQU     *
00252  0256  81A0            LDK,L   A9,0                    SET LOADING ADDRESS
       0258  0000
00253                 *
00254                 *
00255                 *                                 GO LOAD SYSTEM
00256  025A  84A0            LDK,L   A12,MNLO-HCIPL
       025C  029C
00257  025E  948F            ADR     A12,A11
00258  0260  F693            CFR     A14,A12
00259                 *
00260                 *
00261  0262  810A            LDR     A1,A10                  TEST START ADDRESS
00262  0264  8C04            ABR(4)  A1                      EOS  OR EOF HAS BEEN READ
00263  0266  207F            HLT                             NO START ADDRESS
00264  0268  0000    BADDR   DATA    0
00265  026A  0300    RAFL    LDK     A3,0
00266  026C  0700            LDK     A7,0
00267  026E  8520            LDK,L   A5,BUFF-HCIPL
       0270  008A
00268  0272  950E            ADR     A5,A11
00269  0274  80CE            LD      A8,SAVA15-HCIPL,A11
       0276  0354
00270  0278  5610            RF(6)   INPA
00271  027A  0604            LDK     A6,4                    4*4 SO ASR
00272  027C  0111            LDK     A1,/11                  SEND X-ON
00273  027E  46D0            CIO     A6,S,ASR
00274  0280  4110            OTR     A1,0,ASR
00275  0282  5C04            RB(4)   *-2
00276  0284  4290            CIO     A2,H,ASR
00277  0286  4AD0            SST     A2,ASR
00278  0288  5C04            RB(4)   *-2
00279         028A    INPA    EQU     *
00280  028A  0650            LDK     A6,80                   LENGTH
00281  028C  7600    WER1    WER     A6,0                    INIT MLX WHATEVER
00282  028E  7501    WER2    WER     A5,1                    CHANNEL IS
00283  0290  8602            LDR     A6,A8
00284  0292  3E68            SRL     A6,8
00285         0294    CIOPTR  EQU     *
00286  0294  46C0            CIO     A6,S,PTR                AVAILABLE ON THE PAPER READER
00287  0296  5C04            RB(4)   *-2
00288  0298  824E            LD      A2,SAVA15-HCIPL,A11
       029A  0354
```

```
00289  029C  3AC3            SLC      A2,3              MLX ?
00290  029E  523A            RF(2)    INP2              NO
00291        02A0      SSTMLX EQU      *
00292  02A0  49C0            SST      A1,PTR
00293  02A2  5C04            RB(4)    *+2
00294        02A4      TMTEST EQU      *
00295  02A4  A120            ANKL     A1,/1007          TAPE MARK OR FATAL ERROR ?
       02A6  1007
00296  02A8  501A            RF(0)    MLX1              NO
00297  02AA  2107            ANK      A1,/7             FATAL ERROR ?
00298  02AC  5486            RF(4)    ECCLS             YES
00299                *
00300  02AE  8120            LDK.L    A1,'↕E'           PUT EOF IN BUFF
       02B0  3A45
00301  02B2  8135            STR      A1,A5
00302  02B4  8120            LDK.L    A1,'OF'
       02B6  4F46
00303  02B8  8155            ST       A1,2,A5
       02BA  0002
00304  02BC  0704            LDK      A7,4              UPDATA POINTRERS
00305  02BE  951C            ADR      A5,A7
00306  02C0  579E            RF       PRASCI
00307                *
00308  02C2  5F5A      RAFL3 RB       RAFL              RELAY TOWARDS RAFL
00309                *
00310        02C4      MLX1  EQU      *
00311  02C4  E134            LCR      A1,A5
00312  02C6  E920            CWK      A1,/18            IS IT ASCII
       02C8  0018
00313  02CA  55EA            RF(5)    RELAY             NO
00314  02CC  0750            LDK      A7,80             YES
00315  02CE  7E00      RER   RER      A6,0              READ REMAINING LENGTH
00316  02D0  A620            ANK.L    A6,/FFF
       02D2  0FFF
00317  02D4  9F18            SUR      A7,A6             COMPUTE TRANSMITTED LGT
00318  02D6  951C            ADR      A5,A7             UPDATE BUFF POINTER
00319  02D8  5786            RF       PRASCI
00320  02DA  4A00      INP2  INR      A2,0,PTR
00321  02DC  50DA            RF(0)    SWITCH            INR HAS BEEN ACCEPTED ;
00322                *                                  PROCESS THE CHARTER
00323  02DE  4AC0      SSTPR2 SST     A2,PTR            TRY A SST
00324  02E0  5C08            RB(4)    INP2              TRY AGAIN INR WHEN SST REFUSED
00325  02E2  8108            LDR      A1,A2
00326  02E4  1300            ADK      A3,0              CHECK WETHER ASCII OR OBJECT
00327  02E6  5278            RF(2)    PRASCI            RECORD WAS ASCII ; PRINT IT
00328  02E8  0600            LDK      A6,0              STORE A NON ASCII CHARACTER AT
00329  02EA  8635            STR      A6,A5             THE END OF BUFFER ,
00330                *                                  TO BE USED IN MX1
00331  02EC  5F4A            RB       TMTEST
00332        02EE      OBJINP EQU      *
00333  02EE  8082            LDR      A8,A8
00334  02F0  560C            RF(6)    EIGHT             8=8 DO NOT INPUT 2ND CHARACTER
00335  02F2  220F            ANK      A2,/F
00336  02F4  4E00      INR44 INR      A6,0,PTR
00337  02F6  5C04            RB(4)    *+2
00338  02F8  260F            ANK      A6,/F
00339  02FA  3A44            SLL      A2,4
00340  02FC  9218            ADR      A2,A6             JOIN THE TWO HALF CHARACTERS
```

```
00341          02FE   EIGHT   EQU     *
00342  02FE    EF04           CWR     A7,A1
00343  0300    501E           RF(0)   END2
00344  0302    E235           SCR     A2,A5
00345  0304    B408           XRR     A4,A2
00346  0306    1501           ADK     A5,1
00347  0308    EF20           CWK     A7,1
       030A    0001
00348  030C    5408           RF(4)   OBJIN1
00349  030E    0400           LDK     A4,0
00350  0310    8108           LDR     A1,A2
00351  0312    9108           ADR     A1,A2
00352  0314    1103           ADK     A1,3
00353  0316    1701   OBJIN1  ADK     A7,1
00354  0318    5F40   INP2R   RB      INP2
00355  031A    2207   FIRST   ANK     A2,/7
00356  031C    0150           LDK     A1,80
00357  031E    5F22           RB(7)   EIGHT
00358  0320    4280   END2    CIO     A2,H,PTR
00359          0322   SSTPTR  EQU     *
00360  0322    49C0           SST     A1,PTR
00361  0324    5C04           RB(4)   *-2
00362  0326    8082           LDR     A8,A8                   IF ASR WAIT
00363  0328    5606           RF(6)   *+8
00364  032A    904F           IM      COREND+2-HCIPL,A11      WAIT
       032C    0358
00365  032E    5C06           RB(4)   *-4
00366  0330    24FF           ANK     A4,/FF
00367  0332    50D4           RF(0)   PROL01+2
00368          0334   ECCLS   EQU     *
00369  0334    0180           LDK     A1,ECMSG-HCIPL
00370  0336    910E           ADR     A1,A11
00371  0338    0204           LDK     A2,4
00372  033A    84A0           LDK,L   A12,OUTMSG-HCIPL
       033C    00FA
00373  033E    948E           ADR     A12,A11
00374  0340    F693           CFR     A14,A12
00375  0342    207F   STOP    HLT
00376  0344    5FDC           RB(7)   RAFL
00377  0346    EA20   ASCINP  CWK     A2,/0D
       0348    000D
00378  034A    500E           RF(0)   END1
00379  034C    EF20           CWK     A7,68
       034E    0044
00380  0350    5878           RB(0)   INP2
00381  0352    E235           SCR     A2,A5
00382  0354    1501           ADK     A5,1
00383  0356    1701           ADK     A7,1
00384  0358    5F80           RB(7)   INP2
00385  035A    4280   END1    CIO     A2,H,PTR
00386          035C   SSTPR1  EQU     *
00387  035C    4AC0           SST     A2,PTR
00388  035E    5C04           RB(4)   *-2
00389          0360   PRASCI  EQU     *
00390  0360    018A           LDK     A1,BUFF-HCIPL
00391  0362    910E           ADR     A1,A11
00392  0364    821C           LDR     A2,A7
00393  0366    0320           LDK     A3,/20
```

```
00394  0368  E335              SCR       A3,A5
00395  036A  1501              ADK       A5,1
00396  036C  8320              LDK,L     A3,/0D0A
       036E  0D0A
00397  0370  8335              STR       A3,A5
00398  0372  1203              ADK       A2,3
00399  0374  3A61              SRL       A2,1
00400  0376  3A41              SLL       A2,1
00401  0378  8082              LDR       A8,A8                  IF ASR DO NOT OUTPUT ASCII
00402  037A  5608              RF(6)     PRASCA                 A SECOND TIME (ALREADY PRINTED WHEN READ
00403  037C  904F              IM        COREND+2-HCIPL,A11     BUT WAIT ABIT
       037E  0358
00404  0380  5C06              RB(4)     *-4
00405  0382  5702              RF        PRASCB
00406        0384    PRASCA    EQU       *
00407  0384  F697              CFR       A14,A13
00408        0386    PRASCB    EQU       *
00409  0386  028A              LDK       A2,BUFF-HCIPL
00410  0388  920E              ADR       A2,A11
00411  038A  8328              LDR*      A3,A2
00412  038C  EB20              CWK       A3,/3A45               * IE
       038E  3A45
00413  0390  5CD0              RB(4)     RAFL3
00414  0392  1202              ADK       A2,2
00415  0394  8328              LDR*      A3,A2
00416  0396  EB20              CWK       A3,/4F46               *OF
       0398  4F46
00417  039A  5CDA              RB(4)     RAFL3
00418                    *
00419                    *
00420                    *                                SET RETURN PARAMETERS
00421                    *
00422  039C  82CE              LD        A10,COREND-HCIPL,A11   START ADDRESS IN A10
       039E  0356
00423  03A0  84A0              LDK,L     A12,MNLD-HCIPL
       03A2  029C
00424  03A4  948E              ADR       A12,A11                MNLD ADDRESS  IN A12
00425  03A6  81CE              LD        A9,SAVBAS-HCIPL,A11    LOADING BASE  IN A9
       03A8  00DC
00426  03AA  80CE              LD        A8,BADDR-HCIPL,A11     ENDING ADDRESS IN A8
       03AC  0116
00427  03AE  874E              LD        A7,MASTFG-HCIPL,A11    MASTER FLAG IN A7
       03B0  00DA
00428                    *
00429                    *
00430  03B2  F03A              RTN       A14                    RETURN TO CALLING
00431                    *
00432  03B4  5FF4      RAFL2   RB        RAFL3                  * RELAY TOWARDS RAFL
00433  03B6  5750      RELAY   RF        PROL01+2
00434                    *
00435  03B8  1300      SWITCH  ADK       A3,0
00436  03BA  59CE              RB(1)     OBJINP
00437  03BC  227F              ANK       A2,/7F
00438  03BE  58E6              RB(0)     INP2
00439  03C0  EA20              CWK       A2,/7F
       03C2  007F
00440  03C4  58EC              RB(0)     INP2
00441  03C6  1300              ADK       A3,0
```

```
00442  03C8  5A84            RB(2)    ASCINP
00443  03CA  EA20            CWK      A2,/1F
       03CC  001F
00444  03CF  5116            RF(1)    ASCII
00445  03D0  EA20            CWK      A2,/14
       03D2  0014
00446  03D4  510C            RF(1)    OBJEC
00447  03D6  EA20            CWK      A2,8
       03D8  0008
00448  03DA  5206            RF(2)    OBJEC
00449  03DC  EA20            CWK      A2,/10
       03DE  0010
00450  03E0  5CCA            RB(4)    INP2R
00451  03E2  1301    OBJEC   ADK      A3,1
00452  03E4  5FCC            RB(7)    FIRST
00453  03E6  1B01    ASCII   SUK      A3,1
00454  03E8  5FA4            RB(7)    ASCINP
00455                        EJECT
00456                *
00457                *
00458                *  MAIN  LOADING  PART          ENTRY PARAMETERS :
00459                *
00460                *                                 A1 = MASTER FLAG
00461                *                                 A9 = BASE ADDRESS
00462                *
00463                *
00464                *
00465                *                 1- OUTPUT MSG REQUESTING  TAPE ON READER
00466                *                 2- PERFORMS A 'HALT' - THE USER HAS THEN THE
00467                *                    POSSIBILITY TO ALTER THE BASE ADDRESS (REG,A9)
00468                *                                AND  THE MASTER FLAG  (REG,A1)
00469                *                                                      (A1=0 MASTER)
00470                *                                                      (A1=1 USER)
00471                *                 3- LOADING PROCESS STARTS  WHEN USER DEPPRESS START
00472                *                    BUTTON
00473                *
00474                *
00475                *
00476                *
00477         03EA   RAFL1   EQU      *
00478  03EA   5F38           RB       RAFL2
00479                *
00480                *
00481                *                                 ERRONNOUS CLUSTER,PRINT ERR MESSAGE
00482         03EC   CLC01   EQU      *
00483  03EC   5FBA           RB       ECCLS
00484                *
00485                *
00486                *
00487                *
00488         03EF   MNLD    EQU      *
00489  03EE   8404           LDR      A4,A1                SAVE  MASTER FLAG
00490  03F0   0200           LDK      A2,0
00491  03F2   824F           ST       A2,COREND-HCIPL,A11  RAZ  START  ADDRESS
       03F4   0356
00492                        IFT      MODE=BOM
00504                        XIF
```

```
00505  03F6  81CF              ST       A9,SAVBAS-HCIPL,A11
       03F8  00DC
00506                  *
00507                  *
00508                  *
00509                  * PROCESS LOADING : THIS  MODULE READ A  CLUSTER
00510                  *          AND  BRANCH ACCORDING TO THE CLUSTER TYPE
00511                  *          ON EXIT    A1 = BUFF  ADDRESS +1
00512                  *                     A2 = WORD  COUNT
00513                  *                     A3 = TYPE
00514                  *          THE TYPE MUST BE    3,4,7   IF NOT THIS ;HALT
00515  03FA  82A0      PROLO    LDK,L    A10,STAD-HCIPL         END ADDRESS
       03FC  0080
00516  03FE  928E              ADR      A10,A11
00517        0000      ABA      EQU      0
00518  0400  81CF              ST       A9,BADDR-HCIPL,A11     BADDR =BASE ADDRESS
       0402  0116
00519  0404  20BF      PROGLD   INH
00520        0406      PROL01   EQU      *
00521  0406  5F1E              RB       RAFL1
00522  0408  8120              LDK,L    A1,BUFF-HCIPL
       040A  008A
00523  040C  910E              ADR      A1,A11
00524  040E  0401              LDK      A4,1
00525  0410  0300              LDK      A3,0
00526  0412  E324              LCR      A3,A1                  A3 = TYPE
00527  0414  1101              ADK      A1,1
00528  0416  0200              LDK      A2,0
00529  0418  E224              LCR      A2,A1                  A2 = WORD COUNT
00530  041A  1101              ADK      A1,1
00531  041C  EB20              CWK      A3,3
       041E  0003
00532  0420  500E              RF(0)    CLCODE                 BRANCH  ON CLUSTER CODE
00533  0422  EB20              CWK      A3,4
       0424  0004
00534  0426  503A              RF(0)    CLIMOD                 INTERNAL MODIFICATION
00535  0428  EB20              CWK      A3,7
       042A  0007
00536  042C  505E              RF(0)    CLEND                  END/START
00537  042E  5F2A              RB(7)    PROL01
00538                  ************************************
00539                  * CLUSTER CODE  TYPE 3
00540                  *          UPON ENTRY: A1=ADDRESS OF BUFF+1 (RBK
00541                  *                      A2=WORD COUNT
00542                  *                      A9=BADDRESS
00543                  *                      A10=ENDADDRESS
00544                  ********************************************
00545  0430  834E      CLCODE   LD       A3,BUFF+6-HCIPL,A11
       0432  0090
00546  0434  5C30              RB(4)    PROL01                 EMBK SET SKIP THE CLUSTER
00547  0436  834E      CLC01A   LD       A3,BUFF+4-HCIPL,A11
       0438  008E
00548  043A  A311              TM       A3,A4                  IS IT  RELOCATABLE SECTION
00549  043C  5004              RF(0)    CLC04
00550  043E  3301              XRK      A3,1
00551  0440  9306              ADR      A3,A9
00552  0442  8524      CLC04    LDR*     A5,A1                  A5=(RBK)
```

```
00553  0444  1106              ADK       A1,6                  A1= ADDRESS OF ST CODE WORD IN BUFF
00554  0446  1A03              SUK       A2,3                  A2= NUMBER OF CODE WORD
00555                    *                                  A3= STORAGE ADDRESS
00556                    *                                  A4= MASK FOR RBK
00557                    *                                  A6= CODE WORD
00558  0448  3CE1        CLC05   SRC       A4,1
00559  044A  8624              LDR*      A6,A1
00560  044C  EB0A              CWR       A3,A10                COMPARE LOAD ADDRESS WITH AD OF IPL
00561  044E  5864              RB(0)     CLC01
00562  0450  A511              TM        A5,A4
00563  0452  5002              RF(0)     CLC07
00564  0454  9606              ADR       A6,A9
00565  0456  862D        CLC07   STR       A6,A3                 STORE CODE W=RDS
00566  0458  1102              ADK       A1,2
00567  045A  1302              ADK       A3,2
00568  045C  1A01              SUK       A2,1
00569  045E  5C18              RB(4)     CLC05
00570  0460  5F68              RB(7)     PROL0
00571                    *
00572                    ********************************************************************
00573                    * INTERNAL MODIFICATION CLUSTER
00574                    ********************************************
00575  0462  0701        CLIMOD  LDK       A7,1                  A7= MASK FOR ADDRESS
00576  0464  8524              LDR*      A5,A1                 A5=(RBK)
00577  0466  1A01              SUK       A2,1
00578  0468  3CE1        CLIM1   SRC       A4,1
00579  046A  1102              ADK       A1,2
00580  046C  8324              LDR*      A3,A1                 A3=ADDRESS
00581  046E  A31D              TM        A3,A7                 IS IT RELOCATABLE
00582  0470  5008              RF(0)     CLIM2                 NO
00583  0472  3301              XRK       A3,1
00584  0474  9306              ADR       A3,A9                 YES AD BASE
00585  0476  EB0A              CWR       A3,A10                ADDRESS OK
00586  0478  588E              RB(0)     CLC01
00587  047A  1102        CLIM2   ADK       A1,2
00588  047C  8624              LDR*      A6,A1                 TAKE CODE WORD
00589  047E  A511              TM        A5,A4                 IS IT RELOCATABLE
00590  0480  5002              RF(0)     CLIM3
00591  0482  9606              ADR       A6,A9                 AD BASE
00592  0484  862D        CLIM3   STR       A6,A3                 STORE CODE WORD
00593  0486  1A02              SUK       A2,2
00594  0488  5C22              RB(4)     CLIM1
00595  048A  5F92              RB(7)     PROL0
00596                    ****************************************
00597                    *  CLUSTER  END/START
00598                    ****************************************
00599  048C  8324        CLEND   LDR*      A3,A1
00600  048E  500A              RF(0)     CLEN3A                FINISH WO START
00601  0490  A311              TM        A3,A4
00602  0492  5002              RF(0)     CLEN1
00603  0494  9306              ADR       A3,A9
00604                    *
00605                    *
00606        0496        CLEN1   EQU       *
00607  0496  834F              ST        A3,COREND-HCIPL,A11
       0498  0356
00608  049A  814E        CLEN3A  LD        A1,BUFF+6-HCIPL,A11   UPDATE BASE ADDRESS
       049C  0090
```

```
00609  049E  914F            AD,S    A1,BADDR-HCIPL,A11
       04A0  0116
00610  04A2  9184            ADR     A9,A1
00611  04A4  5FAC            RB      PROLO
00612                  *
00613                  *
00614  04A6  0000      SAVA15  DATA    0
00615                  *
00616                  ********************************************************************************
00617                  *
00618  04A8  0000      COREND  DATA    0
00619                  *
00620                  *
00621                  ********************************************************************************
00622  04AA  0000              DATA    0
00623  04AC  0000              DATA    0
00624  04AE  0000              DATA    0
00625  04B0  0000              DATA    0
00626  04B2  0000              DATA    0
00627  04B4  0000              DATA    0
00628  04B6  0000              DATA    0
00629  04B8  0000              DATA    0
00630  04BA  0000              DATA    0
00631  04BC  0000              DATA    0
00632  04BE  0000              DATA    0
00633  04C0  0000              DATA    0
00634  04C2  0000              DATA    0
00635  04C4  0000              DATA    0
00636  04C6  0000              DATA    0
00637  04C8  0000              DATA    0
00638  04CA  0000              DATA    0
00639  04CC  0000              DATA    0
00640  04CE  0000              DATA    0
00641  04D0  0000              DATA    0
00642  04D2  0000              DATA    0
00643  04D4  0000              DATA    0
00644  04D6  0000              DATA    0
00645  04D8  0000              DATA    0
00646  04DA  0000              DATA    0
00647  04DC  0000              DATA    0
00648  04DE  0000              DATA    0
00649  04E0  0000              DATA    0
00650  04E2  0000              DATA    0
00651  04E4  0000              DATA    0
00652  04E6  0000              DATA    0
00653  04E8  0000              DATA    0
00654  04EA  0000              DATA    0
00655  04EC  0000              DATA    0
00656  04EE  0000              DATA    0
00657  04F0  0000              DATA    0
00658  04F2  0000              DATA    0
00659  04F4  0000              DATA    0
00660  04F6  0000              DATA    0
00661  04F8  0000              DATA    0
00662  04FA  0000              DATA    0
00663  04FC  0000              DATA    0
00664  04FE  0000              DATA    0
00665  0500  0000              DATA    0
```

```
00666  0502  0000                    DATA    0
00667  0504  0000                    DATA    0
00668  0506  0000                    DATA    0
00669  0508  0000                    DATA    0
00670  050A  0000                    DATA    0
00671  050C  0000                    DATA    0
00672  050E  0000                    DATA    0
00673  0510  0000                    DATA    0
00674  0512  0000                    DATA    0
00675  0514  0000                    DATA    0
00676  0516  0000                    DATA    0
00677  0518  0000                    DATA    0
00678  051A  0000                    DATA    0
00679  051C  0000                    DATA    0
00680  051E  0000                    DATA    0
00681  0520  0000                    DATA    0
00682                                EJECT
00683        0522        IPL44       EQU     *
00684  0522  030C                    LDK     A3,NRREC
00685  0524  8420                    LDKL    A4,BIPL
       0526  0100  R
00686  0528  0785                    LDK     A7,/85
00687                    * BASIC WRITE
00688  052A  80A0                    LDKL    A8,DECH44
       052C  0582  R
00689        052E        PUNCH4      EQU     *
00690  052E  0550                    LDK     A5,80
00691  0530  8620                    LDKL    A6,BUF44
       0532  058E  R
00692        0534        PCH400      EQU     *
00693  0534  0100                    LDK     A1,0
00694  0536  0200                    LDK     A2,0
00695  0538  E130                    LCR     A1,A4
00696  053A  3964                    SRL     A1,4
00697  053C  E921                    CCK     A1,0
       053E  0000
00698  0540  5006                    RF(0)   PCH405
00699  0542  E921                    CCK     A1,/500
       0544  0500
00700  0546  5202                    RF(2)   PCH410                  LESS THAN 5
00701        0548        PCH405      EQU     *
00702  0548  2910                    ORK     A1,/10
00703        054A        PCH410      EQU     *
00704  054A  3948                    SLL     A1,8
00705  054C  E130                    LCR     A1,A4
00706  054E  A120                    ANKL    A1,/FF0F
       0550  FF0F
00707  0552  E921                    CCK     A1,0
       0554  0000
00708  0556  5006                    RF(0)   PCH415
00709  0558  E921                    CCK     A1,/500
       055A  0500
00710  055C  5202                    RF(2)   PCH420
00711        055E        PCH415      EQU     *
00712  055E  2910                    ORK     A1,/10
00713        0560        PCH420      EQU     *
00714  0560  8139                    STR     A1,A6
00715                    * STORE WORD
```

```
00716  0562  1602                 ADK       A6,2                  NEXT WORD IN BUFFER
00717  0564  1401                 ADK       A4,1                  NEXT CHZR, IN IPL
00718  0566  1D01                 SUK       A5,1
00719                      *
00720  0568  5936                 RB(1)     PCH400
00721  056A  FB20                 CWK       A3,1                  LAST RECORD ?
       056C  0001
00722  056E  5406                 RF(4)     PCH430                NO
00723  0570  0113                 LDK       A1,/13                XOFF
00724  0572  8141                 ST        A1,BUF44+158
       0574  062C  R
00725        0576       PCH430    EQU       *
00726  0576  2804                 LKM
00727  0578  0001                 DATA      1
00728  057A  1801                 SUK       A3,1
00729  057C  5950                 RB(1)     PUNCH4
00730                      *
00731  057E  2804                 LKM
00732  0580  0003                 DATA      3
00733                      *
00734        0582       DECB44    EQU       *
00735  0582  0003                 DATA      3
00736  0584  058E  R              DATA      BUF44
00737  0586  00A0                 DATA      160
00738  0588  0000                 DATA      0
00739  058A  0000                 DATA      0
00740  058C  0000                 DATA      0
00741  058E             BUF44     RES       80
00742                      *
00743                      *
00744                      *
00745                                END       BIPLPR
```

SYMBOL TABLE

ABA	0000	A	ASCII	03E6	R	ASCINP	0346	R	ASR	0010	A
BADDR	0268	R	BIPL	0100	R	BIPLPR	0000	R	BOM	0001	A
BUF44	058E	R	BUFF	01DC	R	CFZON	023A	R	CIOPTR	0294	R
CKOK	0256	R	CLCO1	03EC	R	CLCO1A	0436	R	CLCO4	0442	R
CLCO5	0448	R	CLCO7	0456	R	CLCODE	0430	R	CLEN1	0496	R
CLEN3A	049A	R	CLEND	048C	R	CLIM1	0468	R	CLIM2	047A	R
CLIM3	0484	R	CLIMOD	0462	R	CNTFLG	014C	R	COREND	04A8	R
DECB	001E	R	DECB44	0582	R	DECBUF	0020	R	ECCLS	0334	R
ECMSG	01D2	R	EIGHT	02FF	R	END1	035A	R	END2	0320	R
FIRST	031A	R	H	0000	A	HCIPL	0152	R	HCIPL1	0150	R
HCIPL2	018A	R	INP2-	02DA	R	INP2R	0318	R	INPA	028A	R
INR44	02F4	R	IPL	0104	R	IPL010	0116	R	IPL020	0122	R
IPL100	012C	R	IPL110	0130	R	IPL44	0522	R	IPL88	0000	R
LDFLG	0100	R	MASTFG	022C	R	MLX1	02C4	R	MNLD	03EE	R
MODE	0000	A	NBREC	000C	A	OBJEC	03F2	R	OBJIN1	0316	R
OBJINP	02EE	R	OFLMSG	01D6	R	OTR	0242	R	OUTMSG	023C	R
PCH400	0534	R	PCH405	0548	R	PCH410	054A	R	PCH415	055E	R
PCH420	0560	R	PCH430	0576	R	PRASCA	0384	R	PRASCB	0386	R
PRASC1	0360	R	PROGLD	0404	R	PROLO	03FA	R	PROL01	0406	R
PTR	0000	A	PUNCH	000C	R	PUNCH4	052E	R	RAFL	026A	R
RAFL1	03EA	R	RAFL2	03B4	R	RAFL3	02C2	R	RELAY	03B6	R
RER	02CE	R	S	0001	A	SA	0000	A	SAVA15	04A6	R
SAVBAS	022E	R	SSTMLX	02A0	R	SSTPR1	035C	R	SSTPR2	02DE	R
SSTPTR	0322	R	STAD	01D2	R	STOP	0342	R	SWITCH	0388	R
SYSMSG	01AA	R	TMTEST	02A4	R	WER1	028C	R	WER2	028E	R

IPL52H

```
00000                                IDENT   IPL52H
00001                                ENTRY   IPL88
00002                                ENTRY   IPL44
00003                          *
00004                          *                              BELIER IPL GENERATION
00005          000C            NBREC   EQU     12
00006                          *
00007          0000            BIPLPR  EQU     *
00008          0000            IPL88   EQU     *
00009  0000    010C                    LDK     A1,NBREC
00010  0002    8220                    LDK,L   A2,BIPL-80
       0004    0080  R
00011  0006    0785                    LDK     A7,/85                 BASIC WRITE
00012  0008    80A0                    LDKL    A8,DECB
       000A    001E  R
00013          000C            PUNCH   EQU     *
00014  000C    1250                    ADK     A2,80
00015  000E    8241                    ST      A2,DECBUF              SET BUFF ADDR
       0010    0020  R
00016  0012    2804                    LKM                            PUNCH ONE RECORD
00017  0014    0001                    DATA    1
00018  0016    1901                    SUK     A1,1                   COUNT DONE ?
00019  0018    590E                    RB(1)   PUNCH                 NO
00020  001A    2804                    LKM                            YES
00021  001C    0003                    DATA    3                      EXIT
00022                          *
00023          001E            DECB    EQU     *
00024  001E    0003                    DATA    3                      FILE CODE
00025  0020                    DECBUF  RES     1
00026  0022    0050                    DATA    80
00027  0024    0000                    DATA    0
00028  0026    0000                    DATA    0
00029  0028    0000                    DATA    0
00030                                  EJECT
00031                                  RORG    BIPLPR+/100
00032                          *
00033          0100            BIPL    EQU     *
00034                          *                              1ST RECORD = LOW CORE IPL
00035                          *                              LOW CORE IPL,1 RECORD,LOADED AT /80 AND
00036                          *                              STARTED AT /84
00037                          *
00038  0100    FFFF            LDFLG   DATA    /FFFF                  LDFLG = -1 IF JUST LOADED
00039  0102    0000                    DATA    0                      NOT USED
00040          0104            IPL     EQU     *
00041                          *                              ADDR OF IPL = BASE ADDR
00042  0104    82A0                    LDKL    A10,/84
       0106    0084
00043                          *                              CHECK IF JUST LOADED OR NOT
00044  0108    904B                    IM      LDFLG-IPL,A10
       010A    FFFC
00045  010C    511E                    RF(1)   IPL100                 NO,NOT THE 1ST TIME
```

```
00046                          *
00047                          *
00048                          *                                YES,COMPUTE HIGH CORE LIMIT
00049                          *
00050  010E  8720                      LDK.L    A7,/5555              PATTERN
       0110  5555
00051  0112  81A0                      LDKL     A9,/FC00              HIGH CORE =/400
       0114  FC00
00052        0116  IPL010    EQU      *
00053  0116  8727                      STR      A7,A9
00054  0118  EF26                      CWR*     A7,A9                 IF THE LOCATION (A9) EXISTS,(A7)=((A9))
00055  011A  5006                      RF(0)    IPL020                MATCH
00056  011C  99A0                      SUKL     A9,/2000              NO,NEXT LOWER BLOCK OF 4K
       011E  2000
00057  0120  5F0C                      RB       IPL010
00058        0122  IPL020    EQU      *
00059  0122  8386                      LDR      A11,A9                SAVE BASE ADDR OF HIGH CORE IPL
00060  0124  93A0                      ADKL     A11,2                 START ADD.
       0126  0002
00061  0128  8486                      LDR      A12,A9                SAVE LOAD ADD.
00062  012A  5704                      RF       IPL110
00063        012C  IPL100    EQU      *
00064  012C  94A0                      ADKL     A12,80
       012E  0050
00065                          * LOAD ADDRESS OF NEXT RECORD
00066        0130  IPL110    EQU      *
00067  0130  871E                      LDR      A7,A15                RESTORE CIO INST IN BOOT
00068  0132  273F                      ANK      A7,/3F
00069  0134  9720                      ADK.L    A7,/41C0              CIO INSTRUCTION
       0136  41C0
00070  0138  872D                      STR      A7,A3
00071                          *
00072  013A  0557                      LDK      A5,/57                CHANGE LOC. /5A OF BOOT IN ORDER TO
00073  013C  E541                      SC       A5,/5A                CANCEL LEADING CHARACTER FLAG
       013E  005A
00074                          *                                RE INITIALIZE A5,A6
00075  0140  0550                      LDK      A5,80                 # OF CHARACTERS
00076  0142  8612                      LDR      A6,A12                LOAD ADDR.
00077                          *                                CHECK IF COUNT DONE
00078  0144  904B                      IM       CNTFLG-IPL,A10
       0146  0048
00079  0148  890E                      ABR(1)   A11
00080  014A  0F42                      AB       /42                   START BOOT ABAIN
00081        014C  CNTFLG    EQU      *                              = NUMBER OF RECORDS OF HIGH CORE IPL
00082  014C  FFF5                      DATA     1-NBREC
00083  014E  0000                      DATA     0
00084  0150  0000                      DATA     0
00085  0152  0000                      DATA     0
00086  0154  0000                      DATA     0
00087  0156  0000                      DATA     0
00088  0158  0000                      DATA     0
00089  015A  0000                      DATA     0
00090  015C  0000                      DATA     0
00091  015E  0000                      DATA     0
00092  0160  0000                      DATA     0
00093  0162  0000                      DATA     0
00094  0164  0000                      DATA     0
00095  0166  0000                      DATA     0
```

```
00096  0168  0000              DATA    0
00097  016A  0000              DATA    0
00098  016C  0000              DATA    0
00099  016E  0000              DATA    0
00100  0170  0000              DATA    0
00101  0172  0000              DATA    0
00102  0174  0000              DATA    0
00103  0176  0000              DATA    0
00104  0178  0000              DATA    0
00105  017A  0000              DATA    0
00106  017C  0000              DATA    0
00107  017E  0000              DATA    0
00108  0180  0000              DATA    0
00109  0182  0000              DATA    0
00110  0184  0000              DATA    0
00111  0186  0000              DATA    0
00112  0188  0000              DATA    0
00113  018A  0000              DATA    0
00114  018C  0000              DATA    0
00115  018E  0000              DATA    0
00116  0190  0000              DATA    0
00117  0192  0000              DATA    0
00118  0194  0000              DATA    0
00119  0196  0000              DATA    0
00120  0198  0000              DATA    0
00121  019A  0000              DATA    0
00122  019C  0000              DATA    0
00123  019E  0000              DATA    0
00124  01A0  0000              DATA    0
00125  01A2  0000              DATA    0
00126  01A4  0000              DATA    0
00127  01A6  0000              DATA    0
00128  01A8  0000              DATA    0
00129  01AA  0000              DATA    0
00130  01AC  0000              DATA    0
00131  01AE  0000              DATA    0
00132  01B0  0000              DATA    0
00133  01B2  0000              DATA    0
00134  01B4  0000              DATA    0
00135  01B6  0000              DATA    0
00136  01B8  0000              DATA    0
00137  01BA  0000              DATA    0
00138  01BC  0000              DATA    0
00139  01BE  0000              DATA    0
00140  01C0  0000              DATA    0
00141  01C2  0000              DATA    0
00142  01C4  0000              DATA    0
00143  01C6  0000              DATA    0
00144  01C8  0000              DATA    0
00145  01CA  0000              DATA    0
00146  01CC  0000              DATA    0
00147  01CE  0000              DATA    0
00148  01D0  0000              DATA    0
00149  01D2  0000              DATA    0
00150  01D4  0000              DATA    0
00151  01D6  0000              DATA    0
00152  01D8  0000              DATA    0
00153  01DA  0000              DATA    0
```

```
00154  01DC  0000              DATA    0
00155  01DE  0000              DATA    0
00156  01E0  0000              DATA    0
00157  01E2  0000              DATA    0
00158  01E4  0000              DATA    0
00159  01E6  0000              DATA    0
00160  01E8  0000              DATA    0
00161  01EA  0000              DATA    0
00162  01EC  0000              DATA    0
00163  01EE  0000              DATA    0
00164  01F0  0000              DATA    0
00165  01F2  0000              DATA    0
00166  01F4  0000              DATA    0
00167  01F6  0000              DATA    0
00168  01F8  0000              DATA    0
00169  01FA  0000              DATA    0
00170  01FC  0000              DATA    0
00171  01FE  0000              DATA    0
00172  0200  0000              DATA    0
00173  0202  0000              DATA    0
00174  0204  0000              DATA    0
00175                          EJECT
00176                                                  HIGH CORE IPL
00177        0000      PTR     EQU     0               DEVICE ADDR WILL BE INITIALIZED BY HCIPL
00178        0010      ASR     EQU     /10
00179        0001      S       EQU     1
00180        0000      H       EQU     0
00181        0001      BOM     EQU     1
00182        0000      SA      EQU     0
00183                  *
00184        0001      MODE    EQU     BOM             FOR MESSAGE  OUTPUT
00185                          RORG    BIPL+80         1ST RECORD OF HIGH CORE IPL
00186        0150      HCIPL1  EQU     *               1ST WORD OF THE CURRENT RECORD
00187  0150  FEFE              DATA    /FEFF
00188        0152      HCIPL   EQU     *
00189  0152  85A0              LDKL    A13,OUTMSG-HCIPL
       0154  00EA
00190  0156  958E              ADR     A13,A11         LOAD A13 WITH ADDR OF OUTMSG ROUTINE
00191  0158  868E              LDR     A14,A11
00192  015A  96A0              ADKL    A14,CFZON-HCIPL
       015C  00E8
00193                  *               ADDR OF STACK
00194  015E  871E              LDR     A7,A15          GET DEVICE ADDR
00195  0160  87CF              ST      A15,SAVA15-HCIPL,A11  SAVE  4*4  FLAG
       0162  0366
00196  0164  5606              RF(6)   *+8
00197  0166  904F              IM      COREND+2-HCIPL,A11
       0168  036A
00198  016A  5C06              RB(4)   *-4
00199  016C  273F              ANK     A7,/3F
00200                  *                       INIT IO INSTRUCTIONS
00201  016E  974F              ADS     A7,CIOPTR-HCIPL,A11
       0170  0142
00202  0172  974F              ADS     A7,INP2-HCIPL,A11
       0174  0188
00203  0176  974F              ADS     A7,END2-HCIPL,A11
       0178  01CE
```

```
00204  017A  974F            ADS     A7,END1-HCIPL,A11
       017C  0208
00205  017E  974F            ADS     A7,INR44-HCIPL,A11
       0180  01A2
00206  0182  811E            LDR     A1,A15
00207  0184  39C8            SLC     A1,8                    TEST MULTI-SINGLE DEVICE CONT,
00208  0186  5602            RF(6)   HCIPL2
00209  0188  270F            ANK     A7,/F                   MULTI DEVICE CONTROLLER
00210        018A  HCIPL2     EQU     *
00211  018A  974F            AD,S    A7,SSTMLX-HCIPL,A11
       018C  014E
00212  018E  974F            ADS     A7,SSTPR2-HCIPL,A11
       0190  018C
00213  0192  974F            ADS     A7,SSTPTR-HCIPL,A11
       0194  01D0
00214  0196  974F            ADS     A7,SSTPR1-HCIPL,A11
       0198  020A
00215  019A  3F41            SLL     A7,1
00216  019C  974F            AD,S    A7,WER1-HCIPL,A11       INIT WER/RER INST
       019E  013A
00217  01A0  974F            AD,S    A7,WER2-HCIPL,A11
       01A2  013C
00218  01A4  974F            AD,S    A7,RER-HCIPL,A11
       01A6  017C
00219  01A8  57AC            RF      CKOK
00220  01AA  4F42  SYSMSG    DATA    'OBJCT TAPE ON RF'
       01AC  4A43
       01AE  5420
       01B0  5441
       01B2  5045
       01B4  204F
       01B6  4E20
       01B8  5245
00221  01BA  4144            DATA    'ADER, THINK OF B'
       01BC  4552
       01BE  2E20
       01C0  5448
       01C2  494E
       01C4  4B20
       01C6  4F46
       01C8  2042
00222  01CA  4153            DATA    'ASE 1'
       01CC  4520
       01CE  2120
00223  01D0  0D0A            DATA    /0D0A
00224        01D2  STAD       EQU     *
00225  01D2  4543  ECMSG     DATA    'EC'
00226  01D4  0D0A            DATA    X'0D0A'
00227  01D6  4F56  OFLMSG    DATA    'OVFR'
       01D8  4652
00228  01DA  0D0A            DATA    /0D0A
00229  01DC        BUFF      RES     39
00230  022A  FFFF            DATA    /FFFF
00231  022C  0000  MASTFG    DATA    0
00232  022E  0000  SAVBAS    DATA    0
00233  0230                  RES     6                       * STACK  AREA
00234        023A  CFZON      EQU     *-2
00235        023C  OUTMSG     EQU     *
```

```
00236  023C  0304            LDK       A3,4
00237  023E  43D0            CIO       A3,8,ASR
00238  0240  E324            LCR       A3,A1
00239  0242  4310    OTR     OTR       A3,0,ASR
00240  0244  5C04            RB(4)     *-2
00241  0246  1101            ADK       A1,1
00242  0248  1A01            SUK       A2,1
00243  024A  5C0C            RB(4)     OTR-2
00244  024C  0300            LDK       A3,0
00245  024E  4390            CIO       A3,H,ASR
00246  0250  4BD0            SST       A3,ASR
00247  0252  5C04            RB(4)     *-2
00248  0254  F03A            RTN       A14
00249                 *
00250                 *
00251        0256    CKOK    EQU       *
00252  0256  81A0            LDK,L     A9,0                    SET LOADING ADDRESS
       0258  0000
00253                 *
00254                 *
00255                 *                                 GO LOAD SYSTEM
00256  025A  84A0            LDK,L     A12,MNLD-HCIPL
       025C  029C
00257  025E  948E            ADR       A12,A11
00258  0260  F693            CFR       A14,A12
00259                 *
00260                 *
00261  0262  810A            LDR       A1,A10                  TEST START ADDRESS
00262  0264  8C04            ABR(4)    A1                      EOS  OR EOF HAS BEEN READ
00263  0266  207F            HLT                               NO START ADDRESS
00264  0268  0000    BADDR   DATA      0
00265  026A  0300    RAFL    LDK       A3,0
00266  026C  0700            LDK       A7,0
00267  026E  8520            LDK,L     A5,BUFF-HCIPL
       0270  008A
00268  0272  950E            ADR       A5,A11
00269  0274  80CE            LD        A8,SAVA15-HCIPL,A11
       0276  0366
00270  0278  5610            RF(6)     INPA
00271  027A  0604            LDK       A6,4                    4*4 SO ASR
00272  027C  0111            LDK       A1,/11                  SEND X-ON
00273  027E  46D0            CIO       A6,S,ASR
00274  0280  4110            OTR       A1,0,ASR
00275  0282  5C04            RB(4)     *-2
00276  0284  4290            CIO       A2,H,ASR
00277  0286  4AD0            SST       A2,ASR
00278  0288  5C04            RB(4)     *-2
00279        028A    INPA    EQU       *
00280  028A  0650            LDK       A6,80                   LENGTH
00281  028C  7600    WER1    WER       A6,0                    INIT MLX WHATEVER
00282  028E  7501    WER2    WER       A5,1                    CHANNEL IS
00283  0290  8602            LDR       A6,A8
00284  0292  3E68            SRL       A6,8
00285        0294    CIOPTR  EQU       *
00286  0294  46C0            CIO       A6,S,PTR                AVAILABLE ON THE PAPER READER
00287  0296  5C04            RB(4)     *-2
00288  0298  824E            LD        A2,SAVA15-HCIPL,A11
       029A  0366
```

```
00289  029C  3AC3          SLC     A2,3            MLX ?
00290  029E  523A          RF(2)   INP2            NO
00291        02A0    SSTMLX  EQU     *
00292  02A0  49C0          SST     A1,PTR
00293  02A2  5C04          RB(4)   *-2
00294        02A4    TMTEST  EQU     *
00295  02A4  A120          ANKL    A1,/1007        TAPE MARK OR FATAL ERROR ?
       02A6  1007
00296  02A8  501A          RF(0)   MLX1            NO
00297  02AA  2107          ANK     A1,/7           FATAL ERROR ?
00298  02AC  5486          RF(4)   ECCLS           YES
00299                      *
00300  02AE  8120          LDK,L   A1,'1E'         PUT EOF IN BUFF
       02B0  3A45
00301  02B2  8135          STR     A1,A5
00302  02B4  8120          LDK,L   A1,'0F'
       02B6  4F46
00303  02B8  8155          ST      A1,2,A5
       02BA  0002
00304  02BC  0704          LDK     A7,4            UPDATA POINTRERS
00305  02BE  951C          ADR     A5,A7
00306  02C0  579E          RF      PRASCI
00307                      *
00308  02C2  5F5A    RAFL3 RB      RAFL            RELAY TOWARDS RAFL
00309                      *
00310        02C4    MLX1    EQU     *
00311  02C4  E134          LCR     A1,A5
00312  02C6  E920          CWK     A1,/18          IS IT ASCII
       02C8  0018
00313  02CA  55EA          RF(5)   RELAY           NO
00314  02CC  0750          LDK     A7,80           YES
00315  02CE  7E00    RER   RER     A6,0            READ REMAINING LENGTH
00316  02D0  A620          ANK,L   A6,/FFF
       02D2  0FFF
00317  02D4  9F18          SUR     A7,A6           COMPUTE TRANSMITTED LGT
00318  02D6  951C          ADR     A5,A7           UPDATE BUFF POINTER
00319  02D8  5786          RF      PRASCI
00320  02DA  4A00    INP2  INR     A2,0,PTR
00321  02DC  50DA          RF(0)   SWITCH          INR HAS BEEN ACCEPTED ;
00322                      *                       PROCESS THE CHARACTER
00323  02DE  4AC0    SSTPR2 SST    A2,PTR          TRY A SST
00324  02E0  5C08          RB(4)   INP2            TRY AGAIN INR WHEN SST REFUSED
00325  02E2  8108          LDR     A1,A2           SAVE STATUS IN A1
00326  02E4  1300          ADK     A3,0            CHECK WETHER ASCII OR OBJECT
00327  02E6  5278          RF(2)   PRASCI          RECORD WAS ASCII ; PRINT IT
00328  02E8  0600          LDK     A6,0            STORE A NON ASCII CHARACTER AT
00329  02EA  8645          STR     A6,A5           THE END OF BUFFER ;
00330                      *                       TO BE USED IN MX1
00331  02EC  5F4A          RB      TMTEST
00332        02EE    OBJINP  EQU     *
00333  02EE  8082          LDR     A8,A8
00334  02F0  560C          RF(6)   EIGHT           8=8 DO NOT INPUT 2ND CHARACTER
00335  02F2  220F          ANK     A2,/F
00336  02F4  4E00    INR44 INR     A6,0,PTR
00337  02F6  5C04          RB(4)   *-2
00338  02F8  260F          ANK     A6,/F
00339  02FA  3A44          SLL     A2,4
00340  02FC  9218          ADR     A2,A6           JOIN THE TWO HALF CHARACTERS
```

```
00341          02FE     EIGHT   EQU     *
00342  02FE    EF04             CWR     A7,A1
00343  0300    501E             RF(0)   END2
00344  0302    F235             SCR     A2,A5
00345  0304    8408             XRR     A4,A2
00346  0306    1501             ADK     A5,1
00347  0308    EF20             CWK     A7,1
       030A    0001
00348  030C    5408             RF(4)   OBJIN1
00349  030E    0400             LDK     A4,0
00350  0310    8108             LDR     A1,A2
00351  0312    9108             ADR     A1,A2
00352  0314    1103             ADK     A1,3
00353  0316    1701     OBJIN1  ADK     A7,1
00354  0318    5F40     INP2R   RB      INP2
00355  031A    2207     FIRST   ANK     A2,/7
00356  031C    0150             LDK     A1,80
00357  031E    5F22             RB(7)   EIGHT
00358  0320    4280     END2    CIO     A2,H,PTR
00359          0322     SSTPTR  EQU     *
00360  0322    49C0             SST     A1,PTR
00361  0324    5C04             RB(4)   *-2
00362  0326    8082             LDR     A8,A8                   IF ASR WAIT
00363  0328    5606             RF(6)   *+8
00364  032A    904F             IM      COREND+2-HCIPL,A11      WAIT
       032C    036A
00365  032E    5C06             RB(4)   *-4
00366  0330    24FF             ANK     A4,/FF
00367  0332    50E6             RF(0)   PROLO1+2
00368          0334     ECCLS   EQU     *
00369  0334    0180             LDK     A1,ECMSG-HCIPL
00370  0336    910E             ADR     A1,A11
00371  0338    0204             LDK     A2,4
00372  033A    84A0             LDK,L   A12,OUTMSG-HCIPL
       033C    00EA
00373  033E    948E             ADR     A12,A11
00374  0340    F693             CFR     A14,A12
00375  0342    207F     STOP    HLT
00376  0344    5FDC             RB(7)   RAFL
00377  0346    EA20     ASCINP  CWK     A2,/0D
       0348    000D
00378  034A    500E             RF(0)   END1
00379  034C    EF20             CWK     A7,68
       034E    0044
00380  0350    5878             RB(0)   INP2
00381  0352    E235             SCR     A2,A5
00382  0354    1501             ADK     A5,1
00383  0356    1701             ADK     A7,1
00384  0358    5F80             RB(7)   INP2
00385  035A    4280     END1    CIO     A2,H,PTR
00386          035C     SSTPR1  EQU     *
00387  035C    4AC0             SST     A2,PTR
00388  035E    5C04             RB(4)   *-2
00389          0360     PRASCI  EQU     *
00390  0360    018A             LDK     A1,BUFF-HCIPL
00391  0362    910E             ADR     A1,A11
00392  0364    821C             LDR     A2,A7
00393  0366    0320             LDK     A3,/20
```

```
00394  0368  E335          SCR     A3,A5
00395  036A  1501          ADK     A5,1
00396  036C  8320          LDK,L   A3,/0D0A
       036E  0D0A
00397  0370  8335          STR     A3,A5
00398  0372  1203          ADK     A2,3
00399  0374  3A61          SRL     A2,1
00400  0376  3A41          SLL     A2,1
00401  0378  8082          LDR     A8,A8                 IF ASR DO NOT OUTPUT ASCII
00402  037A  5608          RF(6)   PRASCA                A SECOND TIME (ALREADY PRINTED WHEN READ
00403  037C  904F          IM      COREND+2-HCIPL,A11    BUT WAIT ABIT
       037E  036A
00404  0380  5C06          RB(4)   *-4
00405  0382  5702          RF      PRASCB
00406        0384    PRASCA EQU    *
00407  0384  F697          CFR     A14,A13
00408        0386    PRASCB EQU    *
00409  0386  028A          LDK     A2,BUFF-HCIPL
00410  0388  920E          ADR     A2,A11
00411  038A  8328          LDR*    A3,A2
00412  038C  EB20          CWK     A3,/3A45              * :E
       038E  3A45
00413  0390  5CD0          RB(4)   RAFL3
00414  0392  1202          ADK     A2,2
00415  0394  8328          LDR*    A3,A2
00416  0396  EB20          CWK     A3,/4F46              *OF
       0398  4F46
00417  039A  5CDA          RB(4)   RAFL3
00418              *
00419              *
00420              *                          SET RETURN PARAMETERS
00421              *
00422  039C  82CE          LD      A10,COREND-HCIPL,A11  START ADDRESS IN A10
       039E  0368
00423  03A0  84A0          LDK,L   A12,MNLD-HCIPL
       03A2  029C
00424  03A4  948E          ADR     A12,A11               MNLD ADDRESS  IN A12
00425  03A6  81CE          LD      A9,SAVBAS-HCIPL,A11   LOADING BASE  IN A9
       03A8  00DC
00426  03AA  80CE          LD      A8,BADDR-HCIPL,A11    ENDING ADDRESS IN A8
       03AC  0116
00427  03AE  874E          LD      A7,MASTFG-HCIPL,A11   MASTER FLAG IN A7
       03B0  00DA
00428              *
00429              *
00430  03B2  F03A          RTN     A14                   RETURN TO CALLING
00431              *
00432  03B4  5FF4    RAFL2 RB      RAFL3                 * RELAY TOWARDS RAFL
00433  03B6  5762    RELAY RF      PROLO1+2
00434              *
00435  03B8  1300    SWITCH ADK    A3,0
00436  03BA  59CE          RB(1)   OBJINP
00437  03BC  227F          ANK     A2,/7F
00438  03BE  58E6          RB(0)   INP2
00439  03C0  EA20          CWK     A2,/7F
       03C2  007F
00440  03C4  58EC          RB(0)   INP2
00441  03C6  1300          ADK     A3,0
```

```
00442  03C8  5A84            RB(2)    ASCINP
00443  03CA  EA20            CWK      A2,/1F
       03CC  001F
00444  03CE  5116            RF(1)    ASCII
00445  03D0  EA20            CWK      A2,/14
       03D2  0014
00446  03D4  510C            RF(1)    OBJEC
00447  03D6  EA20            CWK      A2,8
       03D8  0008
00448  03DA  5206            RF(2)    OBJEC
00449  03DC  EA20            CWK      A2,/10
       03DE  0010
00450  03E0  5CCA            RB(4)    INP2R
00451  03E2  1301     OBJEC  ADK      A3,1
00452  03E4  5FCC            RB(7)    FIRST
00453  03E6  1B01     ASCII  SUK      A3,1
00454  03E8  5FA4            RB(7)    ASCINP
00455                        EJECT
00456                 *
00457                 *
00458                 *  MAIN  LOADING  PART          ENTRY PARAMETERS :
00459                 *
00460                 *                                 A1 = MASTER FLAG
00461                 *                                 A9 = BASE ADDRESS
00462                 *
00463                 *
00464                 *
00465                 *                 1- OUTPUT MSG REQUESTING  TAPE ON READER
00466                 *                 2- PERFORMS A 'HALT' - THE USER HAS THEN THE
00467                 *                    POSSIBILITY TO ALTER THE BASE ADDRESS (REG,A9)
00468                 *                                AND  THE MASTER FLAG  (REG,A1)
00469                 *                                                      (A1=0 MASTER)
00470                 *                                                      (A1=1 USER)
00471                 *                 3- LOADING PROCESS STARTS  WHEN USER DEPPRESS START
00472                 *                    BUTTON
00473                 *
00474                 *
00475                 *
00476                 *
00477         03EA    RAFL1    EQU      *
00478  03EA   5F38            RB       RAFL2
00479                 *
00480                 *
00481                 *                                ERRONNOUS CLUSTER,PRINT ERR MESSAGE
00482         03EC    CLC01    EQU      *
00483  03EC   5FBA            RB       FCCLS
00484                 *
00485                 *
00486                 *
00487                 *
00488         03EE    MNLD     EQU      *
00489  03EE   8404            LDR      A4,A1                  SAVE  MASTER FLAG
00490  03F0   0200            LDK      A2,0
00491  03F2   824F            ST       A2,COREND-HCIPL,A11    RAZ  START  ADDRESS
       03F4   0368
00492                         IFT      MODE=BOM
00493  03F6   8120            LDK,L    A1,SYSMSG-HCIPL
       03F8   0058
```

```
00494  03FA  910E              ADR       A1,A11
00495  03FC  0228              LDK       A2,/28
00496  03FE  F697              CFR       A14,A13                SEND  MESSAGE
00497                    *
00498                    *
00499  0400  8110              LDR       A1,A4                  RESTORE MASTER FLAG
00500                    *
00501  0402  207F              HLT                              HALT FOR EVENTUAL REGISTERS MODIF
00502                    *
00503  0404  814F              ST        A1,MASTFG-HCIPL,A11
       0406  00DA
00504                          XIF
00505  0408  81CF              ST        A9,SAVBAS-HCIPL,A11
       040A  00DC
00506                    *
00507                    *
00508                    *
00509                    * PROCESS LOADING : THIS  MODULE READ A  CLUSTER
00510                    *          AND  BRANCH ACCORDING TO THE CLUSTER TYPE
00511                    *          ON EXIT   A1 = BUFF  ADDRESS +1
00512                    *                    A2 = WORD  COUNT
00513                    *                    A3 = TYPE
00514                    *          THE TYPE MUST BE    3,4,7    IF NOT THIS :HALT
00515  040C  82A0        PROLO     LDK,L     A10,STAD-HCIPL         END ADDRESS
       040E  0080
00516  0410  928E              ADR       A10,A11
00517        0000        ABA       EQU       0
00518  0412  81CF              ST        A9,BADDR-HCIPL,A11     BADDR =BASE ADDRESS
       0414  0116
00519  0416  20BF        PROGLD    INH
00520        0418        PROLO1    EQU       *
00521  0418  5F30              RB        RAFL1
00522  041A  8120              LDK,L     A1,BUFF-HCIPL
       041C  008A
00523  041E  910E              ADR       A1,A11
00524  0420  0401              LDK       A4,1
00525  0422  0300              LDK       A3,0
00526  0424  E324              LCR       A3,A1                  A3 = TYPE
00527  0426  1101              ADK       A1,1
00528  0428  0200              LDK       A2,0
00529  042A  E224              LCR       A2,A1                  A2 = WORD COUNT
00530  042C  1101              ADK       A1,1
00531  042E  EB20              CWK       A3,3
       0430  0003
00532  0432  500E              RF(0)     CLCODE                 BRANCH  ON CLUSTER CODE
00533  0434  EB20              CWK       A3,4
       0436  0004
00534  0438  503A              RF(0)     CLIMOD                 INTERNAL MODIFICATION
00535  043A  EB20              CWK       A3,7
       043C  0007
00536  043E  505E              RF(0)     CLEND                  END/START
00537  0440  5F2A              RB(7)     PROLO1
00538                    ************************************
00539                    * CLUSTER CODE  TYPE 3
00540                    *         UPON ENTRY: A1=ADDRESS OF BUFF+1 (RBK
00541                    *                     A2=WORD COUNT
00542                    *                     A9=BADDRESS
00543                    *                     A10=ENDADDRESS
00544                    ********************************************
```

```
00545  0442  834E      CLCODE   LD       A3,BUFF+6-HCIPL,A11
       0444  0090
00546  0446  5C30               RB(4)    PROLO1                    EMBK SET SKIP THE CLUSTER
00547  0448  834E      CLCO1A   LD       A3,BUFF+4-HCIPL,A11
       044A  008E
00548  044C  A311               TM       A3,A4                     IS IT  RELOCATABLE SECTION
00549  044E  5004               RF(0)    CLCO4
00550  0450  3301               XRK      A3,1
00551  0452  9306               ADR      A3,A9
00552  0454  8524      CLCO4    LDR*     A5,A1                     A5=(RBK)
00553  0456  1106               ADK      A1,6                      A1= ADDRESS OF ST CODE WORD IN BUFF
00554  0458  1A03               SUK      A2,3                      A2= NUMBER OF CODE WORD
00555                    *                                  A3= STORAGE ADDRESS
00556                    *                                  A4= MASK FOR RBK
00557                    *                                  A6= CODE WORD
00558  045A  3CE1      CLCO5    SRC      A4,1
00559  045C  8624               LDR*     A6,A1
00560  045E  EB0A               CWR      A3,A10                    COMPARE LOAD ADDRESS WITH AD OF IPL
00561  0460  5876               RB(0)    CLCO1
00562  0462  A511               TM       A5,A4
00563  0464  5002               RF(0)    CLCO7
00564  0466  9606               ADR      A6,A9
00565  0468  862D      CLCO7    STR      A6,A3                     STORE CODE W-RDS
00566  046A  1102               ADK      A1,2
00567  046C  1302               ADK      A3,2
00568  046E  1A01               SUK      A2,1
00569  0470  5C18               RB(4)    CLCO5
00570  0472  5F68               RB(7)    PROLO
00571                    *
00572                    ****************************************************************
00573                    * INTERNAL MODIFICATION CLUSTER
00574                    ************************************************
00575  0474  0701      CLIMOD   LDK      A7,1                      A7= MASK FOR ADDRESS
00576  0476  8524               LDR*     A5,A1                     A5=(RBK)
00577  0478  1A01               SUK      A2,1
00578  047A  3CE1      CLIM1    SRC      A4,1
00579  047C  1102               ADK      A1,2
00580  047E  8324               LDR*     A3,A1                     A3=ADDRESS
00581  0480  A31D               TM       A3,A7                     IS IT RELOCATABLE
00582  0482  5008               RF(0)    CLIM2                     NO
00583  0484  3301               XRK      A3,1
00584  0486  9306               ADR      A3,A9                     YES AD BASE
00585  0488  EB0A               CWR      A3,A10                    ADDRESS OK
00586  048A  58A0               RB(0)    CLCO1
00587  048C  1102      CLIM2    ADK      A1,2
00588  048E  8624               LDR*     A6,A1                     TAKE CODE WORD
00589  0490  A511               TM       A5,A4                     IS IT RELOCATABLE
00590  0492  5002               RF(0)    CLIM3
00591  0494  9606               ADR      A6,A9                     AD BASE
00592  0496  862D      CLIM3    STR      A6,A3                     STORE CODE WORD
00593  0498  1A02               SUK      A2,2
00594  049A  5C22               RB(4)    CLIM1
00595  049C  5F92               RB(7)    PROLO
00596                    ****************************************
00597                    *  CLUSTER  END/START
00598                    ****************************************
00599  049E  8324      CLEND    LDR*     A3,A1
```

```
00600  04A0  500A              RF(0)    CLEN3A                FINISH WO START
00601  04A2  A311              TM       A3,A4
00602  04A4  5002              RF(0)    CLEN1
00603  04A6  9306              ADR      A3,A9
00604                    *
00605                    *
00606        04A8      CLEN1   EQU      *
00607  04A8  834F              ST       A3,COREND-HCIPL,A11
       04AA  0368
00608  04AC  814E      CLEN3A  LD       A1,BUFF+6-HCIPL,A11   UPDATE BASE ADDRESS
       04AE  0090
00609  04B0  914F              AD,S     A1,BADDR-HCIPL,A11
       04B2  0116
00610  04B4  9184              ADR      A9,A1
00611  04B6  5FAC              RB       PROLO
00612                    *
00613                    *
00614  04B8  0000      SAVA15  DATA     0
00615                    *
00616                    ********************************************************************************
00617                    *
00618  04BA  0000      COREND  DATA     0
00619                    *
00620                    *
00621                    ********************************************************************************
00622  04BC  0000              DATA     0
00623  04BE  0000              DATA     0
00624  04C0  0000              DATA     0
00625  04C2  0000              DATA     0
00626  04C4  0000              DATA     0
00627  04C6  0000              DATA     0
00628  04C8  0000              DATA     0
00629  04CA  0000              DATA     0
00630  04CC  0000              DATA     0
00631  04CE  0000              DATA     0
00632  04D0  0000              DATA     0
00633  04D2  0000              DATA     0
00634  04D4  0000              DATA     0
00635  04D6  0000              DATA     0
00636  04D8  0000              DATA     0
00637  04DA  0000              DATA     0
00638  04DC  0000              DATA     0
00639  04DE  0000              DATA     0
00640  04E0  0000              DATA     0
00641  04E2  0000              DATA     0
00642  04E4  0000              DATA     0
00643  04F6  0000              DATA     0
00644  04E8  0000              DATA     0
00645  04EA  0000              DATA     0
00646  04EC  0000              DATA     0
00647  04EE  0000              DATA     0
00648  04F0  0000              DATA     0
00649  04F2  0000              DATA     0
00650  04F4  0000              DATA     0
00651  04F6  0000              DATA     0
00652  04F8  0000              DATA     0
00653  04FA  0000              DATA     0
```

```
00654  04FC  0000                DATA    0
00655  04FE  0000                DATA    0
00656  0500  0000                DATA    0
00657  0502  0000                DATA    0
00658  0504  0000                DATA    0
00659  0506  0000                DATA    0
00660  0508  0000                DATA    0
00661  050A  0000                DATA    0
00662  050C  0000                DATA    0
00663  050E  0000                DATA    0
00664  0510  0000                DATA    0
00665  0512  0000                DATA    0
00666  0514  0000                DATA    0
00667  0516  0000                DATA    0
00668  0518  0000                DATA    0
00669  051A  0000                DATA    0
00670  051C  0000                DATA    0
00671  051E  0000                DATA    0
00672  0520  0000                DATA    0
00673  0522  0000                DATA    0
00674  0524  0000                DATA    0
00675  0526  0000                DATA    0
00676  0528  0000                DATA    0
00677  052A  0000                DATA    0
00678  052C  0000                DATA    0
00679  052E  0000                DATA    0
00680  0530  0000                DATA    0
00681  0532  0000                DATA    0
00682                            EJECT
00683        0534      IPL44     EQU     *
00684  0534  030C                LDK     A3,NBREC
00685  0536  8420                LDKL    A4,BIPL
       0538  0100  R
00686  053A  0785                LDK     A7,/85
00687                      * BASIC WRITE
00688  053C  80A0                LDKL    A8,DECB44
       053E  0594  R
00689        0540      PUNCH4    EQU     *
00690  0540  0550                LDK     A5,80
00691  0542  8620                LDKL    A6,BUF44
       0544  05A0  R
00692        0546      PCH400    EQU     *
00693  0546  0100                LDK     A1,0
00694  0548  0200                LDK     A2,0
00695  054A  E130                LCR     A1,A4
00696  054C  3964                SRL     A1,4
00697  054E  E921                CCK     A1,0
       0550  0000
00698  0552  5006                RF(0)   PCH405
00699  0554  E921                CCK     A1,/500
       0556  0500
00700  0558  5202                RF(2)   PCH410                  LESS THAN 5
00701        055A      PCH405    EQU     *
00702  055A  2910                ORK     A1,/10
00703        055C      PCH410    EQU     *
00704  055C  3948                SLL     A1,8
00705  055E  E130                LCR     A1,A4
```

```
00706  0560  A120              ANKL    A1,/FF0F
       0562  FF0F
00707  0564  E921              CCK     A1,0
       0566  0000
00708  0568  5006              RF(0)   PCH415
00709  056A  E921              CCK     A1,/500
       056C  0500
00710  056E  5202              RF(2)   PCH420
00711        0570        PCH415 EQU     *
00712  0570  2910              ORK     A1,/10
00713        0572        PCH420 EQU     *
00714  0572  8139              STR     A1,A6
00715                    * STORE WORD
00716  0574  1602              ADK     A6,2                NEXT WORD IN BUFFER
00717  0576  1401              ADK     A4,1                NEXT CHZR, IN IPL
00718  0578  1D01              SUK     A5,1
00719                    *
00720  057A  5936              RB(1)   PCH400
00721  057C  EB20              CWK     A3,1                LAST RECORD ?
       057E  0001
00722  0580  5406              RF(4)   PCH430              NO
00723  0582  0113              LDK     A1,/13              XOFF
00724  0584  8141              ST      A1,BUF44+158
       0586  063E  R
00725        0588        PCH430 EQU     *
00726  0588  2804              LKM
00727  058A  0001              DATA    1
00728  058C  1B01              SUK     A3,1
00729  058E  5950              RB(1)   PUNCH4
00730                    *
00731  0590  2804              LKM
00732  0592  0003              DATA    3
00733                    *
00734        0594        DECB44 EQU     *
00735  0594  0003              DATA    3
00736  0596  05A0  R           DATA    BUF44
00737  0598  00A0              DATA    160
00738  059A  0000              DATA    0
00739  059C  0000              DATA    0
00740  059E  0000              DATA    0
00741  05A0              BUF44  RES     80
00742                    *
00743                    *
00744                    *
00745                    END     BIPLPR
```

```
IPL52H
 SYMBOL TABLE

ABA     0000  A   ASCII   03E6  R   ASCINP  0346  R   ASR     0010  A
BADDR   0268  R   BIPL    0100  R   BIPLPR  0000  R   BOM     0001  A
BUF44   05A0  R   BUFF    01DC  R   CFZON   023A  R   CIOPTR  0294  R
CKOK    0256  R   CLCO1   03EC  R   CLCO1A  0448  R   CLCO4   0454  R
CLCO5   045A  R   CLCO7   0468  R   CLCODE  0442  R   CLEN1   04A8  R
CLEN3A  04AC  R   CLEND   049E  R   CLIM1   047A  R   CLIM2   048C  R
CLIM3   0496  R   CLIMOD  0474  R   CNTFLG  014C  R   COREND  04BA  R
DECB    001E  R   DECB44  0594  R   DECBUF  0020  R   ECCLS   0334  R
ECMSG   01D2  R   EIGHT   02FE  R   END1    035A  R   END2    0320  R
FIRST   031A  R   H       0000  A   HCIPL   0152  R   HCIPL1  0150  R
HCIPL2  018A  R   INP2    02DA  R   INP2R   0318  R   INPA    028A  R
INR44   02F4  R   IPL     0104  R   IPL010  0116  R   IPL020  0122  R
IPL100  012C  R   IPL110  0130  R   IPL44   0534  R   IPL88   0000  R
LDFLG   0100  R   MASTFG  022C  R   MLX1    02C4  R   MNLD    03EE  R
MODE    0001  A   NBREC   000C  A   OBJEC   03E2  R   OBJIN1  0316  R
OBJINP  02FE  R   OFLMSG  01D6  R   OTR     0242  R   OUTMSG  023C  R
PCH400  0546  R   PCH405  055A  R   PCH410  055C  R   PCH415  0570  R
PCH420  0572  R   PCH430  0588  R   PRASCA  0384  R   PRASCB  0386  R
PRASCI  0360  R   PROGLD  0416  R   PROLO   040C  R   PROLO1  0418  R
PTR     0000  A   PUNCH   000C  R   PUNCH4  0540  R   RAFL    026A  R
RAFL1   03EA  R   RAFL2   03B4  R   RAFL3   02C2  R   RELAY   03B6  R
RER     02CE  R   S       0001  A   SA      0000  A   SAVA15  04B8  R
SAVBAS  022E  R   SSTMLX  02A0  R   SSTPR1  035C  R   SSTPR2  02DE  R
SSTPTR  0322  R   STAD    01D2  R   STOP    0342  R   SWITCH  03B8  R
SYSMSG  01AA  R   TMTEST  02A4  R   WER1    028C  R   WER2    028E  R


          ASS,ERR,    00000
:EOF
PROG ELAPSED TIME:  00H-00M-00S-000MS-
```

IPL52N

```
00000                                IDENT   IPL52N
00001                                ENTRY   IPL88
00002                                ENTRY   IPL44
00003                          *
00004                          *                          BELIER IPL GENERATION
00005          000C            NBREC   EQU     12
00006                          *
00007          0000            BIPLPR  EQU     *
00008          0000            IPL88   EQU     *
00009  0000    010C                    LDK     A1,NBREC
00010  0002    8220                    LDK,L   A2,BIPL-80
       0004    0080  R
00011  0006    0785                    LDK     A7,/85                BASIC WRITE
00012  0008    80A0                    LDKL    A8,DECB
       000A    001E  R
00013          000C            PUNCH   EQU     *
00014  000C    1250                    ADK     A2,80
00015  000E    8241                    ST      A2,DECBUF             SET BUFF ADDR
       0010    0020  R
00016  0012    2804                    LKM                           PUNCH ONE RECORD
00017  0014    0001                    DATA    1
00018  0016    1901                    SUK     A1,1                  COUNT DONE ?
00019  0018    590E                    RB(1)   PUNCH                 NO
00020  001A    2804                    LKM                           YES
00021  001C    0003                    DATA    3                     EXIT
00022                          *
00023          001E            DECB    EQU     *
00024  001E    0003                    DATA    3                     FILE CODE
00025  0020                    DECBUF  RES     1
00026  0022    0050                    DATA    80
00027  0024    0000                    DATA    0
00028  0026    0000                    DATA    0
00029  0028    0000                    DATA    0
00030                                  EJECT
00031                                  RORG    BIPLPR+/100
00032                          *
00033          0100            BIPL    EQU     *
00034                          *                          1ST RECORD = LOW CORE IPL
00035                          *                          LOW CORE IPL,1 RECORD,LOADED AT /80 AND
00036                          *                          STARTED AT /84
00037                          *
00038  0100    FFFF            LDFLG   DATA    /FFFF                 LDFLG = -1 IF JUST LOADED
00039  0102    0000                    DATA    0                     NOT USED
00040          0104            IPL     EQU     *
00041                          *                          ADDR OF IPL = BASE ADDR
00042  0104    82A0                    LDKL    A10,/84
       0106    0084
00043                          *                          CHECK IF JUST LOADED OR NOT
00044  0108    904B                    IM      LDFLG-IPL,A10
       010A    FFFC
00045  010C    511E                    RF(1)   IPL100                NO,NOT THE 1ST TIME
```

```
00046                          *
00047                          *
00048                          *                                YES,COMPUTE HIGH CORE LIMIT
00049                          *
00050  010E  8720                      LDK.L    A7,/5555              PATTERN
       0110  5555
00051  0112  81A0                      LDKL     A9,/FC00              HIGH CORE =/400
       0114  FC00
00052        0116  IPL010    EQU      *
00053  0116  8727                      STR      A7,A9
00054  0118  EF26                      CWR*     A7,A9                 IF THE LOCATION (A9) EXISTS,(A7)=((A9))
00055  011A  5006                      RF(0)    IPL020                MATCH
00056  011C  99A0                      SUKL     A9,/2000              NO,NEXT LOWER BLOCK OF 4K
       011E  2000
00057  0120  5F0C                      RB       IPL010
00058        0122  IPL020    EQU      *
00059  0122  8386                      LDR      A11,A9                SAVE BASE ADDR OF HIGH CORE IPL
00060  0124  93A0                      ADKL     A11,2                 START ADD.
       0126  0002
00061  0128  8486                      LDR      A12,A9                SAVE LOAD ADD.
00062  012A  5704                      RF       IPL110
00063        012C  IPL100    EQU      *
00064  012C  94A0                      ADKL     A12,80
       012E  0050
00065                          * LOAD ADDRESS OF NEXT RECORD
00066        0130  IPL110    EQU      *
00067  0130  871E                      LDR      A7,A15                RESTORE CIO INST IN BOOT
00068  0132  273F                      ANK      A7,/3F
00069  0134  9720                      ADK.L    A7,/41C0              CIO INSTRUCTION
       0136  41C0
00070  0138  872D                      STR      A7,A3
00071                          *
00072  013A  0557                      LDK      A5,/57                CHANGE LOC. /5A OF BOOT IN ORDER TO
00073  013C  E541                      SC       A5,/5A                CANCEL LEADING CHARACTER FLAG
       013E  005A
00074                          *                                RE INITIALIZE A5,A6
00075  0140  0550                      LDK      A5,80                 # OF CHARACTERS
00076  0142  8612                      LDR      A6,A12                LOAD ADDR.
00077                          *                                CHECK IF COUNT DONE
00078  0144  904B                      IM       CNTFLG-IPL,A10
       0146  0048
00079  0148  890E                      ABR(1)   A11
00080  014A  0F42                      AB       /42                   START BOOT ABAIN
00081        014C  CNTFLG    EQU      *                              = NUMBER OF RECORDS OF HIGH CORE IPL
00082  014C  FFF5                      DATA     1-NBREC
00083  014E  0000                      DATA     0
00084  0150  0000                      DATA     0
00085  0152  0000                      DATA     0
00086  0154  0000                      DATA     0
00087  0156  0000                      DATA     0
00088  0158  0000                      DATA     0
00089  015A  0000                      DATA     0
00090  015C  0000                      DATA     0
00091  015E  0000                      DATA     0
00092  0160  0000                      DATA     0
00093  0162  0000                      DATA     0
00094  0164  0000                      DATA     0
00095  0166  0000                      DATA     0
```

```
00096  0168  0000              DATA    0
00097  016A  0000              DATA    0
00098  016C  0000              DATA    0
00099  016E  0000              DATA    0
00100  0170  0000              DATA    0
00101  0172  0000              DATA    0
00102  0174  0000              DATA    0
00103  0176  0000              DATA    0
00104  0178  0000              DATA    0
00105  017A  0000              DATA    0
00106  017C  0000              DATA    0
00107  017E  0000              DATA    0
00108  0180  0000              DATA    0
00109  0182  0000              DATA    0
00110  0184  0000              DATA    0
00111  0186  0000              DATA    0
00112  0188  0000              DATA    0
00113  018A  0000              DATA    0
00114  018C  0000              DATA    0
00115  018E  0000              DATA    0
00116  0190  0000              DATA    0
00117  0192  0000              DATA    0
00118  0194  0000              DATA    0
00119  0196  0000              DATA    0
00120  0198  0000              DATA    0
00121  019A  0000              DATA    0
00122  019C  0000              DATA    0
00123  019E  0000              DATA    0
00124  01A0  0000              DATA    0
00125  01A2  0000              DATA    0
00126  01A4  0000              DATA    0
00127  01A6  0000              DATA    0
00128  01A8  0000              DATA    0
00129  01AA  0000              DATA    0
00130  01AC  0000              DATA    0
00131  01AE  0000              DATA    0
00132  01B0  0000              DATA    0
00133  01B2  0000              DATA    0
00134  01B4  0000              DATA    0
00135  01B6  0000              DATA    0
00136  01B8  0000              DATA    0
00137  01BA  0000              DATA    0
00138  01BC  0000              DATA    0
00139  01BE  0000              DATA    0
00140  01C0  0000              DATA    0
00141  01C2  0000              DATA    0
00142  01C4  0000              DATA    0
00143  01C6  0000              DATA    0
00144  01C8  0000              DATA    0
00145  01CA  0000              DATA    0
00146  01CC  0000              DATA    0
00147  01CE  0000              DATA    0
00148  01D0  0000              DATA    0
00149  01D2  0000              DATA    0
00150  01D4  0000              DATA    0
00151  01D6  0000              DATA    0
00152  01D8  0000              DATA    0
00153  01DA  0000              DATA    0
```

```
00154  01DC  0000              DATA    0
00155  01DE  0000              DATA    0
00156  01E0  0000              DATA    0
00157  01E2  0000              DATA    0
00158  01E4  0000              DATA    0
00159  01E6  0000              DATA    0
00160  01E8  0000              DATA    0
00161  01EA  0000              DATA    0
00162  01EC  0000              DATA    0
00163  01EE  0000              DATA    0
00164  01F0  0000              DATA    0
00165  01F2  0000              DATA    0
00166  01F4  0000              DATA    0
00167  01F6  0000              DATA    0
00168  01F8  0000              DATA    0
00169  01FA  0000              DATA    0
00170  01FC  0000              DATA    0
00171  01FE  0000              DATA    0
00172  0200  0000              DATA    0
00173  0202  0000              DATA    0
00174  0204  0000              DATA    0
00175                          EJECT
00176                                                  HIGH CORE IPL
00177        0000      PTR     EQU     0               DEVICE ADDR WILL BE INITIALIZED BY HCIPL
00178        0010      ASR     EQU     /10
00179        0001      S       EQU     1
00180        0000      H       EQU     0
00181        0001      BOM     EQU     1
00182        0000      SA      EQU     0
00183                  *
00184        0000      MODE    EQU     SA              FOR MESSAGE  OUTPUT
00185                          RORG    BIPL+80         1ST RECORD OF HIGH CORE IPL
00186        0150      HCIPL1  EQU     *               1ST WORD OF THE CURRENT RECORD
00187  0150  FEFE              DATA    /FEFF
00188        0152      HCIPL   EQU     *
00189  0152  85A0              LDKL    A13,OUTMSG-HCIPL
       0154  00EA
00190  0156  958E              ADR     A13,A11         LOAD A13 WITH ADDR OF OUTMSG ROUTINE
00191  0158  868E              LDR     A14,A11
00192  015A  96A0              ADKL    A14,CFZON-HCIPL
       015C  00E8
00193                  *               ADDR OF STACK
00194  015E  871E              LDR     A7,A15          GET DEVICE ADDR
00195  0160  87CF              ST      A15,SAVA15-HCIPL,A11  SAVE  4*4  FLAG
       0162  0354
00196  0164  5606              RF(6)   *+8
00197  0166  904F              IM      COREND+2-HCIPL,A11
       0168  0358
00198  016A  5C06              RB(4)   *-4
00199  016C  273F              ANK     A7,/3F
00200                  *                       INIT IO INSTRUCTIONS
00201  016E  974F              ADS     A7,CIOPTR-HCIPL,A11
       0170  0142
00202  0172  974F              ADS     A7,INP2-HCIPL,A11
       0174  0188
00203  0176  974F              ADS     A7,END2-HCIPL,A11
       0178  01CE
00204  017A  974F              ADS     A7,END1-HCIPL,A11
       017C  0208
```

```
00205  017E  974F            ADS     A7,INR44-HCIPL,A11
       0180  01A2
00206  0182  811E            LDR     A1,A15
00207  0184  39C8            SLC     A1,8                    TEST MULTI-SINGLE DEVICE CONT,
00208  0186  5602            RF(6)   HCIPL2
00209  0188  270F            ANK     A7,/F                   MULTI DEVICE CONTROLLER
00210        018A    HCIPL2  EQU     *
00211  018A  974F            AD,S    A7,SSTMLX-HCIPL,A11
       018C  014E
00212  018E  974F            ADS     A7,SSTPR2-HCIPL,A11
       0190  018C
00213  0192  974F            ADS     A7,SSTPTR-HCIPL,A11
       0194  01D0
00214  0196  974F            ADS     A7,SSTPR1-HCIPL,A11
       0198  020A
00215  019A  3F41            SLL     A7,1
00216  019C  974F            AD,S    A7,WER1-HCIPL,A11       INIT WER/RER INST
       019E  013A
00217  01A0  974F            AD,S    A7,WER2-HCIPL,A11
       01A2  013C
00218  01A4  974F            AD,S    A7,RER-HCIPL,A11
       01A6  017C
00219  01A8  57AC            RF      CKOK
00220  01AA  4F42    SYSMSG  DATA    'OBJCT TAPE ON RE'
       01AC  4A43
       01AE  5420
       01B0  5441
       01B2  5045
       01B4  204F
       01B6  4E20
       01B8  5245
00221  01BA  4144            DATA    'ADER, THINK OF B'
       01BC  4552
       01BE  2E20
       01C0  5448
       01C2  494E
       01C4  4820
       01C6  4F46
       01C8  2042
00222  01CA  4153            DATA    'ASE 1'
       01CC  4520
       01CE  2120
00223  01D0  0D0A            DATA    /0D0A
00224        01D2    STAD    EQU     *
00225  01D2  4543    ECMSG   DATA    'EC'
00226  01D4  0D0A            DATA    X'0D0A'
00227  01D6  4F56    OFLMSG  DATA    'OVFR'
       01D8  4652
00228  01DA  0D0A            DATA    /0D0A
00229  01DC          BUFF    RES     39
00230  022A  FFFF            DATA    /FFFF
00231  022C  0000    MASTFG  DATA    0
00232  022E  0000    SAVBAS  DATA    0
00233  0230                  RES     6                       * STACK  AREA
00234        023A    CFZON   EQU     *-2
00235        023C    OUTMSG  EQU     *
```

```
00236  023C  0304            LDK       A3,4
00237  023E  43D0            CIO       A3,S,ASR
00238  0240  E324            LCR       A3,A1
00239  0242  4310    OTR     OTR       A3,0,ASR
00240  0244  5C04            RB(4)     *-2
00241  0246  1101            ADK       A1,1
00242  0248  1A01            SUK       A2,1
00243  024A  5C0C            RB(4)     OTR-2
00244  024C  0300            LDK       A3,0
00245  024E  4390            CIO       A3,H,ASR
00246  0250  4BD0            SST       A3,ASR
00247  0252  5C04            RB(4)     *-2
00248  0254  F03A            RTN       A14
00249                 *
00250                 *
00251        0256    CKOK    EQU       *
00252  0256  81A0            LDK,L     A9,0                    SET LOADING ADDRESS
       0258  0000
00253                 *
00254                 *
00255                 *                                   GO LOAD SYSTEM
00256  025A  84A0            LDK,L     A12,MNLD-HCIPL
       025C  029C
00257  025E  948E            ADR       A12,A11
00258  0260  F693            CFR       A14,A12
00259                 *
00260                 *
00261  0262  810A            LDR       A1,A10                  TEST START ADDRESS
00262  0264  8C04            ABR(4)    A1                      EOS  OR EOF HAS BEEN READ
00263  0266  207F            HLT                               NO START ADDRESS
00264  0268  0000    HADDR   DATA      0
00265  026A  0300    RAFL    LDK       A3,0
00266  026C  0700            LDK       A7,0
00267  026E  8520            LDK,L     A5,BUFF-HCIPL
       0270  008A
00268  0272  950E            ADR       A5,A11
00269  0274  80CE            LD        A8,SAVA15-HCIPL,A11
       0276  0354
00270  0278  5610            RF(6)     INPA
00271  027A  0604            LDK       A6,4                    4*4 BO ASR
00272  027C  0111            LDK       A1,/11                  SEND X-ON
00273  027E  46D0            CIO       A6,S,ASR
00274  0280  4110            OTR       A1,0,ASR
00275  0282  5C04            RB(4)     *-2
00276  0284  4290            CIO       A2,H,ASR
00277  0286  4AD0            SST       A2,ASR
00278  0288  5C04            RB(4)     *-2
00279        028A    INPA    EQU       *
00280  028A  0650            LDK       A6,80                   LENGTH
00281  028C  7600    WER1    WER       A6,0                    INIT MLX WHATEVER
00282  028E  7501    WER2    WER       A5,1                    CHANNEL IS
00283  0290  8602            LDR       A6,A8
00284  0292  3E68            SRL       A6,8
00285        0294    CIOPTR  EQU       *
00286  0294  46C0            CIO       A6,S,PTR                AVAILABLE ON THE PAPER READER
00287  0296  5C04            RB(4)     *-2
00288  0298  824E            LD        A2,SAVA15-HCIPL,A11
       029A  0354
```

```
00289  029C  3AC3            SLC     A2,3            MLX ?
00290  029E  523A            RF(2)   INP2            NO
00291        02A0  SSTMLX    EQU     *
00292  02A0  49C0            SST     A1,PTR
00293  02A2  5C04            RB(4)   *-2
00294        02A4  TMTEST    EQU     *
00295  02A4  A120            ANKL    A1,/1007        TAPE MARK OR FATAL ERROR ?
       02A6  1007
00296  02A8  501A            RF(0)   MLX1            NO
00297  02AA  2107            ANK     A1,/7           FATAL ERROR ?
00298  02AC  5486            RF(4)   ECCLS           YES
00299                        *
00300  02AE  8120            LDK,L   A1,'±E'         PUT EOF IN BUFF
       02B0  3A45
00301  02B2  8135            STR     A1,A5
00302  02B4  8120            LDK,L   A1,'OF'
       02B6  4F46
00303  02B8  8155            ST      A1,2,A5
       02BA  0002
00304  02BC  0704            LDK     A7,4            UPDATA POINTRERS
00305  02BE  951C            ADR     A5,A7
00306  02C0  579E            RF      PRASCI
00307                        *
00308  02C2  5F5A  RAFL3     RB      RAFL            RELAY TOWARDS RAFL
00309                        *
00310        02C4  MLX1      EQU     *
00311  02C4  F134            LCR     A1,A5
00312  02C6  E920            CWK     A1,/18          IS IT ASCII
       02C8  0018
00313  02CA  55EA            RF(5)   RELAY           NO
00314  02CC  0750            LDK     A7,80           YES
00315  02CE  7E00  RER       RER     A6,0            READ REMAINING LENGTH
00316  02D0  A620            ANK,L   A6,/FFF
       02D2  0FFF
00317  02D4  9F18            SUR     A7,A6           COMPUTE TRANSMITTED LGT
00318  02D6  951C            ADR     A5,A7           UPDATE BUFF POINTER
00319  02D8  5786            RF      PRASCI
00320  02DA  4A00  INP2      INR     A2,0,PTR
00321  02DC  50DA            RF(0)   SWITCH          INR HAS BEEN ACCEPTED ;
00322                        *                       PROCESS THE CHARTER
00323  02DE  4AC0  SSTPR2    SST     A2,PTR          TRY A SST
00324  02E0  5C08            RB(4)   INP2            TRY AGAIN INR WHEN SST REFUSED
00325  02E2  8108            LDR     A1,A2
00326  02E4  1300            ADK     A3,0            CHECK WETHER ASCII OR OBJECT
00327  02E6  5278            RF(2)   PRASCI          RECORD WAS ASCII ; PRINT IT
00328  02E8  0600            LDK     A6,0            STORE A NON ASCII CHARACTER AT
00329  02EA  8635            STR     A6,A5           THE END OF BUFFER ,
00330                        *                       TO BE USED IN MX1
00331  02EC  5F4A            RB      TMTEST
00332        02EE  OBJINP    EQU     *
00333  02EE  8082            LDR     A8,A8
00334  02F0  560C            RF(6)   EIGHT           8=8 DO NOT INPUT 2ND CHARACTER
00335  02F2  220F            ANK     A2,/F
00336  02F4  4E00  INR44     INR     A6,0,PTR
00337  02F6  5C04            RB(4)   *-2
00338  02F8  260F            ANK     A6,/F
00339  02FA  3A44            SLL     A2,4
00340  02FC  9218            ADR     A2,A6           JOIN THE TWO HALF CHARACTERS
```

```
00341          02FE    EIGHT   EQU     *
00342  02FE    EF04            CWR     A7,A1
00343  0300    501E            RF(0)   END2
00344  0302    E235            SCR     A2,A5
00345  0304    B408            XRR     A4,A2
00346  0306    1501            ADK     A5,1
00347  0308    EF20            CWK     A7,1
       030A    0001
00348  030C    5408            RF(4)   OBJIN1
00349  030E    0400            LDK     A4,0
00350  0310    8108            LDR     A1,A2
00351  0312    9108            ADR     A1,A2
00352  0314    1103            ADK     A1,3
00353  0316    1701    OBJIN1  ADK     A7,1
00354  0318    5F40    INP2R   RB      INP2
00355  031A    2207    FIRST   ANK     A2,/7
00356  031C    0150            LDK     A1,80
00357  031E    5F22            RB(7)   EIGHT
00358  0320    4280    END2    CIO     A2,H,PTR
00359          0322    SSTPTR  EQU     *
00360  0322    49C0            SST     A1,PTR
00361  0324    5C04            RB(4)   *-2
00362  0326    8082            LDR     A8,A8                   IF ASR WAIT
00363  0328    5606            RF(6)   *+8
00364  032A    904F            IM      COREND+2-HCIPL,A11      WAIT
       032C    0358
00365  032E    5C06            RB(4)   *-4
00366  0330    24FF            ANK     A4,/FF
00367  0332    50D4            RF(0)   PROL01+2
00368          0334    ECCLS   EQU     *
00369  0334    0180            LDK     A1,ECMSG-HCIPL
00370  0336    910E            ADR     A1,A11
00371  0338    0204            LDK     A2,4
00372  033A    84A0            LDK.L   A12,OUTMSG-HCIPL
       033C    00EA
00373  033E    948E            ADR     A12,A11
00374  0340    F693            CFR     A14,A12
00375  0342    207F    STOP    HLT
00376  0344    5FDC            RB(7)   RAFL
00377  0346    EA20    ASCINP  CWK     A2,/0D
       0348    000D
00378  034A    500E            RF(0)   END1
00379  034C    EF20            CWK     A7,68
       034E    0044
00380  0350    5878            RB(0)   INP2
00381  0352    E235            SCR     A2,A5
00382  0354    1501            ADK     A5,1
00383  0356    1701            ADK     A7,1
00384  0358    5F80            RB(7)   INP2
00385  035A    4280    END1    CIO     A2,H,PTR
00386          035C    SSTPR1  EQU     *
00387  035C    4AC0            SST     A2,PTR
00388  035E    5C04            RB(4)   *-2
00389          0360    PRASCI  EQU     *
00390  0360    018A            LDK     A1,BUFF-HCIPL
00391  0362    910E            ADR     A1,A11
00392  0364    821C            LDR     A2,A7
00393  0366    0320            LDK     A3,/20
```

```
00394  0368  E335            SCR     A3,A5
00395  036A  1501            ADK     A5,1
00396  036C  8320            LDK,L   A3,/0D0A
       036E  0D0A
00397  0370  8335            STR     A3,A5
00398  0372  1203            ADK     A2,3
00399  0374  3A61            SRL     A2,1
00400  0376  3A41            SLL     A2,1
00401  0378  8082            LDR     A8,A8                   IF ASR DO NOT OUTPUT ASCII
00402  037A  5608            RF(6)   PRASCA                  A SECOND TIME (ALREADY PRINTED WHEN READ
00403  037C  904F            IM      COREND+2-HCIPL,A11      BUT WAIT ABIT
       037E  0358
00404  0380  5C06            RB(4)   *-4
00405  0382  5702            RF      PRASCB
00406        0384    PRASCA  EQU     *
00407  0384  1100            ADK     A1,0                    PASS WAS CFR A14,A13 IN IPL52S
00408                * TO AVOID PRINT MESSAGE
00409        0386    PRASCB  EQU     *
00410  0386  028A            LDK     A2,BUFF-HCIPL
00411  0388  920E            ADR     A2,A11
00412  038A  8328            LDR*    A3,A2
00413  038C  EB20            CWK     A3,/3A45                * 1E
       038E  3A45
00414  0390  5CD0            RB(4)   RAFL3
00415  0392  1202            ADK     A2,2
00416  0394  8328            LDR*    A3,A2
00417  0396  EB20            CWK     A3,/4F46                *OF
       0398  4F46
00418  039A  5CDA            RB(4)   RAFL3
00419                *
00420                *
00421                *                               SET RETURN PARAMETERS
00422                *
00423  039C  82CE            LD      A10,COREND-HCIPL,A11    START ADDRESS IN A10
       039E  0356
00424  03A0  84A0            LDK,L   A12,MNLD-HCIPL
       03A2  029C
00425  03A4  948E            ADR     A12,A11                 MNLD ADDRESS  IN A12
00426  03A6  81CE            LD      A9,SAVBAS-HCIPL,A11     LOADING BASE  IN A9
       03A8  00DC
00427  03AA  80CE            LD      A8,BADDR-HCIPL,A11      ENDING ADDRESS IN A8
       03AC  0116
00428  03AE  874E            LD      A7,MASTFG-HCIPL,A11     MASTER FLAG IN A7
       03B0  00DA
00429                *
00430                *
00431  03B2  F03A            RTN     A14                     RETURN TO CALLING
00432                *
00433  03B4  5FF4    RAFL2   RB      RAFL3                   * RELAY TOWARDS RAFL
00434  03B6  5750    RELAY   RF      PROL01+2
00435                *
00436  03B8  1300    SWITCH  ADK     A3,0
00437  03BA  59CE            RB(1)   OBJINP
00438  03BC  227F            ANK     A2,/7F
00439  03BE  58E6            RB(0)   INP2
00440  03C0  EA20            CWK     A2,/7F
       03C2  007F
00441  03C4  58EC            RB(0)   INP2
```

```
00442  03C6  1300              ADK       A3,0
00443  03C8  5A84              RB(2)     ASCINP
00444  03CA  EA20              CWK       A2,/1F
       03CC  001F
00445  03CE  5116              RF(1)     ASCII
00446  03D0  EA20              CWK       A2,/14
       03D2  0014
00447  03D4  510C              RF(1)     OBJEC
00448  03D6  EA20              CWK       A2,8
       03D8  0008
00449  03DA  5206              RF(2)     OBJEC
00450  03DC  EA20              CWK       A2,/10
       03DE  0010
00451  03F0  5CCA              RB(4)     INP2R
00452  03E2  1301      OBJEC   ADK       A3,1
00453  03F4  5FCC              RB(7)     FIRST
00454  03E6  1B01      ASCII   SUK       A3,1
00455  03E8  5FA4              RB(7)     ASCINP
00456                          EJECT
00457                  *
00458                  *
00459                  *  MAIN  LOADING  PART          ENTRY PARAMETERS :
00460                  *
00461                  *                                 A1 = MASTER FLAG
00462                  *                                 A9 = BASE ADDRESS
00463                  *
00464                  *
00465                  *
00466                  *                 1- OUTPUT MSG REQUESTING  TAPE ON READER
00467                  *                 2- PERFORMS A 'HALT' - THE USER HAS THEN THE
00468                  *                    POSSIBILITY TO ALTER THE BASE ADDRESS (REG,A9)
00469                  *                                  AND  THE MASTER FLAG  (REG,A1)
00470                  *                                                  (A1=0 MASTER)
00471                  *                                                  (A1=1 USER)
00472                  *                 3- LOADING PROCESS STARTS  WHEN USER DEPPRESS START
00473                  *                    BUTTON
00474                  *
00475                  *
00476                  *
00477                  *
00478         03EA     RAFL1   EQU       *
00479  03EA   5F38              RB        RAFL2
00480                  *
00481                  *
00482                  *                              ERRONNOUS CLUSTER,PRINT ERR MESSAGE
00483         03EC     CLCO1   EQU       *
00484  03EC   5FBA              RB        ECCLS
00485                  *
00486                  *
00487                  *
00488                  *
00489         03EE     MNLD    EQU       *
00490  03FE   8404              LDR       A4,A1                  SAVE  MASTER FLAG
00491  03F0   0200              LDK       A2,0
00492  03F2   824F              ST        A2,CORENDHCIPL,A11     RAZ  START  ADDRESS
       03F4   0356
00493                          IFT       MODE=BOM
00505                          XIF
```

```
00506  03F6  81CF              ST       A9,SAVBAS-HCIPL,A11
       03F8  00DC
00507                  *
00508                  *
00509                  *
00510                  * PROCESS LOADING : THIS  MODULE READ A  CLUSTER
00511                  *          AND  BRANCH ACCORDING TO THE CLUSTER TYPE
00512                  *          ON EXIT    A1 = BUFF  ADDRESS +1
00513                  *                     A2 = WORD  COUNT
00514                  *                     A3 = TYPE
00515                  *          THE TYPE MUST BE    3,4,7    IF NOT THIS :HALT
00516  03FA  82A0      PROLO    LDK,L    A10,STAD-HCIPL          END ADDRESS
       03FC  0080
00517  03FE  928E              ADR      A10,A11
00518        0000      ABA      EQU      0
00519  0400  81CF              ST       A9,BADDR-HCIPL,A11      BADDR =BASE ADDRESS
       0402  0116
00520  0404  20BF      PROGLD   INH
00521        0406      PROL01   EQU      *
00522  0406  5F1E              RB       RAFL1
00523  0408  8120              LDK,L    A1,BUFF-HCIPL
       040A  008A
00524  040C  910E              ADR      A1,A11
00525  040E  0401              LDK      A4,1
00526  0410  0300              LDK      A3,0
00527  0412  F324              LCR      A3,A1                   A3 = TYPE
00528  0414  1101              ADK      A1,1
00529  0416  0200              LDK      A2,0
00530  0418  E224              LCR      A2,A1                   A2 = WORD COUNT
00531  041A  1101              ADK      A1,1
00532  041C  EB20              CWK      A3,3
       041E  0003
00533  0420  500E              RF(0)    CLCODE                  BRANCH  ON CLUSTER CODE
00534  0422  EB20              CWK      A3,4
       0424  0004
00535  0426  503A              RF(0)    CLIMOD                  INTERNAL MODIFICATION
00536  0428  EB20              CWK      A3,7
       042A  0007
00537  042C  505E              RF(0)    CLEND                   END/START
00538  042E  5F2A              RB(7)    PROL01
00539                  ************************************
00540                  * CLUSTER CODE  TYPE 3
00541                  *          UPON ENTRY: A1=ADDRESS OF BUFF+1 (RBK
00542                  *                      A2=WORD COUNT
00543                  *                      A9=BADDRESS
00544                  *                      A10=ENDADDRESS
00545                  ********************************************
00546  0430  834F      CLCODE   LD       A3,BUFF+6-HCIPL,A11
       0432  0090
00547  0434  5C30              RB(4)    PROL01                  EMBK SET SKIP THE CLUSTER
00548  0436  834F      CLC01A   LD       A3,BUFF+4-HCIPL,A11
       0438  008E
00549  043A  A311              TM       A3,A4                   IS IT  RELOCATABLE SECTION
00550  043C  5004              RF(0)    CLC04
00551  043E  3301              XRK      A3,1
00552  0440  9306              ADR      A3,A9
00553  0442  8524      CLC04    LDR*     A5,A1                   A5=(RBK)
00554  0444  1106              ADK      A1,6                    A1= ADDRESS OF ST CODE WORD IN BUFF
```

```
00555  0446  1A03              SUK      A2,3                    A2= NUMBER OF CODE WORD
00556                    *                                  A3= STORAGE ADDRESS
00557                    *                                  A4= MASK FOR RBK
00558                    *                                  A6= CODE WORD
00559  0448  3CE1       CLC05  SRC      A4,1
00560  044A  8624              LDR*     A6,A1
00561  044C  EB0A              CWR      A3,A10                  COMPARE LOAD ADDRESS WITH AD OF IPL
00562  044E  5864              RB(0)    CLC01
00563  0450  A511              TM       A5,A4
00564  0452  5002              RF(0)    CLC07
00565  0454  9606              ADR      A6,A9
00566  0456  862D       CLC07  STR      A6,A3                   STORE CODE W=RDS
00567  0458  1102              ADK      A1,2
00568  045A  1302              ADK      A3,2
00569  045C  1A01              SUK      A2,1
00570  045E  5C18              RB(4)    CLC05
00571  0460  5F68              RB(7)    PROL0
00572                    *
00573                    ****************************************************************
00574                    * INTERNAL MODIFICATION CLUSTER
00575                    ***********************************************
00576  0462  0701       CLIMOD LDK      A7,1                    A7= MASK FOR ADDRESS
00577  0464  8524              LDR*     A5,A1                   A5=(RBK)
00578  0466  1A01              SUK      A2,1
00579  0468  3CE1       CLIM1  SRC      A4,1
00580  046A  1102              ADK      A1,2
00581  046C  8324              LDR*     A3,A1                   A3=ADDRESS
00582  046E  A31D              TM       A3,A7                   IS IT RELOCATABLE
00583  0470  5008              RF(0)    CLIM2                   NO
00584  0472  3301              XRK      A3,1
00585  0474  9306              ADR      A3,A9                   YES AD BASE
00586  0476  EB0A              CWR      A3,A10                  ADDRESS OK
00587  0478  588E              RB(0)    CLC01
00588  047A  1102       CLIM2  ADK      A1,2
00589  047C  8624              LDR*     A6,A1                   TAKE CODE WORD
00590  047E  A511              TM       A5,A4                   IS IT RELOCATABLE
00591  0480  5002              RF(0)    CLIM3
00592  0482  9606              ADR      A6,A9                   AD BASE
00593  0484  862D       CLIM3  STR      A6,A3                   STORE CODE WORD
00594  0486  1A02              SUK      A2,2
00595  0488  5C22              RB(4)    CLIM1
00596  048A  5F92              RB(7)    PROL0
00597                    ****************************************
00598                    *  CLUSTER  END/START
00599                    ****************************************
00600  048C  8324       CLEND  LDR*     A3,A1
00601  048E  500A              RF(0)    CLEN3A                  FINISH WO START
00602  0490  A311              TM       A3,A4
00603  0492  5002              RF(0)    CLEN1
00604  0494  9306              ADR      A3,A9
00605                    *
00606                    *
00607        0496       CLEN1  EQU      *
00608  0496  834F              ST       A3,COREND-HCIPL,A11
       0498  0356
00609  049A  814E       CLEN3A LD       A1,BUFF+6-HCIPL,A11     UPDATE BASE ADDRESS
       049C  0090
```

```
00610  049E  914F            AD,S     A1,BADDR=HCIPL,A11
       04A0  0116
00611  04A2  9184            ADR      A9,A1
00612  04A4  5FAC            RB       PROLO
00613                  *
00614                  *
00615  04A6  0000      SAVA15  DATA     0
00616                  *
00617                  ********************************************************************************
00618                  *
00619  04A8  0000      COREND  DATA     0
00620                  *
00621                  *
00622                  ********************************************************************************
00623  04AA  0000              DATA     0
00624  04AC  0000              DATA     0
00625  04AE  0000              DATA     0
00626  04B0  0000              DATA     0
00627  04B2  0000              DATA     0
00628  04B4  0000              DATA     0
00629  04B6  0000              DATA     0
00630  04B8  0000              DATA     0
00631  04BA  0000              DATA     0
00632  04BC  0000              DATA     0
00633  04BE  0000              DATA     0
00634  04C0  0000              DATA     0
00635  04C2  0000              DATA     0
00636  04C4  0000              DATA     0
00637  04C6  0000              DATA     0
00638  04C8  0000              DATA     0
00639  04CA  0000              DATA     0
00640  04CC  0000              DATA     0
00641  04CE  0000              DATA     0
00642  04D0  0000              DATA     0
00643  04D2  0000              DATA     0
00644  04D4  0000              DATA     0
00645  04D6  0000              DATA     0
00646  04D8  0000              DATA     0
00647  04DA  0000              DATA     0
00648  04DC  0000              DATA     0
00649  04DE  0000              DATA     0
00650  04E0  0000              DATA     0
00651  04E2  0000              DATA     0
00652  04E4  0000              DATA     0
00653  04E6  0000              DATA     0
00654  04F8  0000              DATA     0
00655  04EA  0000              DATA     0
00656  04EC  0000              DATA     0
00657  04EE  0000              DATA     0
00658  04F0  0000              DATA     0
00659  04F2  0000              DATA     0
00660  04F4  0000              DATA     0
00661  04F6  0000              DATA     0
00662  04F8  0000              DATA     0
00663  04FA  0000              DATA     0
00664  04FC  0000              DATA     0
00665  04FE  0000              DATA     0
00666  0500  0000              DATA     0
```

```
00667  0502  0000                DATA    0
00668  0504  0000                DATA    0
00669  0506  0000                DATA    0
00670  0508  0000                DATA    0
00671  050A  0000                DATA    0
00672  050C  0000                DATA    0
00673  050E  0000                DATA    0
00674  0510  0000                DATA    0
00675  0512  0000                DATA    0
00676  0514  0000                DATA    0
00677  0516  0000                DATA    0
00678  0518  0000                DATA    0
00679  051A  0000                DATA    0
00680  051C  0000                DATA    0
00681  051E  0000                DATA    0
00682  0520  0000                DATA    0
00683                            EJECT
00684        0522      IPL44     EQU     *
00685  0522  030C                LDK     A3,NHREC
00686  0524  8420                LDKI    A4,BIPL
       0526  0100  R
00687  0528  0785                LDK     A7,/85
00688                  * BASIC WRITE
00689  052A  80A0                LDKL    A8,DECH44
       052C  0582  R
00690        052E      PUNCH4    EQU     *
00691  052E  0550                LDK     A5,80
00692  0530  8620                LDKL    A6,BUF44
       0532  058E  R
00693        0534      PCH400    EQU     *
00694  0534  0100                LDK     A1,0
00695  0536  0200                LDK     A2,0
00696  0538  E130                LCR     A1,A4
00697  053A  3964                SRL     A1,4
00698  053C  E921                CCK     A1,0
       053E  0000
00699  0540  5006                RF(0)   PCH405
00700  0542  E921                CCK     A1,/500
       0544  0500
00701  0546  5202                RF(2)   PCH410          LESS THAN 5
00702        0548      PCH405    EQU     *
00703  0548  2910                ORK     A1,/10
00704        054A      PCH410    EQU     *
00705  054A  3948                SLL     A1,8
00706  054C  E130                LCR     A1,A4
00707  054E  A120                ANKL    A1,/FF0F
       0550  FF0F
00708  0552  F921                CCK     A1,0
       0554  0000
00709  0556  5006                RF(0)   PCH415
00710  0558  E921                CCK     A1,/500
       055A  0500
00711  055C  5202                RF(2)   PCH420
00712        055E      PCH415    EQU     *
00713  055E  2910                ORK     A1,/10
00714        0560      PCH420    EQU     *
00715  0560  8139                STR     A1,A6
00716                  * STORE WORD
```

```
00717  0562  1602                ADK       A6,2                  NEXT WORD IN BUFFER
00718  0564  1401                ADK       A4,1                  NEXT CHZR, IN IPL
00719  0566  1D01                SUK       A5,1
00720                      *
00721  0568  5936                RB(1)     PCH400
00722  056A  EB20                CWK       A3,1                  LAST RECORD ?
       056C  0001
00723  056E  5406                RF(4)     PCH430                NO
00724  0570  0113                LDK       A1,/13                XOFF
00725  0572  8141                ST        A1,BUF44+158
       0574  062C  R
00726        0576          PCH430  EQU       *
00727  0576  2804                LKM
00728  0578  0001                DATA      1
00729  057A  1801                SUK       A3,1
00730  057C  5950                RB(1)     PUNCH4
00731                      *
00732  057E  2804                LKM
00733  0580  0003                DATA      3
00734                      *
00735        0582          DECB44  EQU       *
00736  0582  0003                DATA      3
00737  0584  058E  R             DATA      BUF44
00738  0586  00A0                DATA      160
00739  0588  0000                DATA      0
00740  058A  0000                DATA      0
00741  058C  0000                DATA      0
00742  058E                BUF44   RES       80
00743                      *
00744                      *
00745                      *
00746                              END       BIPLPR
```

SYMBOL TABLE

```
ABA     0000  A   ASCII   03E6  R   ASCINP  0346  R   ASR     0010  A
BADDR   0268  R   BIPL    0100  R   BIPLPR  0000  R   BOM     0001  A
BUF44   058E  R   BUFF    01DC  R   CFZON   023A  R   CIOPTR  0294  R
CKOK    0256  R   CLC01   03EC  R   CLC01A  0436  R   CLC04   0442  R
CLC05   0448  R   CLC07   0456  R   CLCODE  0430  R   CLEN1   0496  R
CLEN3A  049A  R   CLEND   048C  R   CLIM1   0468  R   CLIM2   047A  R
CLIM3   0484  R   CLIMOD  0462  R   CNTFLG  014C  R   CORFND  04A8  R
DECB    001E  R   DECB44  0582  R   DECBUF  0020  R   ECCLS   0334  R
ECMSG   01D2  R   EIGHT   02FE  R   END1    035A  R   END2    0320  R
FIRST   031A  R   H       0000  A   HCIPL   0152  R   HCIPL1  0150  R
HCIPL2  018A  R   INP2.   02DA  R   INP2R   0318  R   INPA    028A  R
INR44   02F4  R   IPL     0104  R   IPL010  0116  R   IPL020  0122  R
IPL100  012C  R   IPL110  0130  R   IPL44   0522  R   IPL88   0000  R
LDFLG   0100  R   MASTFG  022C  R   MLX1    02C4  R   MNLD    03EE  R
MODE    0000  A   NBREC   000C  A   OBJEC   03E2  R   OBJIN1  0316  R
OBJINP  02EE  R   OFLMSG  01D6  R   OTR     0242  R   OUTMSG  023C  R
PCH400  0534  R   PCH405  0548  R   PCH410  054A  R   PCH415  055E  R
PCH420  0560  R   PCH430  0576  R   PRASCA  0384  R   PRASCB  0386  R
PRASCI  0360  R   PROGLD  0404  R   PROLO   03FA  R   PROL01  0406  R
PTR     0000  A   PUNCH   000C  R   PUNCH4  052E  R   RAFL    026A  R
RAFL1   03FA  R   RAFL2   03B4  R   RAFL3   02C2  R   RELAY   03B6  R
RER     02CE  R   S       0001  A   SA      0000  A   SAVA15  04A6  R
SAVBAS  022E  R   SSTMLX  02A0  R   SSTPR1  035C  R   SSTPR2  02DE  R
SSTPTR  0322  R   STAD    01D2  R   STOP    0342  R   SWITCH  0388  R
SYSMSG  01AA  R   TMTEST  02A4  R   WER1    028C  R   WER2    028E  R
```

6.28 SIMPLE TEST PROGRAMS

The Simple Test Programs in this section and the ones in Volume 2 are all Stand Alone programs that can be loaded into the memory direct from the control panel switches. They should be tried if the Standard Test Programs either cannot be loaded or will not run. To avoid errors which may be introduced when loading the programs by hand it is suggested that once the program has been loaded and tested it should be reproduced on a suitable input media so that for future use it can be loaded using the IPL routine. Methods of producing programs on an input media, together with suggestions for producing more sophisticated programs, will be found in paragraphs 6-36 to 6-51 of this section.

6.29 Memory Test Program MEMHAN

If the Standard Memory Test Program (BZRAM) cannot be used, or if you wish to test only a certain area of memory with a particular test pattern, try the following program which can be loaded either by hand from the Control Panel switches or by the IPL routine if it has been reproduced on a suitable input media. If the program is loaded from the Control Panel switches it can be loaded into any area of memory, but if it is loaded by the IPL routine the addresses of the instructions will be as shown in the program listing. The program loads the Test Pattern into each consecutive memory location indicated by the addresses in registers A11 and A12. When all the required locations have been loaded, the program reads each location and compares them with the data in register A13. If an error is found during the read part of the program it will stop and the address where the error has occurred will be found in register A9, the pattern read will be found in register A8, and the expected pattern will be in register A13. If you wish the program to continue after an error has been detected press the RUN button. If there are no errors the program will run until the contents of A7 are zero.

MEMHAM

```
00000                          IDENT    MEMHAN
00001          0000    BEGIN   EQU      *
00002                          RORG     BEGIN+/80
00003                  *
00004                  *       THIS PROGRAM ENABLES THE ENGINEER TO WRITE AND CHECK REA
00005                  *       A SELECTED PATTERN INTO A SELECTED AREA OF MEMORY AND TO
00006                  *       REPEAT THE TEST A SELECTED NUMBER OF TIMES UP TO 255
00007                  *
00008                  *        LOAD THE NUMBER OF TIMES YOU WANT THE TEST REPEATED INTO
00009                  *       REGISTER A7 FROM THE CONTROL PANEL
00010                  *
00011                  *       LOAD THE START ADDRESS OF THE AREA TO BE TESTED INTO
00012                  *       REGISTER A11 FROM THE CONTROL PANEL
00013                  *
00014                  *       LOAD THE END ADDRESS OF THE AREA TO BE TESTED INTO
00015                  *       REGISTER A12 FROM THE CONTROL PANEL
00016                  *
00017                  *       LOAD THE TEST PATTERN INTO REGISTER A13 FROM THE CONTROL PANEL
00018                  *
00019                  *       LOAD THE START ADDRESS OF THE PROGRAM INTO REGISTER A0 FROM THE
00020                  *       CONTROL PANEL THEN PUSH THE RUN BUTTON
00021                  *
00022                  *
00023  0080  FFFF              DATA     /FFFF
00024  0082  0000              DATA     0
00025  0084  207F      START   HLT
00026  0086  818E      WRITE   LDR      A9,A11          LOAD START ADDRESS INTO A9
00027  0088  85A7      LOAD    STR      A13,A9          STORE TEST PATTEN INTO MEMORY
00028  008A  91A0              ADKL     A9,2            UPDATE MEMORY ADDRESS COUNTER
       008C  0002
00029  008E  E992              CWR      A9,A12          CHECK FOR END ADDRESS IN A9
00030  0090  5D0A              RB(NG)   LOAD            IF NOT EQUAL LOAD NEXT ADDRESS
00031  0092  818E              LDR      A9,A11          LOAD START ADDRESS INTO A9
00032  0094  80A6      READ    LDR*     A8,A9           LOAD CONTENTS OF A9'S MEMORY ADDRESS
00033  0096  E896              CWR      A8,A13          AND COMPARE WITH PATTERN IN A13
00034  0098  5002              RF(Z)    CONT            IF EQUAL CONTINUE WITH READ ROUTINE
00035  009A  207F              HLT                      IF NOT EQUAL STOP
00036  009C  91A0      CONT    ADKL     A9,2            UPDATE ADDRESS COUNTER
       009E  0002
00037  00A0  E992              CWR      A9,A12          CHECK FOR END ADDRESS IN A9
00038  00A2  5D10              RB(NG)   READ            IF NOT EQUAL READ NEXT WORD
00039  00A4  1F01              SUK      A7,1            SUBTRACT 1 FROM REPEAT COUNTER
00040  00A6  5C22              RB(NZ)   WRITE           AND START NEXT W/R CYCLE IF NOT ZERO
00041  00A8  5F26              RB       START           WHEN FINISHED GO TO START OF PROGRAM FOR
00042                  *                                POSSIBLE NEW PARAMETERS
00043                          END      START
```

6.30 Program CHECK

This program enables you to check the type of input from the Operator's device i.e. 8-bit data no parity, 7-bit data with parity etc. The program can be loaded by hand with the control panel switches or by using the IPL routine. Once loaded it is only necessary to load the start address into register A0 and then push the RUN button. You can now type in any ten characters from the keyboard which are stored in the Buffer (address 00A8). When the buffer is full the program branches back to the start of program and you can check the codes received for each character using the Read Memory Routine. If you wish to check more characters push the RUN button and repeat as necessary. This program is useful if you wish to check for certain characters in your programs.

CHECK

```
00000                                IDENT    CHECK
00001          0000      BEGIN     EQU      *
00002                                RORG     BEGIN+/80
00003                      *
00004                      *         THIS PROGRAM ENABLES THE ENGINEER TO CHECK
00005                      *         THE CHARACTER CODE FROM THE OPERATORS DEVICE
00006                      *
00007                      *
00008                      *         LOAD THE PROGRAM START ADDRESS INTO A0 FROM THE
00009                      *         CONTROL PANEL - THEN PUSH THE RUN BUTTON
00010                      *
00011                      *
00012  0080    FFFF                  DATA     /FFFF
00013  0082    0000                  DATA     0
00014  0084    207F      START     HLT
00015  0086    208F                  INH
00016                      *
00017  0088    0300                  LDK      A3,0                 ZEROISE BUFF COUNTER
00018  008A    0201                  LDK      A2,1                 LOAD PARAMETERS FOR INPUT CHARACTER
00019                      *                                                                WITH NO
00020  008C    42D0                  CIO      A2,1,/10             SEND START COMMAND TO DEVICE
00021  008E    5C04                  RB(NA)   *-2
00022  0090    4D10      INCHAR      INR      A5,0,/10             SEND INR INSTRUCTION TO DEVICE
00023  0092    5C04                  RB(NA)   *-2                  AND WAIT FOR CHAR TO BE INPUT
00024  0094    E54D                  SC       A5,BUFF,A3           STORE CHAR IN BUFF
       0096    00A8  R
00025  0098    1301                  ADK      A3,1                 UPDATE BUFF COUNTER
00026  009A    EB20                  CWK      A3,10                CHECK FOR BUFFER FULL
       009C    000A
00027  009E    5C10                  RB(NE)   INCHAR               ACCEPT NEW CHAR IF NOT FULL
00028  00A0    4290                  CIO      A2,0,/10             STOP INPUT FUNCTION
00029  00A2    4AD0                  SST      A2,/10               AND GET STATUS
00030  00A4    5C04                  RB(NA)   *-2
00031  00A6    5F24                  RB       START
00032  00A8              BUFF        RES      5
00033                                END      START
```

6.31 ASR, PER3100, and Display

If the Standard Test Program either cannot be loaded or will not run, one of the following programs should be tried. They can all be loaded either by the Control Panel switches or by the IPL routine. Two of the programs (LINE and NOECHO) have been written for devices that have been wired for Even Parity so if your device is wired for any other mode of operation the parameters loaded into register A2 must be changed accordingly.

6.32 Program LINE

This program can be loaded either from the Control Panel switches or by using the IPL routine. If the program is loaded by hand it can be loaded into any area of memory, but if loaded by the IPL routine the locations of the instructions will be as shown in the program listing. The program enables a selected ASCII character to be sent to the device a selected number of times (the number of times selected MUST NOT exceed the maximum number of characters per line specified for the device being tested), then it sends the Carriage Return/Line Feed characters and branches back to the start of program and stops ready for a new character to be selected. If you wish the program can be modified to run continuously by changing two of the instructions as follows:

- Change the LDR instruction in line 28 to LDK A4,64 hexadecimal value 0440.
- Change the RB instruction value in line 42 so that it branches back to line 24 of the program by using hexadecimal value 5F24.

Once the RUN button has been pushed it will then run and continue repeating the same line until you stop the program by pushing the INST button.

LINE

```
00000                          IDENT   LINE
00001          0000    BEGIN   EQU     *
00002                          RORG    BEGIN+/80
00003                  *
00004                  *       THIS PROGRAM ENABLES THE ENGINEER TO SEND A SELECTED
00005                  *       ASCII CHAR TO THE OPERATOR'S DEVICE A SELECTED
00006                  *       NUMBER OF TIMES
00007                  *
00008                  *
00009                  *       LOAD THE ASCII CHARACTER INTO REGISTER A6 FROM THE
00010                  *       CONTROL PANEL
00011                  *
00012                  *       LOAD THE NUMBER OF TIMES YOU WANT IT REPEATED
00013                  *       INTO REGISTER A7 FROM THE CONTROL PANEL
00014                  *
00015                  *       LOAD THE PROGRAM START ADDRESS INTO REGISTER A0
00016                  *       FROM THE CONTROL PANEL AND PUSH THE RUN BUTTON
00017                  *
00018                  *
00019  0080  FFFF              DATA    /FFFF
00020  0082  0000              DATA    0
00021  0084  207F      START   HLT
00022  0086  20BF              INH                             THIS PROGRAM RUNS IN INHIBIT MODE
00023                  *
00024  0088  0204              LDK     A2,4                    LOAD PARAMETERS FOR OUTPUT CHARACTER
00025                  *                                       WITH EVEN PARITY
00026  008A  42D0              CIO     A2,1,/10                SEND START COMMAND TO DEVICE
00027  008C  5C04              RB(NA)  *-2                     KEEP TRYING UNTIL ACCEPTED
00028  008E  841C              LDR     A4,A7                   LOAD COUNT INTO A4
00029  0090  4610      OUCHAR  OTR     A6,0,/10                OUTPUT ASCII CHAR TO DEVICE
00030  0092  5C04              RB(NA)  *-2                     TRY AGAIN IF NOT ACCEPTED
00031  0094  1C01              SUK     A4,1                    SUBTRACT ONE FROM COUNT
00032  0096  5C08              RB(NZ)  OUCHAR                  SEND CHAR AGAIN IF NOT ZERO
00033                  *                                       IF ZERO
00034  0098  8520      CRLF    LDKL    A5,/0A0D                LOAD CR/LF CHARS
       009A  0A0D
00035  009C  4510      OUCRLF  OTR     A5,0,/10                SENDS CR/LF CHARS
00036  009E  5C04              RB(NA)  *-2
00037  00A0  3D68              SRL     A5,8                    POSITIONS 0A CHAR FOR SENDING
00038  00A2  5C08              RB(NZ)  OUCRLF                  BRANCHES FIRST TIME TO SEND SECOND CHAR
00039  00A4  4290              CIO     A2,0,/10                SENDS STOP COMMAND TO DEVICE
00040  00A6  4AD0              SST     A2,/10                  AND THEN GETS STATUS WORD
00041  00A8  5C04              RB(NA)  *-2
00042  00AA  5F28              RB      START                   RETURN TO START  COUNT DONE
00043                          END     START
```

6.33 Programs NOECHO and ECHO

These programs enable you to verify the input/output functions of the peripheral and the SCUZ card and they can be loaded either by hand from the Control Panel or by using the IPL routine. The only difference between the programs is that NOECHO should be used if the ECHO U link (on the SCUZ card) has been positioned to select non-Echo mode, and ECHO should be used if the U link is positioned to select ECHO mode. With the program NOECHO it is the program that sends the character back to the device to be either printed or displayed, and with the program ECHO it is the SCUZ card logic which sends the character back to the device. Once loaded you must load the start address in register A0 and then push the RUN button, thereafter the program will run until stopped by pushing the INST button. Any printable character typed in via the keyboard will be reproduced on the console log, and any control characters (e.g. CR LF) are accepted and executed.

6.34 Program COPY

This program enables you to punch a copy of any paper tape. It can be loaded by hand from the Control Panel or by using the IPL routine. Once loaded push the FEED HOLES button on the Paper Tape Punch to provide a leader of about 10cm, then load the paper tape you want punched on the Paper Tape Reader and push the RUN button. The program will read the tape and punch a copy simultaneously. When the Reader and Punch stop push the FEED HOLES button to provide a trailing edge for the tape. NOTE: The program bas been written to reproduce any length of paper tape so no character counter has been included so the program must be loaded for each tape you wish to reproduce.

6.35 Program DUMP

This program enables the engineer to dump a selected area of memory on either the ASR or PER3100. It can be loaded either by hand from the Control Panel switches or using a program tape that contains the MINI-IPL (paragraph 6.44) and program produced by one of the methods described in paragraph 6.37 to 6.42. If the MINI-IPL method is used care must be taken that the program does not overwrite part of the area you wish dumped. The instructions for operating this program are given in the program listing and when the last address has been dumped the program branches back to the start ready for possible new parameters.

NOECHO

```
00000                          IDENT   NOECHO
00001          0000    BEGIN   EQU     *
00002                          RORG    BEGIN+/80
00003                  *
00004                  *       THIS PROGRAM ENABLES THE ENGINEER TO VERIFY THE INPUT/OUTPUT
00005                  *       FUNCTIONS OF THE CU AND ITS PERIPHERAL
00006                  *
00007                  *
00008                  *       LOAD THE PROGRAM START ADDRESS INTO A0 FROM THE
00009                  *       CONTROL PANEL - THEN PUSH THE RUN BUTTON
00010                  *
00011                  *
00012  0080  FFFF              DATA    /FFFF
00013  0082  0000              DATA    0
00014  0084  207F      START   HLT
00015  0086  208F              INH
00016                  *
00017  0088  0205      INCHAR  LDK     A2,5            LOAD PARAMETERS FOR INPUT CHARACTER
00018                  *                               WITH EVEN PARITY
00019  008A  42D0              CIO     A2,1,/10        SEND START COMMAND TO DEVICE
00020  008C  5C04              RB(NA)  *-2
00021  008E  4D10              INR     A5,0,/10        SEND INR INSTRUCTION TO DEVICE
00022  0090  5C04              RB(NA)  *-2             AND WAIT FOR CHAR TO BE INPUT
00023  0092  4290              CIO     A2,0,/10        STOP INPUT FUNCTION
00024  0094  4AD0              SST     A2,/10          AND GET STATUS
00025  0096  5C04              RB(NA)  *-2
00026  0098  0204              LDK     A2,4            LOAD PARAMETERS FOR OUTPUT CHARACTER
00027                  *                               WITH EVEN PARITY
00028  009A  42D0              CIO     A2,1,/10        SEND START COMMAND TO DEVICE
00029  009C  5C04              RB(NA)  *-2
00030  009E  4510              OTR     A5,0,/10        SEND CHAR BACK TO DEVICE
00031  00A0  5C04              RB(NA)  *-2
00032  00A2  4290              CIO     A2,0,/10        SEND STOP OUTPUT FUNCTION COMMAND
00033  00A4  4AD0              SST     A2,/10          AND GET STATUS
00034  00A6  5C04              RB(NA)  *-2
00035  00A8  5F22              RB      START+4         RETURN TO START OF INPUT ROUTINE
00036                          END     START
```

ECHO

```
00000                          IDENT    ECHO
00001          0000    BEGIN   EQU      *
00002                          RORG     BEGIN+/80
00003                  *
00004                  *       THIS PROGRAM ENABLES THE ENGINEER TO VERIFY THE INPUT/OUTPUT
00005                  *       FUNCTIONS OF THE OPERATOR'S PERIPHERAL IN ECHO MODE
00006                  *
00007                  *
00008                  *       LOAD THE PROGRAM START ADDRESS INTO A0 FROM THE
00009                  *       CONTROL PANEL - THEN PUSH THE RUN BUTTON
00010                  *
00011                  *
00012  0080  FFFF              DATA     /FFFF
00013  0082  0000              DATA     0
00014  0084  207F      START   HLT
00015  0086  20BF              INH
00016                  *
00017  0088  0201      INCHAR  LDK      A2,1                LOAD PARAMETERS FOR INPUT CHARACTER
00018                  *                                                              WITH NO
00019  008A  42D0              CIO      A2,1,/10            SEND START COMMAND TO DEVICE
00020  008C  5C04              RB(NA)   *+2
00021  008E  4D10              INR      A5,0,/10            SEND INR INSTRUCTION TO DEVICE
00022  0090  5C04              RB(NA)   *+2                 AND WAIT FOR CHAR TO BE INPUT
00023  0092  4290              CIO      A2,0,/10            STOP INPUT FUNCTION
00024  0094  4A00              SST      A2,/10              AND GET STATUS
00025  0096  5C04              RB(NA)   *+2
00026  0098  5F12              RB       INCHAR
00027                          END      START
```

COPY

```
00000                          IDENT    COPY
00001                  *
00002                  *       THIS PROGAM  READS A PAPER TAPE AND C PUNCHES A COPY
00003                  *
00004          0000    BEGIN   EQU      *
00005                          RORG     BEGIN+/80
00006  0080    207F    START   HLT
00007  0082    0101            LDK      A1,1
00008  0084    0200            LDK      A2,0
00009  0086    41E0    READ    CIO      A1,1,/20
00010  0088    5C04            RB(NA)   *-2
00011  008A    4D20            INR      A5,0,/20
00012  008C    5C04            RB(NA)   *-2
00013  008E    41A0            CIO      A1,0,/20
00014  0090    4CE0            SST      A4,/20
00015  0092    5C04            RB(NA)   *-2
00016  0094    2420            ANK      A4,/20
00017  0096    EC20            CWK      A4,/20
       0098    0020
00018  009A    581C            RB(E)    START
00019  009C    42F0            CIO      A2,1,/30
00020  009E    5C04            RB(NA)   *-2
00021  00A0    4530            OTR      A5,0,/30
00022  00A2    5C04            RB(NA)   *-2
00023  00A4    42B0            CIO      A2,0,/30
00024  00A6    4BF0            SST      A3,/30
00025  00A8    5C04            RB(NA)   *-2
00026  00AA    5F26            RB       READ
00027                          END      START
```

DUMP

```
00000                          IDENT    DUMP
00001          0000    BEGIN   EQU      *
00002                          RORG     BEGIN+/80
00003                  *
00004                  *       THIS PROGRAM ENABLES THE ENGINEER TO DUMP A SELECTED
00005                  *       AREA OF MEMORY ON EITHER THE ASR OR PER 3100
00006                  *
00007                  *
00008                  *       LOAD THE STARTING ADDRESS OF THE AREA TO BE
00009                  *       DUMPED INTO REGISTER A7 FROM THE CONTROL PANEL
00010                  *
00011                  *       LOAD THE ENDING ADDRESS OF THE AREA INTO
00012                  *       REGISTER A8 FROM THE CONTROL PANEL
00013                  *
00014                  *       LOAD THE PROGRAM STARTING ADDRESS INTO
00015                  *       REGISTER A0 FROM THE CONTROL PANEL THEN
00016                  *       PUSH THE RUN BUTTON
00017                  *
00018                  *
00019  0080    FFFF            DATA     /FFFF
00020  0082    0000            DATA     0
00021  0084    207F    START   HLT
00022  0086    20BF            INH                          THE PROGRAM RUNS IN INHIBIT MODE
00023                  *
00024  0088    0404            LDK      A4,4                LOAD PARAMETERS FOR OUTPUT CHARACTER
00025                  *                                    WITH EVEN PARITY
00026  008A    44D0            CIO      A4,1,/10            SEND START COMMAND TO DEVICE
00027  008C    813C    WORD    LDR*     A1,A7               LOAD THE CONTENTS OF THE MEMORY ADDRESS
00028                  *                                    CONTAINED IN A7 INTO A1
00029  008E    0204            LDK      A2,4                LOAD CHARACTER COUNT FOR WORD
00030  0090    060F    CONT    LDK      A6,/F               LOAD MASK PATTERN
00031  0092    A604            ANR      A6,A1               FIND THE PRINTABLE CHAR ADDRESS IN TABLE
00032  0094    E558            LC       A5,TABLE,A6         AND LOAD INTO REGISTER A5
       0096    00C0  R
00033  0098    E549            SC       A5,BUFF+1,A2        THEN STORE IN BUFF
       009A    00D1  R
00034  009C    39E4            SRC      A1,4                SHIFT TO ISOLATE NEXT CHAR
00035  009E    1A01            SUK      A2,1                SUBTRACT ONE FROM CHAR COUNT
00036  00A0    5C12            RB(NZ)   CONT                AND BRANCH BACK IF NOT LAST CHAR IN WORD
00037  00A2    E348    OUCHAR  LC       A3,BUFF,A2          LOAD CHAR FROM BUFF INTO A3
       00A4    00D0  R
00038  00A6    4310            OTR      A3,0,/10            AND SEND TO THE DEVICE
00039  00A8    5C04            RB(NA)   *+2
00040  00AA    1201            ADK      A2,1                UPDATE BUFF CHAR COUNT
00041  00AC    EA20            CWK      A2,6                AND CHECK FOR LAST CHAR SENT
       00AE    0006
00042  00B0    5C10            RB(NE)   OUCHAR              AND BRANCH BACK IF NOT LAST CHAR
00043  00B2    1702            ADK      A7,2                UPDATE MEMORY ADDRESS COUNTER
00044  00B4    EF02            CWR      A7,A8               AND CHECK FOR LAST ADDRESS DUMPED
00045  00B6    5D2C            RB(NG)   WORD                IF NO  LAST OUTPUT NEW WORD
```

DUMP

```
00046                          *                                    IF LAST ADDRESS THEN
00047  00B8  4490                      CIO       A4,0,/10          SEND STOP COMMAND TO THE DEVICE
00048  00BA  4CD0                      SST       A4,/10            AND GET STATUS
00049  00BC  5C04                      RB(NA)    *+2
00050  00BE  5F3C                      RB        START             DUMP FINISHED GO TO START OF PROGRAM
00051  00C0  3031              TABLE   DATA      '0123456789ABCDEF'
       00C2  3233
       00C4  3435
       00C6  3637
       00C8  3839
       00CA  4142
       00CC  4344
       00CE  4546
00052  00D0  0A0D              BUFF    DATA      /0A0D
00053                                  END       START

          SYMBOL TABLE

BEGIN   0000  R   BUFF    00D0  R   CONT    0090  R   OUCHAR  00A2  R
START   0084  R   TABLE   00C0  R   WORD    008C  R

          ASS.ERR.    00000
:EOF
PROG ELAPSED TIME:  00H-00M-00S-000MS-
```

6.36 PRODUCING PROGRAMS ON PAPER TAPE

There are several methods of producing programs on paper tape either from this section, from the sections in Volume 2, or programs you have written yourself. For short programs that can be loaded into memory by hand using the Control Panel switches the program PUNCH in Volume 2 can be used and provided that the number of characters in the program does not exceed 80 it can be input using the IPL routine; for programs with more than 80 characters the MINI-IPL will have to be included before your program.

6.37 Using the ASR Punch in LOCAL Mode

If the ASR is fitted with a punch the hexadecimal characters of the program can be input from the keyboard provided the following rules are observed:

- The LOCAL/REMOTE switch must be set to LOCAL and the punch must be switched ON.
- The punch will treat any input from the keyboard as data and will reproduce the appropriate code on tape so if you make an error while typing in the instructions there is no way it can be corrected and you will have to begin again.
- To produce the tape leader and the trailing end of tape press the HERE IS key three times.
- The number keys 0 to 9 can be used normally but the letter keys A B C D E and F MUST NOT be used, instead use the J K L M N and O keys respectively which will produce the correct hexadecimal code on the tape.
- If the number of characters exceeds the line length the punch must be switched OFF before you press the CR and LF keys, and it must be switched ON again before you continue typing in the program instructions.
- The resulting program tape is in 8-bit ASCII but the Bootstrap can only decode Object code (either 8+8 or 4x4) so to load the program tape using the IPL routine the Control Panel switches must first be set for 4x4 mode (i.e. switches 0, 3 and 10 set to value 1). The program will then be loaded from the PTR.

6.38 If the program produced is less than 80 8-bit characters the Bootstrap will keep on reading (looking for the missing characters) until all the tape has passed through the read head. This does not affect the instructions that have been loaded but you must stop the CPU by pushing the INST button, then push the MC button to reset the device and CU logic. Any parameters can now be loaded into registers and the program will run normally.

6.39 Program ASC4x4

This program allows you to write programs using the operator's peripheral. The hexadecimal characters of the program can be entered in a similar way to that described above using the number keys 0 to 9 and letter keys J to O. However, the program is more flexible in that erroneous characters can be ignored and certain other characters are recognised which make the production of a program tape much easier. Once the program has been loaded put the start address of the program into register A0 and push the RUN button. The program will run and in addition to the keys mentioned above the following keys are significant:

- ↑ or ^ — The symbol and position of this key will depend on the device being used but its function and 7-bit code (/5E) are the same. The use of this key is to delete a character typed in error. When the program recognises this character it downdates the buffer counter and branches back to the input routine to accept a new character.
- Space Bar — The Space Bar is used to tell the program that the four hexacedimal characters of the program instruction are valid and that any corrections have been made. When the program recognises this character code (/20) it branches to the translate routine and stores them in the output buffer; it then branches back to the input routine for four new characters.
- # — This key is used to indicate that the last program character has been typed. When the program recognises this character code (/23) it branches to the CR/LF routine.

6.40 The program translates the valid characters received into 8-bit ASCII format and store them in a buffer until 40 valid characters have been received or the # character has been recognised. It then sends the CR and LF characters to the Operator's device and starts the punch routine. If the last character code has not been recognised the program branches back to the input routine to accept new characters; if it has been recognised it branches back to the start of program and stops.

6.41 The program has 74 instructions and in the program listing the start address is shown as 00AA. This is so that if your only means of producing program tapes is by one of the examples given in this section you can include the MINI-IPL in front of the program and load it using the 4x4 IPL routine. If you have access to an Assembler and Linkage Editor you will have no trouble in producing a master tape of this program, but if you have to load it by hand from the Control Panel switches before reproducing it by one of the methods suggested in this section great care must be taken that you do not introduce errors into the program whilst it is being loaded.

6.42 Program HEXTAP

This program operates exactly the same way as the program ASC4x4 and all instructions regarding the use of that program apply to this program. The only differences are that this program operates in Echo mode and the output tape is punched in 8+8 hexadecimal format. If you include the MINI-IPL in front of the program it can then be loaded from the PTR using the normal IPL routine.

ASC4X4

```
00000                                IDENT    ASC4X4
00001                          *
00002                          *     THIS PROGRAM ALLOWS ENGINEERS TO WRITE PROGRAMS ON THE OPERATOR'S
00003                          *     PERIPHERAL AND THEN PUNCHES THE PROGRAM IN ASCII FOR 4X4 LOADING
00004                          *
00005          0000      BEGIN  EQU      *
00006                           RORG     BEGIN+/AA
00007                          *
00008                          *     THIS ROUTINE ACCEPTS CHARACTERS FROM THE
00009                          *     OPERATOR'S PERIPHERAL IN THE NON ECHO MODE
00010                          *     AND IS SIMILAR TO THE PROGRAM NOECHO
00011                          *
00012  00AA    2840      START  ENB
00013  00AC    207F             HLT
00014  00AE    20BF             INH
00015  00B0    0100      OUTCT  LDK      A1,0                  OUTPUT BUFFER COUNTER ZEROISED
00016  00B2    0300      INCT   LDK      A3,0                  ZEROISE INPUT BUFFER COUNTER
00017  00B4    0601      INPT   LDK      A6,1
00018  00B6    46D0             CIO      A6,1,/10
00019  00B8    5C04             RB(4)    *-2
00020  00BA    4D10             INR      A5,0,/10
00021  00BC    5C04             RB(4)    *-2
00022  00BE    4690             CIO      A6,0,/10
00023  00C0    4CD0             SST      A4,/10
00024  00C2    5C04             RB(4)    *-2
00025  00C4    0600             LDK      A6,0
00026  00C6    46D0             CIO      A6,1,/10
00027  00C8    5C04             RB(4)    *-2
00028  00CA    4510             OTR      A5,0,/10
00029  00CC    5C04             RB(4)    *-2
00030  00CE    4690             CIO      A6,0,/10
00031  00D0    4CD0             SST      A4,/10
00032  00D2    5C04             RB(4)    * 2
00033  00D4    257F             ANK      A5,/7F                MODIFIES CHARACTER TO 7 DATA BITS
00034                          *
00035                          *     THIS ROUTINE CHECKS FOR  ERRONEOUS CHARS AND END OF PROGRAM
00036                          *
00037  00D6    ED20             CWK      A5,/5E                CHECK IF LAST CHAR WAS TYPED IN ERROR
       00D8    005E
00038  00DA    5404             RF(NE)   SPCE
00039  00DC    1B01             SUK      A3,1                  THE LAST CHAR WAS TYPED IN ERROR SO
00040  00DE    5F2C             RB       INPT                  DOWNDATE CHAR COUNTER AND READ NEXT CHAR
00041  00E0    ED20      SPCE   CWK      A5,/20                CHECK FOR SPACE BAR CHARACTER
       00E2    0020
00042  00E4    500E             RF(E)    TRAN                  GO TO TRAN IF 4 VALID CHARS INPUT
00043  00E6    ED20             CWK      A5,/23                CHECK FOR END OF PROGRAM   #
       00E8    0023
00044  00EA    5038             RF(E)    CRLF
00045                          *
00046                          *     THIS ROUTINE STORES INPUT CHARS IN TEMPORARY BUFFER
```

```
00047                          *
00048  00EC  E54D              SC      A5,TEMBUF,A3          STORE CHAR IN TEMBUF
       00EE  015A  R
00049  00F0  1301              ADK     A3,1                  UPDATE INPUT BUFFER COUNT
00050  00F2  5F40              RB      INPT                  GO TO INPUT ROUTINE
00051                          *
00052                          *       THIS ROUTINE TRANSLATES CHARS INTO ASCII FORMAT
00053                          *
00054  00F4  0300              TRAN    LDK     A3,0          ZEROISE TEMBUF COUNTER
00055  00F6  B208              TRAN1   XRR     A2,A2         CLEAR REGISTER
00056  00F8  E54C              LC      A5,TEMBUF,A3          GET CHAR FROM TEMBUF
       00FA  015A  R
00057  00FC  8214              LDR     A2,A5                 SAVE CHAR
00058  00FE  250F              ANK     A5,/0F                SAVE RIGHT MOST BITS
00059  0100  2240              ANK     A2,/40
00060  0102  EA20              CWK     A2,/40                CHECK IF CHAR IS DECIMAL
       0104  0040
00061  0106  5404              RF(NE)  DECIM
00062  0108  15C0              ADK     A5,/C0                ALPHA CHAR
00063  010A  5702              RF      STORE
00064  010C  15B0              DECIM   ADK     A5,/B0        DECIMAL CHAR
00065                          *
00066                          *       THIS ROUTINE STORES THE PROGRAM CHARS IN BUFFER
00067                          *       AND SENDS CR LF AFTER 40 VALID CHARS HAVE BEEN RECIEVED
00068                          *
00069  010E  E545              STORE   SC      A5,BUFF,A1    STORE VALID CHAR
       0110  0160  R
00070  0112  1101              ADK     A1,1                  UPDATE CHAR COUNTER
00071  0114  1301              ADK     A3,1                  UPDATE TEMBUF COUNT
00072  0116  EB20              CWK     A3,4                  CHECK IF ALL 4 CHARS HAVE BEEN TRANSLATE
       0118  0004
00073  011A  5C26              RB(NE)  TRAN1                 TRANSLATE NEW CHARACTER IF NOT ZERO
00074  011C  E920              CWK     A1,40                 COMPARE FOR BUFFER FULL
       011E  0028
00075  0120  5002              RF(E)   CRLP                  IF YES SEND CR/LF CHARS THEN PUNCH BUFFE
00076  0122  5F72              RB      INCT                  IF ZERO GO TO INPUT ROUTINE FOR NEW CHAR
00077  0124  070A              CRLP    LDK     A7,/0A        LOAD LF CHAR
00078  0126  46D0              CIO     A6,1,/10
00079  0128  4710              OTR     A7,0,/10              AND SEND TO TYPEWRITER
00080  012A  5C04              RB(4)   *-2
00081  012C  070D              LDK     A7,/0D                LOAD CR CHAR
00082  012E  4710              OTR     A7,0,/10              AND SEND TO TYPEWRITER
00083  0130  5C04              RB(4)   *-2
00084  0132  4690              CIO     A6,0,/10
00085  0134  4CD0              SST     A4,/10
00086  0136  5C04              RB(4)   *-2
00087                          *
00088                          *       THIS ROUTINE PUNCHES THE ASCII PROGRAM ON TAPE
00089                          *
00090  0138  0200              LDK     A2,0                  ZEROISE  BUFF COUNTER
00091  013A  46F0              CIO     A6,1,/30              START PUNCH
00092  013C  5C04              RB(4)   *-2
00093  013E  E348              CONT    LC      A3,BUFF,A2    LOAD CHAR FROM BUFF
       0140  0160  R
00094  0142  4330              OTR     A3,0,/30              AND PUNCH
00095  0144  5C04              RB(4)   *-2
00096  0146  1201              ADK     A2,1                  UPDATE BUFF COUNTER
00097  0148  1901              SUK     A1,1                  SUBTRACT 1 FROM CHAR COUNT
```

```
00098  014A  5C0E              RB(4)     CONT                AND CHECK FOR LAST CHAR
00099  014C  46B0              CIO       A6,0,/30            THEN STOP PUNCH
00100  014E  4CF0              SST       A4,/30
00101  0150  5C04              RB(NA)    *-2
00102  0152  ED20              CWK       A5,/23              CHECK FOR LAST PROGRAM CHAR
       0154  0023
00103  0156  5CA8              RB(4)     OUTCT               NO?  ACCEPT MORE CHARACTERS
00104  0158  5FB0              RB        START
00105  015A            TEMBUF  RES       3                   TEMPORARY BUFFER
00106  0160            BUFF    RES       25
00107                          END       START

          SYMBOL TABLE

BEGIN   0000  R   BUFF    0160  R   CONT    013E  R   CRLP    0124  R
DECIM   010C  R   INCT    00B2  R   INPT    00B4  R   OUTCT   00B0  R
SPCE    00E0  R   START   00AA  R   STORE   010E  R   TEMBUF  015A  R
TRAN    00F4  R   TRAN1   00F6  R

          ASS.ERR.     00000
:EOF
PROG ELAPSED TIME:  00H-00M-00S-000MS-
```

HEXTAP

```
00000                                IDENT    HEXTAP
00001                          *
00002                          *     THIS PROGRAM ALLOWS ENGINEERS TO WRITE PROGRAMS ON THE OPERATOR'S
00003                          *     PERIPHERAL AND PUNCHES THE PROGRAM TAPE IN HEXADECIMAL 8+8 FORMAT
00004                          *
00005          0000            BEGIN    EQU      *
00006                                   RORG     BEGIN+/AA
00007                          *
00008                          *     THHIS ROUTINE ACCEPTS CHARACTERS FROM THE OPERATOR'S PERIPHERAL
00009                          *     IN ECHO MODE AND IS SIMILAR TO THE PROGRAM ECHO
00010                          *
00011  00AA  2840              START    ENB
00012  00AC  207F                       HLT
00013  00AE  20BF                       INH
00014  00B0  0100              OUTCT    LDK      A1,0                  OUTPUT BUFFER COUNTER ZEROISED
00015  00B2  0300              INCT     LDK      A3,0                  ZEROISE INPUT BUFFER COUNTER
00016  00B4  0621              INPT     LDK      A6,/21
00017  00B6  46D0                       CIO      A6,1,/10
00018  00B8  5C04                       RB(4)    *-2
00019  00BA  4D10                       INR      A5,0,/10
00020  00BC  5C04                       RB(4)    *-2
00021  00BE  4690                       CIO      A6,0,/10
00022  00C0  4CD0                       SST      A4,/10
00023  00C2  5C04                       RB(4)    *-2
00024  00C4  257F                       ANK      A5,/7F                MODIFIES CHARACTER TO 7 DATA BITS
00025                          *
00026                          *     THIS ROUTINE CHECKS FOR  ERRONEOUS CHARS AND END OF PROGRAM
00027                          *
00028  00C6  ED20                       CWK      A5,/5E                CHECK IF LAST CHAR WAS TYPED IN ERROR
       00C8  005E
00029  00CA  5404                       RF(NE)   SPCE
00030  00CC  1B01                       SUK      A3,1                  THE LAST CHAR WAS TYPED IN ERROR SO
00031  00CE  5F1C                       RB       INPT                  DOWNDATE CHAR COUNTER AND READ NEXT CHAR
00032  00D0  ED20              SPCE     CWK      A5,/20                CHECK FOR SPACE BAR CHARACTER
       00D2  0020
00033  00D4  500E                       RF(E)    TRAN                  GO TO TRAN IF 4 VALID CHARS INPUT
00034  00D6  ED20                       CWK      A5,/23                CHECK FOR END OF PROGRAM   #
       00D8  0023
00035  00DA  502C                       RF(E)    CRLP
00036                          *
00037                          *     THIS ROUTINE STORES INPUT CHARS IN TEMPORARY BUFFER
00038                          *
00039  00DC  E54D                       SC       A5,TEMBUF,A3          STORE CHAR IN TEMBUF
       00DE  0140  R
00040  00E0  1301                       ADK      A3,1                  UPDATE INPUT BUFFER COUNT
00041  00E2  5F30                       RB       INPT                  GO TO INPUT ROUTINE
00042                          *
00043                          *     THIS ROUTINE TRANSLATES CHARS INTO ASCII FORMAT
00044                          *
00045  00E4  0300              TRAN     LDK      A3,0                  ZEROISE TEMBUF COUNTER
00046  00E6  0404                       LDK      A4,4                  LOAD SHIFT COUNTER
```

```
00047  00E8  B208      TRAN1   XRR     A2,A2                   CLEAR REGISTER
00048  00EA  3A44              SLL     A2,4                    SHIFT CHARS 4 PLACES LEFT
00049  00EC  E54C              LC      A5,TEMBUF,A3            GET CHAR FROM TEMBUF
       00EE  0140  R
00050  00F0  250F              ANK     A5,/0F                  SAVE RIGHT MOST BITS
00051  00F2  9214              ADR     A2,A5                   ADD 4 BIT CHAR FROM A5 INTO A2
00052  00F4  1301              ADK     A3,1                    UPDATE TEMBUF COUNTER
00053  00F6  1C01              SUK     A4,1                    DOWNDATE SHIFT COUNTER AND IF NOT ZERO
00054  00F8  5C10              RB(NZ)  TRAN1+2                 BRANCH AND TRANSLATE NEW CHAR
00055                  *
00056                  *       THIS ROUTINE STORES THE PROGRAM CHARS IN BUFFER
00057                  *       AND SENDS CR LF AFTER 40 VALID CHARS HAVE BEEN RECIEVED
00058                  *
00059  00FA  8245      STORE   ST      A2,BUFF,A1              STORE TWO 8 BIT HEXADECIMAL CHARS IN BUF
       00FC  0146  R
00060  00FE  1102              ADK     A1,2                    UPDATE CHAR COUNTER
00061  0100  E920              CWK     A1,20                   COMPARE FOR BUFFER FULL
       0102  0014
00062  0104  5002              RF(E)   CRLP                    IF YES SEND CR/LF CHARS THEN PUNCH BUFFE
00063  0106  5F56              RB      INCT                    IF ZERO GO TO INPUT ROUTINE FOR NEW CHAR
00064  0108  070A      CRLP    LDK     A7,/0A                  LOAD LF CHAR
00065  010A  0600              LDK     A6,0                    LOAD START OUTPUT FOR DEVICE
00066  010C  46D0              CIO     A6,1,/10
00067  010E  4710              OTR     A7,0,/10                AND SEND TO TYPEWRITER
00068  0110  5C04              RB(4)   *-2
00069  0112  070D              LDK     A7,/0D                  LOAD CR CHAR
00070  0114  4710              OTR     A7,0,/10                AND SEND TO TYPEWRITER
00071  0116  5C04              RB(4)   *-2
00072  0118  4690              CIO     A6,0,/10
00073  011A  4CD0              SST     A4,/10
00074  011C  5C04              RB(4)   *-2
00075                  *
00076                  *       THIS ROUTINE PUNCHES THE ASCII PROGRAM ON TAPE
00077                  *
00078  011E  0200              LDK     A2,0                    ZEROISE  BUFF COUNTER
00079  0120  46F0              CIO     A6,1,/30                START PUNCH
00080  0122  5C04              RB(4)   *-2
00081  0124  E348      CONT    LC      A3,BUFF,A2              LOAD CHAR FROM BUFF
       0126  0146  R
00082  0128  4330              OTR     A3,0,/30                AND PUNCH
00083  012A  5C04              RB(4)   *-2
00084  012C  1201              ADK     A2,1                    UPDATE BUFF COUNTER
00085  012E  1901              SUK     A1,1                    SUBTRACT 1 FROM CHAR COUNT
00086  0130  5C0E              RB(4)   CONT                    AND CHECK FOR LAST CHAR
00087  0132  46B0              CIO     A6,0,/30                THEN STOP PUNCH
00088  0134  4CF0              SST     A4,/30
00089  0136  5C04              RB(NA)  *-2
00090  0138  ED20              CWK     A5,/23                  CHECK FOR LAST PROGRAM CHAR
       013A  0023
00091  013C  5C8E              RB(4)   OUTCT                   NO?  ACCEPT MORE CHARACTERS
00092  013E  5F96              RB      START
00093  0140          TEMBUF  RES     3                       TEMPORARY BUFFER
00094  0146          BUFF    RES     25
00095                          END     START
```

SYMBOL TABLE

```
BEGIN   0000  R   BUFF    0146  R   CONT    0124  R   CRLP    0108  R
INCT    00B2  R   INPT    00B4  R   OUTCT   00B0  R   SPCE    00D0  R
START   00AA  R   STORE   00FA  R   TEMBUF  0140  R   TRAN    00E4  R
TRAN1   00E8  R
```

6.43 SUGGESTIONS FOR SOPHISTICATED PROGRAMS

As a general rule sophisticated programs perform more than one function and are longer than 40 instructions (so cannot be loaded easily by hand). An easy way to produce these programs is to take routines from the simple programs (or routines you have written yourself) and link them together adding specific routines to perform the task you want. The previous two programs are examples of this technique. They comprise routines for the PTR and PTP (both given in Volume II) slightly modified and a linking routine that translates the character code from the Operator's device to either ASCII or Hexadecimal character code. The only precautions necessary for the two routines taken from Volume II was to make sure that when the same register is used for more than one function you do not corrupt data needed in another part of the program, that the buffer address is the same, that the displacement value of branch instructions is correct for the new program, and that all superfluous instructions are omitted. The program examples which follow have all been produced in this way and should enable you to produce your own programs capable of performing specific and more complicated tests even with a basic system.

6.44 Routine MINI-IPL

This routine allows you to load your program immediately after the routine itself (hexadecimal address 009A) making use of the Bootstrap instructions. You will remember that the Bootstrap is programmed to load 80 8-bit characters from the device before it branches to address 0084 so this routine is used to modify some of the instructions of the Bootstrap so that it will continue loading your program. The first two addresses of the routine only contain data (not used by the routine) and are needed to make sure that the first instruction of the routine is in address 84. The first two instructions of the routine change the displacement value of the AB instruction contained in the last address of the Bootstrap (7E) to the start address of the program, in this example address 9A. As 7E already contains 84 it is only necessary to add the number of locations needed by the MINI-IPL starting at address 86 and counting two for each location (because the Program counter only uses bits 0 to 14). We thus arrive at decimal 22 which when added to the contents of 7E will give the hexadecimal address 9A. The next four instructions (addresses 8A to 92) restore the CIO Start instruction in the Bootstrap locations. The LDKL instruction loads the number of characters that in turn the modified Bootstrap has to load. To calculate the number of characters you must count the total number of locations needed (using the same method described for the start address) beginning at location 80, then multiply by two to give the number of characters. Now subtract 80 (which is the number of characters loaded by the unmodified Bootstrap) and the remainder is the number you need to load in register A5. You must make sure that the count is accurate otherwise the modified Bootstrap will either not load all the instruction (count too short) or will load the blank tape of the trailing end as zeros. In the latter case if only one or two such characters are read it should not affect the program but if the End of Tape is detected then it will almost certainly cause the program to abort. The last instruction of the routine branches back to location 48 of the Bootstrap which is the start of the read routine for paper tape. This routine can be modified to suit your own requirements, for example the start address of the program can be changed to any location in memory greater than 9A. For locations between 9A and D0 care will have to be taken that the branch back to Bootstrap instruction is ignored otherwise the principle remains the same. The start address of the program is calculated in the same way and stored in location 7E and unless it is greater than can be stored in a short constant instruction the same instructions can be used. If it is greater than decimal address 255 than you will have to construct an ABL instruction with the start address and use the same instructions as described in the following routine for using the Interrupt Button. The routine for calculating the character count for register A5 remains the same but you will have to calculate a load address for the first program instruction address. When the Bootstrap branches to address 84 the contents of register A6 will be decimal 208 (hexadecimal D0) so you will have to either add or subtract from this number to give the correct start address for loading the program into register A6, inserting an LDKL instruction for this register after the LDKL instruction for A5. If this start address is greater than D0 you must fill the remaining locations (after the LDKL which loads register A6) with zeros up to address CC then put the AB/48 instruction into location CE. If the start address is less than D0 but greater than 9A the locations from 9A to the location before the start address must be filled with zeros, and you must remember that the program must bypass the AB/48 instruction otherwise the program may abort. As long as you take extreem care when modifying this routine you should have no problems.

MINI IPL ROUTINE

```
00006                       *
00007                       *       THIS IS A MINI IPL FOR LOADING PAPER TAPEIN 4X4 FORMAT ASCII
00008                       *
00009  0080  FFFF                   DATA    /FFFF
00010  0082  0000                   DATA    0000
00011  0084  0516           XYZ     LDK     A5,22               THESE TWO INSTRUCTIONS PUT
00012  0086  9541                   ADS     A5,/7E              THE PROGRAM START ADDRESS IN 7E
       0088  007E
00013  008A  871E                   LDR     A7,A15              GET KEY VALUE FROM A15
00014  008C  273F                   ANK     A7,/3F              SAVE CU ADDRESS
00015  008E  9720                   ADKL    A7,/41C0            ADD CIO START INSTRUCTION FORMAT
       0090  41C0
00016  0092  872D                   STR     A7,A3               STORE CIO START IN A3
00017  0094  8520                   LDKL    A5,8                LOAD CHAR COUNT
       0096  0008
00018  0098  0F48                   AB      /48                 RETURN TO BOOTSTRAP
```

6.45 Routine for Using the INT Button

This is a routine which can be used for a number of different things with only minor modifications. It was written so that either the INT button or BREAK key on the Operator's device can be used to start a program instead of the RUN button, and enables programs to be run with a Mini-Panel. The first two instructions construct an RIT instruction and store it in hexadecimal address E, and the last two instructions construct an AB instruction (to the start address of the program) and store in the address following the RIT which is hexadecimal 10. These addresses were used because of the interrupt level to which the INT button had been connected (all interrupts both internal and external are connected to an interrupt level which corresponds to a memory address). In the example the INT button had been connected to interrupt level 7 so because only MAD lines 0 to 14 are used for memory addressing (MAD15 is only used for character addressing) the address corresponding to interrupt level 7 will be hexadecimal E. The same routine can be used with slight modifications to check the operation of all the interrupts both internal and external. For internal interrupts the construction of the RIT instruction must be changed to give the correct 5-bit hexadecimal constant for the interrupt being checked, and the address where it is stored must be changed to correspond to the address of the interrupt level. The branch instruction can remain the same but it must be stored in the address following the RIT instruction. For external interrupts no RIT instruction is needed, but the branch instruction must be stored in the appropriate address.

6.46 Program CPY4x4

This program is an example of how to link simple routines and simple programs to produce a sophisticated program. It combines the MINI-IPL routine, the INT button routine and the program COPY. You will see from the listing that no change was needed for either the MINI-IPL or INT routines, and for the COPY program the only changes were to include an Enable Interrupt instruction (ENB) as the first instruction and to change the Halt instruction (HLT) to an RB * instruction which is a perpetual branch back to itself until overridden by something else in the program such as the INT button being pushed. In effect this RB instruction acts as a HLT instruction and allows you to load the tape you want copied onto the PTR. Then when you push the INT button the INT routine branches to the start address of program and your tape is read by the PTR and a copy punched by the PTP. The program can be run in the same way with either a Full Control Panel or a Mini-Panel.

INT BUTTON ROUTINE

```
00020                    *
00021                    *         THIS ROUTINE LOADS THE INT BUTTON PARAMETERS
00022                    *
00023                    *
00024  009A  8120       START     LDKL      A1,/20DF
       009C  20DF
00025  009E  8141                 ST        A1,/E
       00A0  000E
00026  00A2  8120                 LDKL      A1,/0FAE
       00A4  0FAE
00027  00A6  8141                 ST        A1,/10
       00A8  0010
```

```
00000                                IDENT    CPY4X4
00001                          *
00002                          *     THIS PROGRAM READS A PAPER TAPE AND PUNCHES A COPY
00003                          *
00004          0000            BEGIN EQU      *
00005                                RORG     BEGIN+/80
00006                          *
00007                          *     THIS IS A MINI IPL FOR LOADING PAPER TAPEIN 4X4 FORMAT ASCII
00008                          *
00009  0080    FFFF                  DATA     /FFFF
00010  0082    0000                  DATA     0000
00011  0084    0516            XYZ   LDK      A5,22              THESE TWO INSTRUCTIONS PUT
00012  0086    9541                  ADS      A5,/7E             THE PROGRAM START ADDRESS IN 7E
       0088    007E
00013  008A    871E                  LDR      A7,A15             GET KEY VALUE FROM A15
00014  008C    273F                  ANK      A7,/3F             SAVE CU ADDRESS
00015  008E    9720                  ADKL     A7,/41C0           ADD CIO START INSTRUCTION FORMAT
       0090    41C0
00016  0092    872D                  STR      A7,A3              STORE CIO START IN A3
00017  0094    8520                  LDKL     A5,8               LOAD CHAR COUNT
       0096    0008
00018  0098    0F48                  AB       /48                RETURN TO BOOTSTRAP
00019                          *
00020                          *
00021                          *     THIS ROUTINE LOADS THE INT BUTTON PARAMETERS
00022                          *
00023                          *
00024  009A    8120            START LDKL     A1,/20DF
       009C    20DF
00025  009E    8141                  ST       A1,/E
       00A0    000E
00026  00A2    8120                  LDKL     A1,/0FAE
       00A4    0FAE
00027  00A6    8141                  ST       A1,/10
       00A8    0010
00028                          *
00029                          *
00030                          *     THE PROGRAM START HERE
00031                          *
00032                          *
00033  00AA    2840                  ENB
00034  00AC    5F02            WAIT  RB       *
00035  00AE    0101                  LDK      A1,1
00036  00B0    0200                  LDK      A2,0
00037  00B2    41E0            READ  CIO      A1,1,/20
00038  00B4    5C04                  RB(NA)   *-2
00039  00B6    4D20                  INR      A5,0,/20
00040  00B8    5C04                  RB(NA)   *-2
00041  00BA    41A0                  CIO      A1,0,/20
00042  00BC    4CE0                  SST      A4,/20
```

```
CPY4x4
00043  00BF  5C04              RB(NA)   *-2
00044  00C0  2420              ANK      A4,/20
00045  00C2  EC20              CWK      A4,/20
       00C4  0020
00046  00C6  581E              RB(E)    WAIT-2
00047  00C8  42F0              CIO      A2,1,/30
00048  00CA  5C04              RB(NA)   *-2
00049  00CC  4530              OTR      A5,0,/30
00050  00CE  5C04              RB(NA)   *-2
00051  00D0  42B0              CIO      A2,0,/30
00052  00D2  4BF0              SST      A3,/30
00053  00D4  5C04              RB(NA)   *-2
00054  00D6  5F26              RB       READ
00055                          END      XYZ

          SYMBOL TABLE

BEGIN   0000  R   READ    00B2  R   START   009A  R   WAIT    00AC  R
XYZ     0084  R

          ASS,ERR.    00000
$EOF
PROG ELAPSED TIME:  00H-00M-00S-000MS-
```

6.47 Program MEM51

This is another example of how to link several routines to the simple program MEMHAN to produce a sophisticated program that will carry out a check of the entire memory without intervention from the engineer unless there is a detected error. It is more of a confidence program (after a memory fault has been repaired) than a specific test of a selected area whilst trying to locate a fault. It is loaded with a standard IPL, starts automatically, and it can be run with either a Full Control Panel or a Mini-Panel. The next routine computes the total memory size. Starting with the highest address (32K) it checks if it is possible to do a write instruction. If no it subtracts 0.5K from the address stored in register A9 and tries again (0.5K is the minimum size by which the memory size can be changed). If no again it continues in this loop until it finds an address where it is possible to do the do the write instruction. When an address has been found the program now knows the last address of memory and stores this information in a buffer called MEMADD. It also stores the start address of where the program can start writing the test patterns (this address is already known because of the size of the program). A second buffer called PATEN holds the four test patterns; these can be changed to suit individual needs. A software timer is also included so that the Write/Read to memory routine can be repeated a specific number of times (in this listing 25). When the starting and end address for the memory routine have been stored by the PARAMS routine it branches to the start address of this routine and starts the first Write/Read operation. The Write/Read routine writes and check reads each pattern once then checks the timer count to see if more Write/Read operations are needed. When the timer count is zero the Write/Read routine if left and the first of the transfer routines is entered to prepare the parameters for when the program is transferred to the top of memory. These routines compute the new start and end addresses for the Write/Read operations, the new addresses for the three buffers, and the new area where the program will be loaded. When all the addresses have been computed the transfer routine transfers the program to the top of memory, and when the last instruction has been loaded it starts the Write/Read routine which then tests the rest of memory from location 0 to wherever the new end address is located. When both parts of the memory have been tested the program stops with a Halt instruction. If an error is detected during the Write/Read operations the program will stop on an HLT instruction, if you wish the program to continue just push the RUN button. With a Full Control Panel you can check the address in error using the same instructions given for MEMHAN, with the Mini-Panel there is no way to display this information.

MEM51

```
00000                          IDENT    MEM51
00001                    *
00002                    *     THIS PROGRAM TESTS THE ENTIRE MEMORY
00003                    *
00004    0000          START    EQU      *
00005                    *
00006                    *     THIS ROUTINE COMPUTES THE MEMORY SIZE
00007                    *
00008  0000  8720               LDKL     A7,/5555            TEST PATTERN
       0002  5555
00009  0004  81A0               LDKL     A9,/8000            LOAD MAXIMUM MEMORY SIZE
       0006  8000
00010  0008  8727       COMP    STR      A7,A9               TRY TO STORE PATTERN
00011  000A  FF26               CWR*     A7,A9               CHECK IF LOCATION EXISTS
00012  000C  5006               RF(0)    PARAMS              YES? GO TO PROGRAM
00013  000E  99A0               SUKL     A9,/200             NO SUBTRACT 0,5K
       0010  0200
00014  0012  5F0C               RB       COMP                AND TRY AGAIN
00015                    *
00016                    *     THIS ROUTINE LOADS THE START AND END ADDRESSES
00017                    *
00018  0014  81C1       PARAMS  ST       A9,MEMADD           STORE END ADDRESS
       0016  0022  R
00019  0018  81A0               LDKL     A9,/100             LOAD START ADDRESS FOR WRITE/READ ROUTIN
       001A  0100
00020  001C  81C1               ST       A9,MEMADD+2         AND STORE IN MEMADD BUFFER
       001E  0024  R
00021  0020  570E               RF(7)    LOAD                GO TO STRAT OF WRITE/READ ROUTINE
00022                    *
00023                    *
00024  0022          MEMADD   RES      2
00025                    *
00026  0026  5555       PATEN   DATA     /5555               TEST
00027  0028  AAAA               DATA     /AAAA               PATTERNS
00028  002A  FFFF               DATA     /FFFF               FOR WRITE
00029  002C  0000               DATA     /0000               READ
00030                    *
00031  002E  0019       TIMER   DATA     25                  SOFTWARE TIMER COUNT
00032                    *
00033                    *     THESE INSTRUCTIONS PREPARE THE PROGRAM PARAMETERS
00034                    *
00035  0030  0700       LOAD    LDK      A7,0                ZEROISE REPEAT COUNTER
00036  0032  0200       LOAD1   LDK      A2,0                ZEROISE PATTERN COUNTER
00037  0034  85C8       LDPAT   LD       A13,PATEN,A2        LOAD TEST PATTERN
       0036  0026  R
00038                    *
00039                    *     THIS IS THE WRITE MEMORY ROUTINE
00040                    *
00041  0038  81C0       TEST    LD       A9,MEMADD+2         LOAD START ADDRESS
       003A  0024  R
00042  003C  85A7       WRITE   STR      A13,A9              WRITE TEST PATTERN IN MEMORY
```

```
00043  003E  91A0          ADKL    A9,2                UPDATE MEMORY ADDRESS
       0040  0002
00044  0042  E9C0          CW      A9,MEMADD           CHECK FOR LAST ADDRESS LOADED
       0044  0022  R
00045  0046  5C0C          RB(NE)  WRITE               REPEAT ROUTINE IF NOT LAST ADDRESS
00046                      *
00047                      *       THIS IS THE READ ROUTINE
00048                      *
00049  0048  81C0          LD      A9,MEMADD+2         LOAD START ADDRESS
       004A  0024  R
00050  004C  80A6    READ  LDR*    A8,A9               READ PATTERN FROM MEMORY
00051  004E  E8C8          CW      A8,PATEN,A2         COMPARE MEM CONTENTS WITH PATTERN
       0050  0026  R
00052  0052  5002          RF(E)   CONT                IF EQUAL CONTINUE READ ROUTINE
00053  0054  207F          HLT                         IF NOT EQUAL STOP ROUTINE
00054  0056  91A0    CONT  ADKL    A9,2                UPDATE MEMORY ADDRESS
       0058  0002
00055  005A  E9C0          CW      A9,MEMADD           CHECK FOR LAST ADDRESS READ
       005C  0022  R
00056  005E  5C14          RB(NE)  READ                CONTINUE IF NOT LAST WORD
00057  0060  1202          ADK     A2,2                UPDATE PATTERN COUNTER
00058  0062  EA20          CWK     A2,8                CHECK FOR LAST PATTERN
       0064  0008
00059  0066  5C34          RB(NE)  LDPAT               IF NOT LOAD NEW PATTERN
00060  0068  1701          ADK     A7,1                UPDATE SOFTWARE TIMER COUNT
00061  006A  EF40          CW      A7,TIMER            CHECK FOR LAST TEST
       006C  002E  R
00062  006E  5C3E          RB(NE)  LOAD1               IF NO CONTINUE WRITE/READ ROUTINES
00063                      *
00064                      *       THIS ROUTINE COMPUTES THE NEW START AND END ADDRESSES
00065                      *
00066  0070  84C0          LD      A12,MEMADD          SAVE END ADDRESS
       0072  0022  R
00067  0074  8140          LD      A1,MEMADD           LOAD END ADDRESS FOR COMPUTATION
       0076  0022  R
00068  0078  024E          LDK     A2,/4E              LOAD PROGRAM SIZE
00069  007A  9908          SUR     A1,A2               GIVES NEW START ADDRESS
00070  007C  8584          LDR     A13,A1              SAVE NEW START ADDRESS
00071  007E  8141          ST      A1,NEWADD           STORE NEW LOAD ADDRESS
       0080  0084  R
00072  0082  5702          RF(7)   *+4                 BYPASS NEWADD
00073  0084        NEWADD  RES     1
00074  0086  190A          SUK     A1,10               GET NEW END ADDRESS
00075  0088  8141          ST      A1,MEMADD           AND STORE IN MEMADD
       008A  0022  R
00076                      *
00077                      *       THIS LOADS THE NEW START ADDRESS IN MEMADD+2
00078                      *
00079  008C  0200          LDK     A2,0
00080  008E  8241          ST      A2,MEMADD+2
       0090  0024  R
00081                      *
00082                      *       THIS LOADS THE NEW END ADDRESS IN THE WRITE/READ ROUTINES
00083                      *
00084  0092  110A          ADK     A1,10
00085  0094  8141          ST      A1,WRITE+8
       0096  0044  R
00086  0098  8141          ST      A1,CONT+6
```

```
       009A  005C  R
00087                      *
00088                      *         THIS LOADS THE NEW ADDRESS OF MEMADD+2
00089                      *
00090  009C  1102                    ADK       A1,2
00091  009E  8141                    ST        A1,TEST+2
       00A0  003A  R
00092  00A2  8141                    ST        A1,READ+2
       00A4  004A  R
00093                      *
00094                      *THIS LOADS THE NEW ADDRESS OF PATEN
00095                      *
00096  00A6  1102                    ADK       A1,2
00097  00A8  8141                    ST        A1,LDPAT+2
       00AA  0036  R
00098  00AC  8141                    ST        A1,READ+4
       00AE  0050  R
00099                      *
00100                      *         THIS LOADS THE NEW ADDRESS OF TIMER
00101                      *
00102  00B0  1108                    ADK       A1,8
00103  00B2  8141                    ST        A1,CONT+22
       00B4  006C  R
00104                      *
00105                      *         THIS ROUTINE DOES THE TRANSFER
00106                      *
00107  00B6  83A0                    LDKL      A11,40
       00B8  0028
00108  00BA  8D40          REP       ML        10,MEMADD
       00BC  0022  R
00109  00BE  86A0                    LDKL      A14,20
       00C0  0014
00110  00C2  96C1                    ADS       A14,REP+2
       00C4  00BC  R
00111  00C6  8D37                    MSR       10,A13
00112  00C8  95A0                    ADKL      A13,20
       00CA  0014
00113  00CC  9BA0                    SUKL      A11,10
       00CE  000A
00114  00D0  5C18                    RB(NZ)    REP
00115  00D2  83A0                    LDKL      A11,/207F
       00D4  207F
00116  00D6  9DA0                    SUKL      A13,2
       00D8  0002
00117  00DA  8387                    STR       A11,A13
00118  00DC  9DA0                    SUKL      A13,64
       00DE  0040
00119  00E0  8F16                    ABR       A13
00120                              END       START
```

SYMBOL TABLE

COMP	0008	R	CONT	0056	R	LDPAT	0034	R	LOAD	0030	R
LOAD1	0032	R	MEMADD	0022	R	NEWADD	0084	R	PARAMS	0014	R
PATEN	0026	R	READ	004C	R	REP	00BA	R	START	0000	R
TEST	0038	R	TIMER	002E	R	WRITE	003C	R			

6.48 Programming for the MINI-PANEL

When running programs using a Mini-Panel it must be remembered that the START button can only be pushed once to start the IPL routine so the programs used must either start automatically or incorporate the INT button routine described previously. Also because there are only three LED's on the Mini-Panel it is not possible to display the contents of registers or memory locations. One way to overcome these minor problems is to write the program so that you can control it from the Operator's peripheral. Of course this will make the program a little longer but with the aid of either ASC4x4 or HEXTAP you should have no difficulty in producing a program tape that you can use with the Mini-Panel. An example of such a program is MINMEM which was produced as described in the previous three or four paragraphs. It is made up from several routines and Simple Test Programs already described. These are:

- The program LINE modified so that it will output a message instead of a character.
- The program NOECHO modified so that it will accept four data characters and the CR LF characters from the Operator's device and store them in a buffer.
- The Translate routine from HEXTAP modified so that it will translate data from the buffer into hexadecimal and load it into a register.
- The MEMHAN program modified so that if an error is detected it branches to a Fault routine instead of stopping on a HLT instruction.
- A new routine called Fault which translates data from the MEMHAN program so that details of the fault can be sent to the Operator's device.
- A new routine to enable the CF and RTN instructions to be used so that Input/Output routines for the Operator's device can be used for more than one function without being changed.
- An area reserved for the messages.

6.49 The program MINMEM was loaded using a standard IPL tape loader and once loaded starts automatically. After initializing the Stack Pointer register (A14) and the output message parameters, a Call Function to the routine TYPOUT (the modified program LINE) is started to output the first message "START ADDRESS?" to the operator. The RTN returns to the instruction after the CF and a 1 is subtracted from the message counter (A1). If the counter is not zero, the program branches to the CF to the routine INPT (the modified program NOECHO) which waits for a character to be input from the Operator's device. When this is received it is translated by the routine TRAN1 into a 4-bit hexadecimal character and then stored in register A2. The program branches back to INPT1 to accept the second character and the loop is continued until register A2 is full when the routine UPDA loads the contents of register A2 into A11. The RTN returns control to the instruction after CF to INPT. The LDKL and ADS instructions modify the contents of the instruction UPDA from 8388 to 8488 which makes the format of the instruction into LDR A12,A2 so that the next four data characters will be loaded into register A12 instead of A11. The RTN returns control to ENC to output the next message "LAST ADDRESS?" to the operator. The whole process is repeated and the last address data is stored in register A12. The next message output is "TEST PATTERN?" and the test pattern is loaded into register A13, then the message "PROGRAM STARTED" is output and the program branches to the routine called WRITE (the modified program MEMHAN) which starts to write and check read the test pattern into the area defined by the contents of A11 and A12. The routine is repeated until the repeat counter in register A7 is zero, then the program branches to the routine TESTOK. This routine outputs the message "TESTED OK RELOAD NEW PARAMETERS" and branches back to the start of program which will output the message "START ADDRESS?" New parameters can now be loaded and the routine WRITE will test the memory with the new data. If an error is detected in the read routine (RDC) the program branches to the routine FAULT. This routine gets the contents of registers A9 and A8 which will be the address in error and the pattern found in that address respectively. These contents are converted into printable characters and stored in the vacant locations of the area reserved for MESS1. When this has been done the routine ERMESS outputs the error message "ERR XXXX XXXX" where the first four Xs will be a memory address and the second four Xs will be the erroneous pattern read which can then be checked against the expected pattern on the Operator's log after the message "TEST PATTERN?"

6.50 There are only two things to remember when using this program:

- The response to the first three messages (which are all followed by a question mark thus ?) must be four hexadecimal characters in the range numerically 0 to 9 and alphabetically A to F.
- The program does not allow you to correct characters typed in error so if you make a mistake you can either let the program run or reload the program and start again.

6.51 The principles of this program can be used as a basis for writing your own programs using one of the other Simple Test Programs, for example the Paper Tape Punch, to test other parts of the system using the Mini-Panel. You can also include the routines for the MINI-IPL and INT button if you do not have the Standard IPL loader.

6.52 STANDARD TEST PROGRAMS

The Standard Test Programs and the minimum requirements for running them is shown in Table 6-1. All programs need a Full Control Panel, Bootstrap ROM for loading the IPL, and a Paper Tape Reader.

6.53 TEST TOOL

A test tool is under development to enable the service engineer to display microprogram addresses and provide step by step microprogram execution.

Table 6-1 Standard Test Programs

Type Number	Program Name (s)	Memory Needed	Description
PV251-120	CPZ8K & BZRAM	8K 8K	Basic Test Package for P851 CPU — comprises two test programs, one for the instruction set and the other for the Dynamic RAM.
PV251-100 *	BZPTR & BZPTP	4K 4K	Basic Test Package for Paper Tape Equipment — comprises test program for Paper Tape Reader P801-001 using either the P801-140 or P840-141 controller, test program for Paper Tape Punch P803-001 or FACIT 4060 using the P840-141 controller.
PV251-101 *	BZSCU	4K	Test Program for Serial Control Unit P845-140 and ASR, PER3100, or Hazeltine Display 1200 with either V24 or current loop interface
PV251-102 *	BZCR & BZLP	4K 4K	Basic Test Package for Card Reader and Line Printer using control unit P840-103 — it tests either Card Reader P806-102 or Line Printers P809-002 (X1415) or P809-004 (X1425).
PV251-103 * * *	BZSACU	8K	Test Program for SALCUZ card P845-160 with synchronous or asynchronous connections
PV251-130 *	BZFLOP	8K	Test Program for Floppy Disc Drive P830-005 using controller P830-140.

* — For these test programs the RTC is highly recommended but not mandatory.

* * — Diagnostic Box P849-901 or Loop Connector is required.

MINMEM

```
00000                                IDENT    MINMEM
00001                          *
00002                          *   THIS PROGRAM TEST THE MEMORY USING THE MINI PANEL
00003                          *
00004          0000            BEGIN    EQU      *
00005                                   RORG     BEGIN+/80
00006  0080                    BUFF     RES      3
00007  0086                             RES      20                    STACK AREA
00008          00AC            STACK    EQU      *+2
00009                          *
00010  00AE    208F            START    INH
00011  00B0    86A0                     LDKL     A14,STACK
       00B2    00AC  R
00012                          *
00013                          *
00014  00B4    0104            PARAM    LDK      A1,4
00015  00B6    80A0                     LDKL     A8,0
       00B8    0000
00016  00BA    0712            ENC      LDK      A7,18
00017  00BC    F6A1                     CF       A14,TYPOUT
       00BE    00D4  R
00018  00C0    1901                     SUK      A1,1
00019  00C2    8820                     ABL(Z)   WRITE
       00C4    0244  R
00020  00C6    F6A1                     CF       A14,INPT
       00C8    00F4  R
00021  00CA    8620                     LDKL     A6,/100
       00CC    0100
00022  00CE    9641                     ADS      A6,UPDA
       00D0    0148  R
00023  00D2    3F1A                     RB       ENC
00024                          *
00025  00D4    8620            TYPOUT   LDKL     A6,0
       00D6    0000
00026  00D8    46D0                     CIO      A6,1,/10              START TLETYPE
00027  00DA    5C04                     RB(NA)   *+2
00028  00DC    E542            TYPE1    LC       A5,MESS,A8
       00DE    01C8  R
00029  00E0    4510                     OTR      A5,0,/10
00030  00E2    5C04                     RB(NA)   *+2
00031  00E4    90A0                     ADKL     A8,1
       00E6    0001
00032  00E8    1F01                     SUK      A7,1
00033  00EA    5C10                     RB(NZ)   TYPE1
00034  00EC    4690                     CIO      A6,0,/10
00035  00EE    4CD0                     SST      A4,/10
00036  00F0    5C04                     RB(NA)   *+2
00037  00F2    F03A                     RTN      A14
00038                          *
00039                          *
```

```
00040  00F4  8320      INPT     LDKL     A3,BUFF
       00F6  0080  R
00041  00F8  0601      INPT1    LDK      A6,1
00042  00FA  46D0               CIO      A6,1,/10
00043  00FC  5C04               RB(NA)   *-2
00044  00FE  4D10               INR      A5,0,/10
00045  0100  5C04               RB(NA)   *-2
00046  0102  4690               CIO      A6,0,/10
00047  0104  4CD0               SST      A4,/10
00048  0106  5C04               RB(NA)   *-2
00049  0108  0600               LDK      A6,0
00050  010A  46D0               CIO      A6,1,/10
00051  010C  5C04               RB(NA)   *-2
00052  010E  4510               OTR      A5,0,/10
00053  0110  5C04               RB(NA)   *-2
00054  0112  4690               CIO      A6,0,/10
00055  0114  4CD0               SST      A4,/10
00056  0116  5C04               RB(NA)   *-2
00057  0118  E52D               SCR      A5,A3
00058  011A  1301               ADK      A3,1
00059  011C  EB20               CWK      A3,BUFF+6
       011E  0086  R
00060  0120  5C2A               RB(NE)   INPT1
00061  0122  9B20      TRANIN   SUKL     A3,6
       0124  0006
00062  0126  B208               XRR      A2,A2
00063  0128  B410      TRAN1    XRR      A4,A4
00064  012A  3A44               SLL      A2,4
00065  012C  E42C               LCR      A4,A3
00066  012E  8510               LDR      A5,A4
00067  0130  240F               ANK      A4,/0F
00068  0132  2540               ANK      A5,/40
00069  0134  ED20               CWK      A5,/40
       0136  0040
00070  0138  5402               RF(NE)   ADD
00071  013A  1409               ADK      A4,9
00072  013C  9210      ADD      ADR      A2,A4
00073  013E  9320               ADKL     A3,1
       0140  0001
00074  0142  EB20               CWK      A3,BUFF+4
       0144  0084  R
00075  0146  5C20               RB(NE)   TRAN1
00076  0148  8388      UPDA     LDR      A11,A2
00077  014A  F03A               RTN      A14
00078                    *
00079  014C  8720      FAULT    LDKL     A7,MESS1+6
       014E  0216  R
00080  0150  B410               XRR      A4,A4
00081  0152  E406               ECR      A4,A9
00082  0154  0502               LDK      A5,2
00083  0156  8290      REP      LDR      A10,A4
00084  0158  3C44               SLL      A4,4
00085  015A  A420               ANKL     A4,/0F0F
       015C  0F0F
00086  015E  A2A0               ANKL     A10,/0F0F
       0160  0F0F
00087  0162  9AA0               SUKL     A10,9
       0164  0009
```

```
00088  0166  5506          RF(5)    ADD9
00089  0168  92A0          ADKL     A10,/10
       016A  0010
00090  016C  5704          RF       SHIFT
00091  016E  92A0    ADD9  ADKL     A10,9
       0170  0009
00092                      *
00093  0172  9C20    SHIFT SUKL     A4,/900
       0174  0900
00094  0176  5106          RF(1)    ADD900
00095  0178  9420          ADKL     A4,/900
       017A  0900
00096  017C  5704          RF       OUREG
00097                      *
00098  017E  9420    ADD900 ADKL    A4,/1000
       0180  1000
00099                      *
00100  0182  AC0A    OUREG ORR      A4,A10
00101  0184  9420          ADKL     A4,/3030
       0186  3030
00102  0188  843D          STR      A4,A7
00103  018A  9720          ADKL     A7,2
       018C  0002
00104  018E  1D01          SUK      A5,1
00105  0190  5004          RF(Z)    CHECK
00106  0192  8406          LDR      A4,A9
00107  0194  5F40          RB       REP
00108                      *
00109  0196  EF20    CHECK CWK      A7,MESS1+16
       0198  0220  R
00110  019A  5008          RF(E)    ERMESS
00111  019C  9720          ADKL     A7,2
       019E  0002
00112  01A0  8182          LDR      A9,A8
00113  01A2  5F56          RB       FAULT+2
00114  01A4  80A0    ERMESS LDKL    A8,72
       01A6  0048
00115  01A8  0712          LDK      A7,18
00116  01AA  F6A1          CF       A14,TYPOUT
       01AC  00D4  R
00117  01AE  8F20          ABL      WRITE
       01B0  0244  R
00118  01B2  80A0    TESTOK LDKL    A8,90
       01B4  005A
00119  01B6  0722          LDK      A7,34
00120  01B8  F6A1          CF       A14,TYPOUT
       01BA  00D4  R
00121  01BC  8620          LDKL     A6,/300
       01BE  0300
00122  01C0  9E41          SUB      A6,UPDA
       01C2  0148  R
00123  01C4  8F20          ABL      START
       01C6  00AE  R
00124  01C8  5354    MESS  DATA     'START ADDRESS ?',/0D0A
       01CA  4152
       01CC  5420
       01CE  4144
```

```
                01D0  4452
                01D2  4553
                01D4  5320
                01D6  3F20
                01D8  0D0A
00125           01DA  4C41              DATA    'LAST ADDRESS ? ',/0D0A
                01DC  5354
                01DE  2041
                01E0  4444
                01E2  5245
                01E4  5353
                01E6  203F
                01E8  2020
                01EA  0D0A
00126           01EC  5445              DATA    'TEST PATTERN ? ',/0D0A
                01EE  5354
                01F0  2050
                01F2  4154
                01F4  5445
                01F6  524E
                01F8  203F
                01FA  2020
                01FC  0D0A
00127           01FE  5052              DATA    'PROGRAM STARTED',/0D0A
                0200  4F47
                0202  5241
                0204  4D20
                0206  5354
                0208  4152
                020A  5445
                020C  4420
                020E  0D0A
00128           0210  4552      MESS1   DATA    'ERR            ',/0D0A
                0212  5220
                0214  2020
                0216  2020
                0218  2020
                021A  2020
                021C  2020
                021E  2020
                0220  0D0A
00129           0222  5445              DATA    'TESTED OK RELOAD NEW PARAMETERS'
                0224  5354
                0226  4544
                0228  204F
                022A  4B20
                022C  5245
                022E  4C4F
                0230  4144
                0232  204E
                0234  4557
                0236  2050
                0238  4152
                023A  414D
                023C  4554
                023E  4552
                0240  5320
00130           0242  0D0A              DATA    /0D0A
```

MINMEM

```
00131  0244  8720       WRITE    LDKL     A7,/FFFF
       0246  FFFF
00132  0248  818E       WRITE1   LDR      A9,A11
00133  024A  85A7       LOAD     STR      A13,A9
00134  024C  91A0                ADKL     A9,2
       024E  0002
00135  0250  E992                CWR      A9,A12
00136  0252  5D0A                RB(5)    LOAD
00137  0254  818E                LDR      A9,A11
00138  0256  80A6       RDC      LDR*     A8,A9
00139  0258  E896                CWR      A8,A13
00140  025A  5004                RF(E)    SUIT
00141  025C  8F20                ABL      FAULT
       025E  014C  R
00142  0260  91A0       SUIT     ADKL     A9,2
       0262  0002
00143  0264  E992                CWR      A9,A12
00144  0266  5D12                RB(5)    RDC
00145  0268  1F01                SUK      A7,1
00146  026A  8820                ABL(Z)   TESTOK
       026C  01B2  R
00147  026E  5F28                RB       WRITE1
00148                             END      START
```

SYMBOL TABLE

```
ADD      013C  R   ADD9     016E  R   ADD900   017E  R   BEGIN    0000  R
BUFF     0080  R   CHECK    0196  R   ENC      00BA  R   ERMESS   01A4  R
FAULT    014C  R   INPT     00F4  R   INPT1    00F8  R   LOAD     024A  R
MESS     01C8  R   MESS1    0210  R   OUREG    0182  R   PARAM    00B4  R
RDC      0256  R   REP      0156  R   SHIFT    0172  R   STACK    00AC  R
START    00AE  R   SUIT     0260  R   TESTOK   01B2  R   TRAN1    0128  R
TRANIN   0122  R   TYPE1    00DC  R   TYPOUT   00D4  R   UPDA     0148  R
WRITE    0244  R   WRITE1   0248  R
```

SECTION VII

SCUZ CU

7.1 INTRODUCTION

SCUZ is a Serial Control Unit type Z designed to interface one peripheral device to the P851 system. There are three device interfaces which may connect the following peripherals:

- V24 Interface — Visual Display Unit — Hazeltine.
- Current Loop Interface 2 wire — ASR33 teletypewriter.
- Current Loop Interface 4 wire — PER 3100 matrix printer.

The peripheral sends and receives data in serial format and SCUZ translates this into parallel format for transfers between SCUZ and BIO. The serial format is selectable with 'U' links on the card, see paragraph headed Installation.

At CIO Start time the contents of a CPU Scratch Pad are transferred to SCUZ via BIO and have the following significance:

- BIO15 low specifies a CIO Start In.
- BIO15 high specifies a CIO Start Out.
- BIO12 and 13 are used to specify parity working, see paragraph heading, UART.

SCUZ synchronises the transfer of data between the device and the P851 system by operating in defined states which are controlled by a State Sequensor. For each state, the Sequensor generates a Control Word to control SCUZ operations.

A typical input operation is as follows:

IDLE — SCUZ is inactive waiting for a CIO Start.

EXIN — SCUZ is receiving data via the device interface.

ECHIN — SCUZ has a character ready so an interrupt is sent to CPU. When the interrupt is accepted CPU sends signal TMPN to SCUZ and the character is transferred via the BIO Lines. Signal TSMN is the reply to TMPN, thus ending the transfer.

PRWST2 — After the exchange is complete a CIO Halt switches the Sequensor to the Pre-Wait state before going to WAIT state.

WAIT — SCUZ sends an interrupt to CPU and waits in this state until CPU checks the status of SCUZ with an SST instruction. SCUZ now switches back to the IDLE state.

From IDLE state SCUZ may switch to either input or output mode depending on whether a CIO Start Input or Start Output is executed. For output mode a similar sequence occurs: IDLE, ECHOUT, EXOUT, PRWST2, WAIT, IDLE. For continuous exchanges SCUZ switches between the ECH (exchange) and EX (execute) states. When ECHO Mode is selected by the 'U' link each character that is sent to CPU is sent back to the device. For example if the device is an ASR33 teletypewriter and ECHO Mode is selected then every character sent to CPU is printed. If ECHO Mode is not selected the characters are not printed. For the tape reader the same is true but as this is a read operation only the control characters are significant e.g. starting and stopping. SCUZ may have two interrupt levels allocated IR0N and IR1N. IR0N requests the exchange and IR1N enables the operator to request an input transfer by pressing the Break Key at the device which causes IR1N to go low and request an input routine. When the SST instruction is executed only 2 bits are significant. BIO8 goes high to indicate that an attempt was made to input a character from the device while SCUZ was operating in output mode. BIO13 going high indicates that a parity error has occurred. Note that the terms H (high) and L (low) are used to indicate the logic levels.

7.2 INTERFACE SIGNALS (Figure 7-1)

- TMPN — Sent from CPU to SCUZ to initiate the data transfer.
- TSMN — SCUZ indicates to CPU that the data transfer is complete
- IR1N — Interrupt Request sent from SCUZ to CPU when the Break Key is pressed at the device.
- CPZ0N — CPU decodes an RIT instruction to enable signal CPZ0N to go low at SCUZ and reset the Break Detect F/F.
- IR0N — Interrupt Request sent to CPU when SCUZ is ready for a transfer ECHIN or ECHOUT states and in the WAIT state when SCUZ requests an SST instruction to end the transfer.

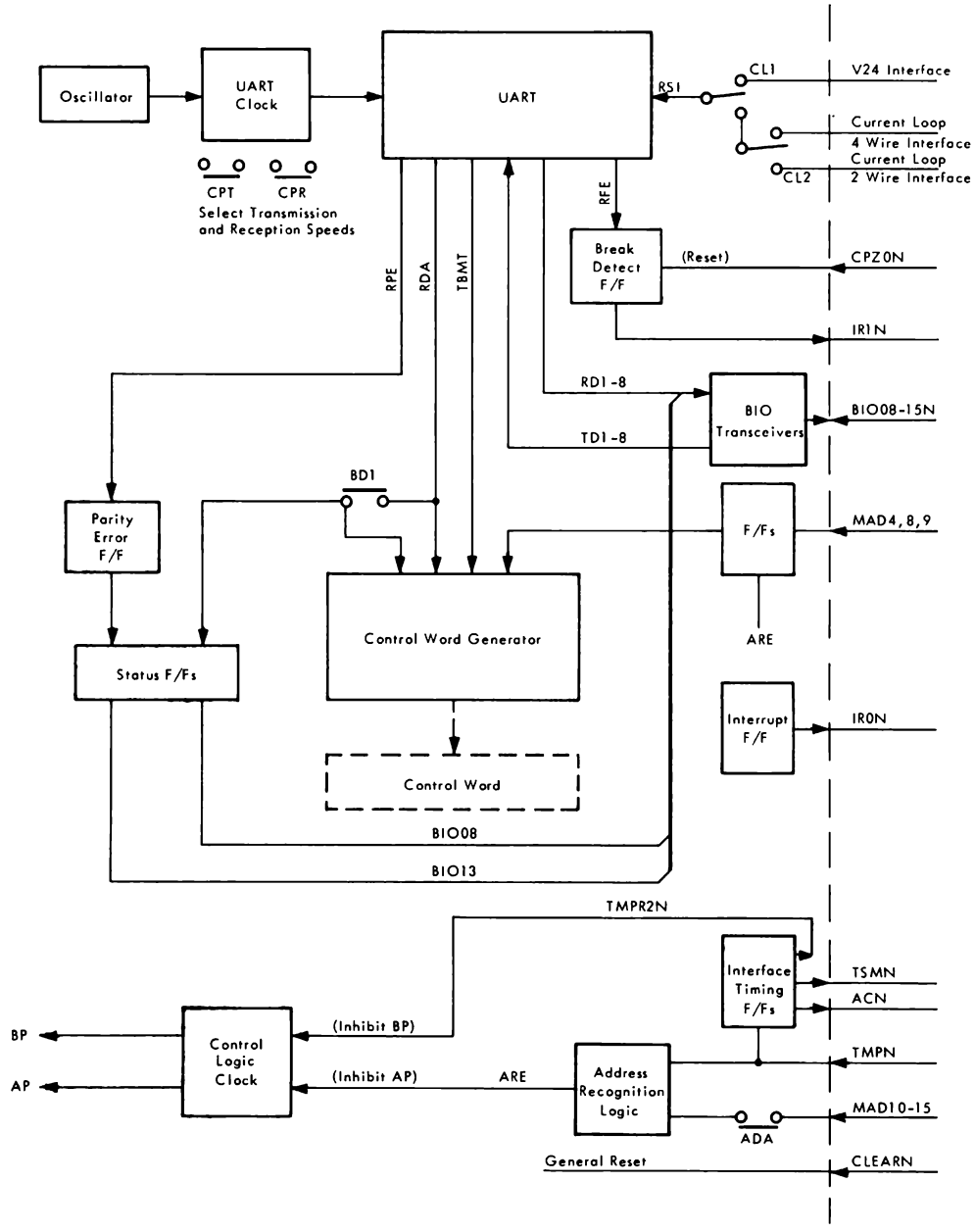


Figure 7-1 Interface Signals and Major Units of SCUZ

- ACN — Sent to CPU to indicate that the TMPN is received. ACN is an input to the CPU Condition Register.
- BIO08-15N — Eight Lines used to input and output data so in one exchange a maximum of 8 data bits may be transferred.
- MAD Lines — The MAD Lines are used in two separate groups by SCUZ: MAD10-15 contain SCUZ's own address, while MAD4,8,9 are inputs to the State Sequensor, specifying the I/O command.

7.3 V24 and Current Loop Interfaces

Selection of these interfaces is by 'U' link, see later in this section, paragraph headed Installation.

7.4 V24 Interface. This interface is connected to SCUZ by 4 signals electrically bonded to the machine, equipment frame or to an external ground.

- CT102 — Signal ground or common return. This conductor establishes the common reference potential for CT103.
- CT103 — Data transmitted from SCUZ to the device. This circuit is held at logic level 1 when no transfers are taking place.
- CT104 — Data received by SCUZ from the device.
- CT133 — Signal from the device when it is ready to receive the next character. This signal is selected by a 'U' link if required.

7.5 Current Loop Interface. There are two current loop interfaces either 2 or 4 wire and selection is by 'U' links. For example if the ASR33 device is to be used the 2-wire interface is selected and for the PER31 the 4-wire interface. Signals are as follows:

- CL1 — input data device to SCUZ.
- MCL1 — return.
- CL0 — output data SCUZ to device.
- MCL0 — return.

7.6 MAJOR LOGIC UNITS

The following paragraphs describe the major logic units of SCUZ.

7.7 BIO Transceivers

The BIO Transceivers enable an 8-bit character to be input or output from SCUZ. The direction of transfer is decided by the State Sequensor signal MVBN.

MVBN = L enables SCUZ → BIO

MVBN = H enables BIO → SCUZ

7.8 UART Clock

This clock receives its input from a 5.0688 Mhz crystal oscillator. The clock is a COM 5016 programmable divider which can be arranged to select any one of 16 different frequencies for transmission and reception. Selection of these frequencies is shown in a table of links at the end of this section.

7.9 Control Logic Clock

This is a 2 Mhz oscillator which generates two timing pulses AP and BP. The Clock is initiated when power goes on and then is switched on and off under control of the exchange timing signals TMPN and TSMN. See Timing Diagram, Figure 7-2.

7.10 Interface Timing F/Fs

These are 4 F/Fs connected to form a counter and control the timing of the exchange. They enable TSMN, ACN and inhibit the BP clock pulse. See Timing Diagram, Figure 7-2.

7.11 TranBuf Ready F/F

This flip-flop is latched when SCUZ is ready to send a new character to line. Signal TBMT or CT133 is selected by link to enable the Next address Generator to switch the State Sequensor to the ECHOUT State.

7.12 Interrupt F/F

SCUZ will set an interrupt in either ECHIN, ECHOUT or WST states. In each of these states State Sequensor signal MET1 is active high. At BP time MET1 sets the Interrupt F/F and enables interface signal IRON to CPU.

7.13 Break Key Detection F/F

The Break Key Detection F/F detects that the operator has pressed the Break Key at the device and sends an interface signal IR1N to interrupt the CPU. This interrupt is normally connected to the same level as the Control Panel interrupt but may be connected to any level depending on system requirements. The interrupt is reset via the I/O Bus with an RIT instruction which enables CPZON to reset the Break Key Detection F/F.

7.14 Status F/Fs

These are 4 F/Fs which record the status of SCUZ during an exchange so that when the SST is executed it will read the status of the device. These F/Fs have the following meaning:

- Not Operable condition — not used by SCUZ so the input to the F/F is held low.
- Throughput Error — not used by SCUZ so the input to the F/F is held low.
- Parity Error — this F/F is set and BIO13 goes low if a parity error has occurred during the exchange.
- Break Detection — selectable by link BD1. This F/F sets if UART detects a character from the device while working in output mode causing BIO08 to go low.

7.15 Address Recognition Logic

The SCUZ address code is received on the MAD10-15 Lines. This code is routed via the ADA Links and if the code is correct signal ARE is enabled. ARE enables interface signal TSMN to be sent to CPU.

7.16 UART (Universal Asynchronous Receiver Transmitter)

The UART Large Scale Integrated Circuit performs all the receiving and transmitting functions of SCUZ. The registers and associated signals of UART are illustrated in Appendix A, Integrated Circuits. The following paragraphs describe the UART registers and their signals.

7.17 UART Control Register. This element memorises the character format used for the data transmission. The signals are as follows:

- CS — This signal goes high to enter bits NDB1, NDB2, NSB, POE, NPB into the Control Register.
- NDB1, NDB2 — These control bits are decoded to select 7 or 8 bits per character. Selection is with a link.
  NDB1 = L, NDB2 = H means 7 data bits per character.
  NDB1 = H, NDB2 = H means 8 data bits per character.
- NSB — This input selects the number of stop bits. Selection is with a link.
  NSB = L means one stop bit.
  NSB = H means two stop bits.
- NPB, POE — These two inputs are enabled at CIO start time with BIO12 and 13 which in turn enable signals NPB and POE to select parity as follows:

BIO12	BIO13	NPB	POE	Mode
H	H	H	H	No Parity
H	L	L	H	Even Parity
L	H	H	L	No Parity
L	L	L	L	Odd Parity

Note that in output mode, if parity has not been selected the stop bit (or bits) immediately follows the last data bit.

7.18 UART Status Word Buffer Register. Output from this register indicates the state of transmission. The signals are as follows:

- SWEN — This signal is hardwired low which holds the Status Buffer Register permanently enabled.
- TBMT — This output is high when the Transmitter Buffer Register is to be loaded.
- RDA — This output is high when an entire character has been received and transferred into the Receiver Buffer Register.
- RDARN — This is a low level input which resets signal RDA.
- RFE — This output is high if the received character has no valid stop bit, occurring when a Break is sent from the device.
- RPE — This output is high if the received character parity bit does not agree with the selected parity.
- ROR — This output is not used.

7.19 UART Transmitter Registers. There are two transmitter Registers one being a Buffer and the other a Shift Register. The Transmitter Buffer Register is used as a waiting room for outgoing characters. When the previous character is completely transmitted the contents of the Buffer are immediately loaded into the Shift Register. The character is transmitted in the following order: start bit, data bits, parity bit when selected, and either one or two stop bits.

- TDSN — When this input goes low the data bits are entered into the Transmitter Buffer Register.
- TD1-8 — Eight data input lines routed from the BIO Lines.
- TSO — The character is transmitted in series from this pin, TSO remaining high when no data is being transmitted.
- TEOC — This output appears high each time that a complete character is transmitted. TEOC remains high up to the start of the next character.

7.20 UART Receiver Registers. There are two Receiver Registers one being a Buffer and the other a Shift Register. The Receiver Buffer Register is used as a waiting room for the incoming characters. The Receiver Shift Register receives the serial bits of the character after the start bit detection. When the complete character is received it is transferred to the Receiver Buffer Register.

- RSI — This input accepts the character in serial format. Data reception is initiated when this signal goes low.
- RDEN — This input is hardwired low so that when data enters the Receiver Buffer Register the RD1-8 lines are set.

7.21 UART Other Signals.

- RCP — The baud rate of this signal = 16 x the selected frequency.
- TCP — The baud rate of this signal = 16 x the selected frequency.
- MR — This input is connected to Master Clear and when active high sets TSO, TEOC, TBMT to a high level and resets RDA, RPE, RFE, ROR to the low level.

7.22 Parity Error Logic

SCUZ may generate the parity bit in input mode and check parity in output mode (see paragraph heading, UART). When the character is received into the UART chip during the OTR signal RDA (character received) is high. The parity check is carried out in UART and if a parity error has occurred then signal RPE (received parity error) goes high to clock and set the Parity Error F/F which sets one of the Status F/Fs ready for the SST instruction.

7.23 Control Word Generator

The control word is generated by a State Sequensor which comprises 3 ROMs with 5 common address lines AD00-AD04. The switching of the Sequensor between states is controlled in three different ways.

- One of the ROMs is selected according to the value of BIO15 and MAD08 as in the following table.

	BIO15	MAD08
Sequensor 1	H	H
Sequensor 2	L	H
Sequensor 3	X	L

- When a control instruction is executed MAD09 and MAD04 set up the address lines AD00 and AD01 respectively.
- For switching between intermediate states, address lines AD02, 03 and 04 set up a code derived from the control logic signals RDA, MTBMT and RDAB in conjunction with Sequensor signals MET1,2,3.

After the Sequensor is addressed a control word appears at the ROM output, the values of which are shown in Tables 1,2 and 3. Note that these values remain active until the next control word is addressed. The control word may be considered in three parts: Select next address (MET1,2 and 3), BIO Control (MAC and MVBN), UART Control (MRDEN, MTDS and MST).

Table 7-1 Sequensor 1 Contents

Function	ROM Address	B0 MET1	B1 MET2	B2 MET3	B3 MAC	B4 MRDEN	B5 MTDS	B6 MST	B7 MVBEN
IDLE → CIO ST OUT	0	L	H	L	H	L	L	H	H
IDLE → CIO HALT	1	L	L	L	H	H	L	L	H
IDLE → SST	2	L	L	L	L	H	L	L	H
IDLE → TST	3	L	L	L	L	H	L	L	H
EXIN → CIO ST OUT	4	L	L	H	L	H	L	L	H
EXIN → CIO HALT	5	L	H	H	H	H	L	L	H
EXIN → SST	6	L	L	H	L	H	L	L	H
EXIN → TST	7	L	L	H	L	H	L	L	H
EXOUT → CIO ST OUT	8	L	H	L	L	H	L	L	H
EXOUT → CIO HALT	9	L	H	H	H	H	L	L	H
EXOUT → SST	10	L	H	L	L	H	L	L	H
EXOUT → TST	11	L	H	L	L	H	L	L	H
PRWST2 → CIO ST OUT	12	L	H	H	L	H	L	L	H
PRWST2 → CIO HALT	13	L	H	H	H	H	L	L	H
PRWST2 → SST	14	L	H	H	L	H	L	L	H
PRWST2 → TST	15	L	H	H	L	H	L	L	H
WST → CIO ST OUT	16	H	L	L	L	H	L	L	H
WST → CIO HALT	17	H	L	L	H	H	L	L	H
WST → SST	18	L	L	L	H	H	L	L	L
WST → TST	19	H	L	L	L	H	L	L	H
ECHIN → CIO ST OUT	20	H	L	H	L	H	L	L	H
ECHIN → CIO HALT	21	L	H	H	H	L	L	L	H
ECHIN → SST	22	H	L	H	L	H	L	L	H
ECHIN → TST	23	H	L	H	L	H	L	L	H
ECHOUT → CIO ST OUT	24	H	H	L	L	H	L	L	H
ECHOUT → CIO HALT	25	L	H	H	H	H	L	L	H
ECHOUT → SST	26	H	H	L	L	H	L	L	H
ECHOUT → TST	27	H	H	L	L	H	L	L	H
PRWST1 → CIO ST OUT	28	H	H	H	L	H	L	L	H
PRWST1 → CIO HALT	29	H	H	H	H	H	L	L	H
PRWST1 → SST	30	H	H	H	L	H	L	L	H
PRWST1 → TST	31	H	H	H	L	H	L	L	H

Table 7-2 Sequensor 2 Contents

Function	ROM Address	B0 MET1	B1 MET2	B2 MET3	B3 MAC	B4 MRDEN	B5 MTDS	B6 MST	B7 MVBN
IDLE → CIO ST IN	0	L	L	H	H	H	L	H	H
IDLE → CIO HALT	1	L	L	L	H	H	L	L	H
IDLE → SST	2	L	L	L	L	H	L	L	H
IDLE → TST	3	L	L	L	L	H	L	L	H
EXIN → CIO ST OUT	4	L	L	H	L	H	L	L	H
EXIN → CIO HALT	5	L	H	H	H	H	L	L	H
EXIN → SST	6	L	L	H	L	H	L	L	H
EXIN → TST	7	L	L	H	L	H	L	L	H
EXOUT → CIO ST OUT	8	L	H	L	L	H	L	L	H
EXOUT → CIO HALT	9	L	H	H	H	H	L	L	H
EXOUT → SST	10	L	H	L	L	H	L	L	H
EXOUT → TST	11	L	H	L	L	H	L	L	H
PRWST2 → CIO ST OUT	12	L	H	H	L	H	L	L	H
PRWST2 → CIO HALT	13	L	H	H	H	H	L	L	H
PRWST2 → SST	14	L	H	H	L	H	L	L	H
PRWST2 → TST	15	L	H	H	L	H	L	L	H
WST → CIO ST OUT	16	H	L	L	L	H	L	L	H
WST → CIO HALT	17	H	L	L	H	H	L	L	H
WST → SST	18	L	L	L	H	H	L	L	L
WST → TST	19	H	L	L	L	H	L	L	H
ECHIN → CIO ST OUT	20	H	L	H	L	H	L	L	H
ECHIN → CIO HALT	21	L	H	H	H	L	L	L	H
ECHIN → SST	22	H	L	H	L	H	L	L	H
ECHIN → TST	23	H	L	H	L	H	L	L	H
ECHOUT → CIO ST OUT	24	H	H	L	L	H	L	L	H
ECHOUT → CIO HALT	25	L	H	H	H	H	L	L	H
ECHOUT → SST	26	H	H	L	L	H	L	L	H
ECHOUT → TST	27	H	H	L	L	H	L	L	H
PRWST1 → CIO ST OUT	28	H	H	H	L	H	L	L	H
PRWST1 → CIO HALT	29	H	H	H	H	H	L	L	H
PRWST1 → SST	30	H	H	H	L	H	L	L	H
PRWST1 → TST	31	H	H	H	L	H	L	L	H

Table 7-3 Sequensor 3 Contents

Function	ROM Address	B0 MET1	B1 MET2	B2 MET3	B3 MAC	B4 MRDEN	B5 MTDS	B6 MST	B7 MVBN
IDLE	0	L	L	L	L	H	L	L	H
IDLE → OTR	1	L	L	L	L	H	L	L	H
IDLE	2	L	L	L	L	H	L	L	H
IDLE → INR	3	L	L	L	L	H	L	L	H
EXIN	4	L	L	H	L	H	L	L	H
EXIN → OTR	5	L	L	H	L	H	L	L	H
EXIN	6	L	L	H	L	H	L	L	H
EXIN → INR	7	L	L	H	L	H	L	L	H
EXOUT	8	L	H	L	L	H	L	L	H
EXOUT → OTR	9	L	H	L	L	H	L	L	H
EXOUT	10	L	H	L	L	H	L	L	H
EXOUT → INR	11	L	H	L	L	H	L	L	H
PRWST2	12	L	H	H	L	H	L	L	H
PRWST2 → OTR	13	L	H	H	L	H	L	L	H
PRWST2	14	L	H	H	L	H	L	L	H
PRWST2 → INR	15	L	H	H	L	H	L	L	H
WST	16	H	L	L	L	H	L	L	H
WST → OTR	17	H	L	L	L	H	L	L	H
WST	18	H	L	L	L	H	L	L	H
WST → INR	19	H	L	L	L	H	L	L	H
ECHIN	20	H	L	H	L	H	L	L	H
ECHIN → OTR	21	H	L	H	L	H	L	L	H
ECHIN	22	H	L	H	L	H	L	L	H
ECHIN → INR	23	L	L	H	H	L	H	L	L
ECHOUT	24	H	H	L	L	H	L	L	H
ECHOUT → OTR	25	L	H	L	H	H	H	L	H
ECHOUT	26	H	H	L	L	H	L	L	H
ECHOUT → INR	27	H	H	L	L	H	L	L	H
PRWST1	28	H	H	H	L	H	L	L	H
PRWST1 → OTR	29	L	H	H	H	H	L	L	H
PRWST1	30	H	H	H	L	H	L	L	H
PRWST1 → INR	31	H	H	H	L	H	L	L	H

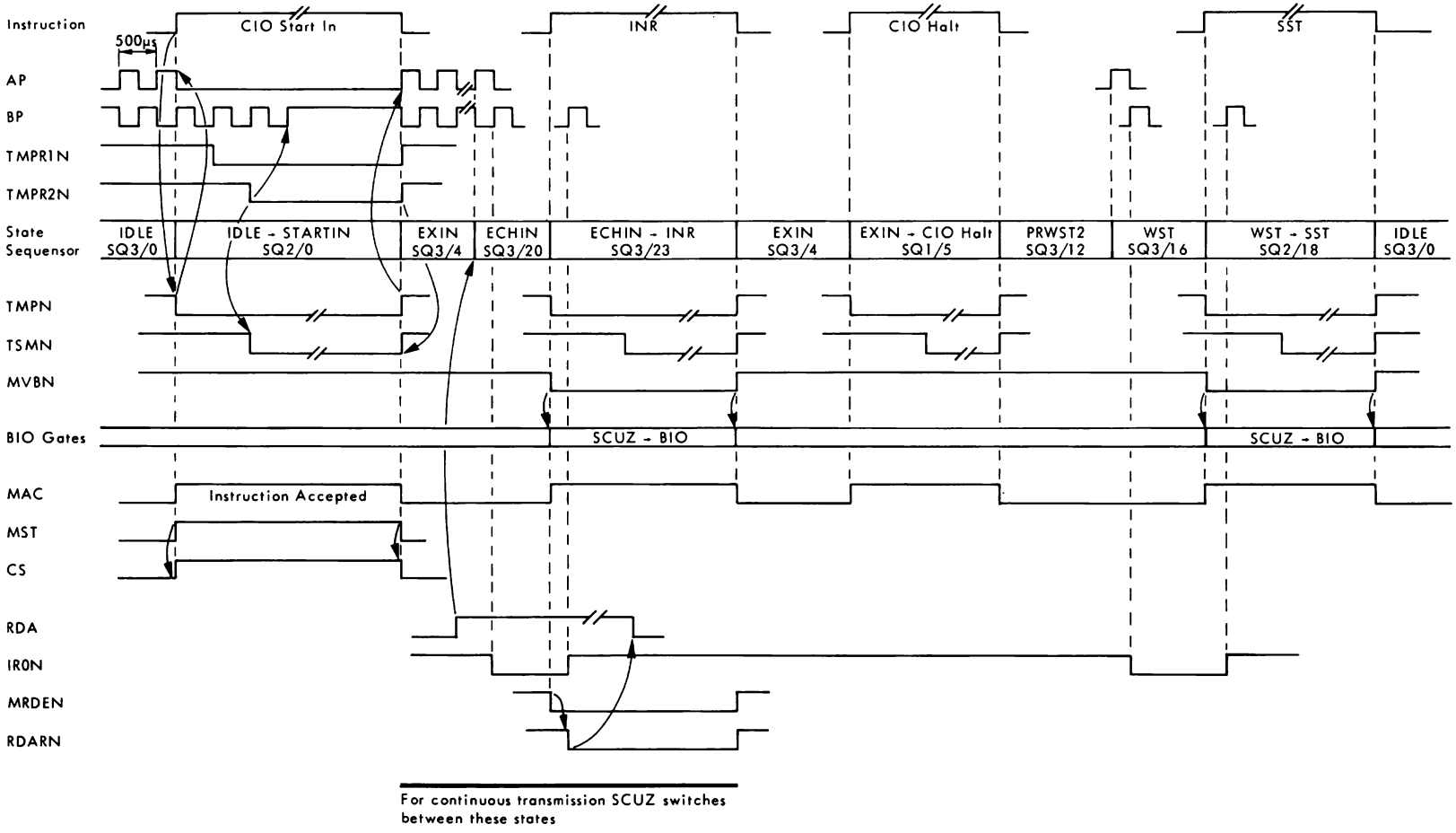
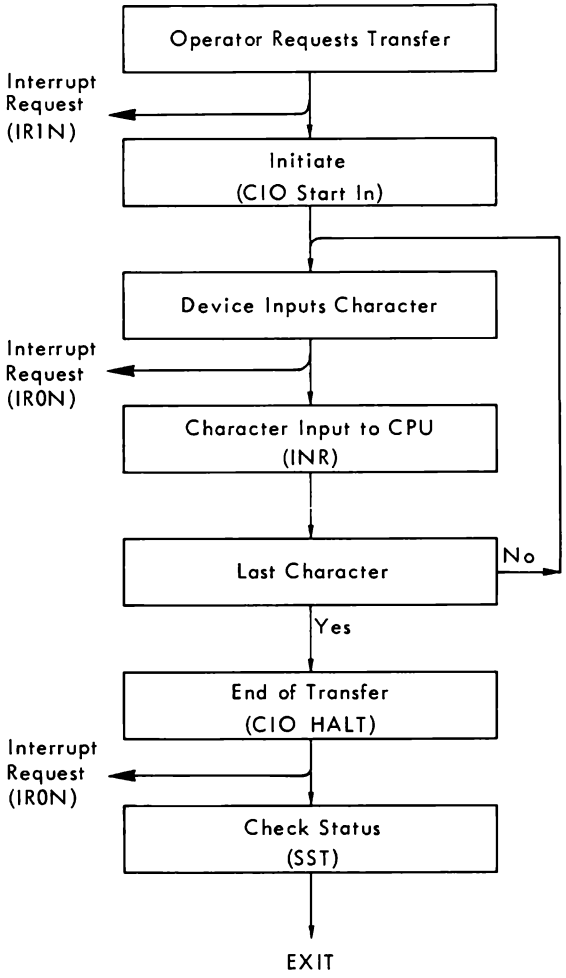


Figure 7-2 Timing for Input Mode

7.24 INPUT OPERATION

Initiation is by the operator and character exchanges take place under control of the Interrupt Request line. The transfer is considered to have ended when the SST is executed which switches SCUZ back to the IDLE state.



7.25 Operator Requests Transfer

Before data is input the operator must indicate to the CPU that a transfer is required by pressing the Break Key at the device. This Key generates a code without a stop bit, UART recognises a code with no stop bit and so signal RFE goes high to clock and set the Request Transfer F/F enabling Interrupt Request IR1N to be sent to CPU for a CIO Start.

7.26 Initiate

When the CIO Start is executed the State Sequensor is switched from Idle to the Idle → CIO ST IN state. Signal MST goes high to enable UART signal CS which clocks the character format, start, stop and parity bits, into UART. At the end of the transfer TMPN goes high to enable clock pulse AP which clocks the State Sequensor address lines to switch the state to EXIN.

7.27 Device Inputs Character

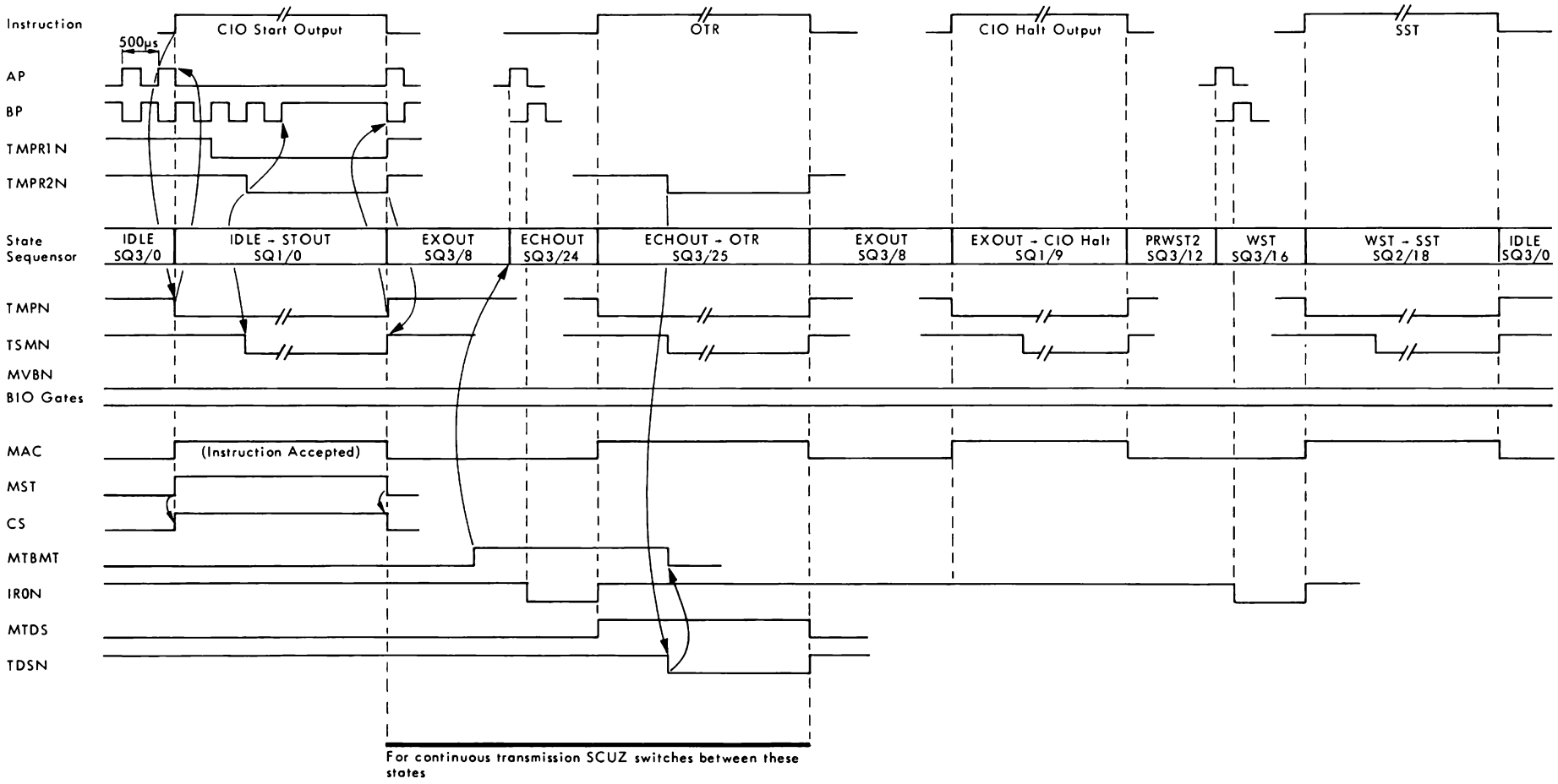
SCUZ remains in the EXIN state until a complete character is assembled using the RSI line into UART. When the character is complete UART output RDA goes high to switch the Sequensor to the ECHIN state. Signal MET1 goes active high so that the next BP clock pulse clocks and sets the Interrupt Request F/F to enable IR0N which requests an INR from CPU.

7.28 Character Input to CPU

The character is now input to CPU under control of the INR. Signal MVBN enables data in the direction SCUZ → BIO and signal MRDEN enables UART signal RDARN to reset RDA. At the end of the exchange TMPN goes high to initiate clock pulse AP, which switches the Sequensor back to the EXIN state. Data exchanges continue under control of TMPN and RDA switching SCUZ through the different states.

7.29 End of Transfer

At the end of transfer a CIO Halt is executed which switches SCUZ from EXIN to EXIN CIO HALT and the exchange stops. At the end of the CIO HALT, SCUZ switches to the PRWST2 state and then to WAIT state when Interrupt Request IR0N is sent to CPU for the SST instruction. SCUZ switches to the WAIT SST state, the Status F/Fs are enabled and signal MVBN enables the BIO gates in the direction SCUZ → BIO. For the significance of SST see paragraph headed, Status F/Fs.

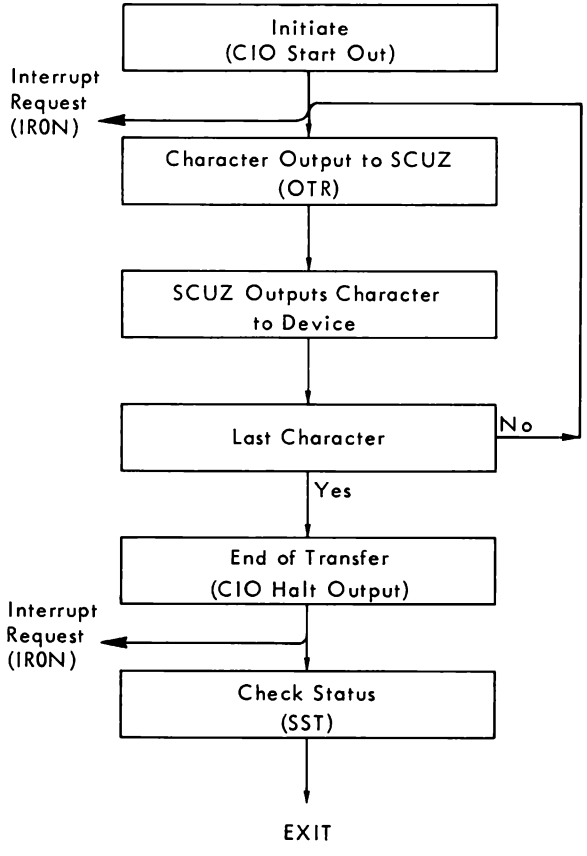


Note: If the device inputs a character and Break Detection is selected with link BD1 then the Sequensor switches to PRWST2 state.
Abbreviations in the form SQ3/8 means Sequensor 3, Address 8.

Figure 7-3 Timing for Output Mode

7.30 OUTPUT OPERATION

Initiation is by program and then character exchanges take place under control of the Interrupt Request line. The transfer has ended when the SST instruction switches SCUZ back to the IDLE state.



7.31 Initiate

When the CIO Start is executed the State Sequensor switches from IDLE to IDLE → ST OUT state. Signal MST goes high to enable UART signal CS which clocks the character format, start, stop and parity bits into UART. At the end of the transfer TMPN goes high to enable clock pulse AP which clocks the State Sequensor address lines to switch the state to EXOUT. When the UART Transmitter Buffer is empty signal TBMT goes high to clock and reset the TranBuff Ready F/F enabling signal MTBMT to go high and switch the Sequensor to the ECHOUT state on the next AP clock pulse.

7.32 Character Output to SCUZ

In the ECHOUT state signal MET1 is active high so on the first BP clock pulse the Interrupt F/F is clocked and set to enable IRON which requests an OTR from CPU. When the OTR is executed the State Sequensor switches to ECHOUT → OTR state. Signal TMPR2N enables TSMN to be sent to CPU to validate the exchange and also enables signal TDSN to clock the character into the Transmitter Buffer and SCUZ switches back to EXOUT. In the EXOUT state the character is transmitted in serial format to the device via the TSO output. For continuous exchanges the Sequensor continues to switch between the EXOUT and ECHOUT states under control of MTBMT and the OTR.

7.33 End of Transfer

When the last character has been output no more OTRs are sent. Instead a CIO Halt instruction switches the State Sequensor to the PRWST2 state. This is the Pre-Wait state and as soon as the last character has been transmitted signals TEOC and MTBMT go high switching the Sequensor to the WAIT state at the next clock pulse. In WAIT state signal MET1 is active high so the next BP pulse clocks and resets the Interrupt F/F enabling IRON to request the SST instruction from CPU.

7.34 INSTALLATION

SCUZ interface connections are via two connectors J1 and J3. J1 connects to the I/O Bus (see Volume 1, Section 2, Interface) and J3 to the device with either the V24 or Current Loop interfaces. Links are required on the card itself (see Table 7-7) and if Break Key Detection is specified then links are connected between SCUZ and CPU as follows:

- Signal CPZ0N between connectors 1A04 (SCUZ) and 2C11 (CPU).
- Signal IR1N between connector 1C04 (SCUZ) and the pin at which the Control Panel interrupt is connected at CPU.
- Signal IR0N between connectors 1C05 (SCUZ) and 2C04 (CPU) for int. level 6 for data/status exchange.

7.35 Device Interfaces

Connections for the three device interfaces are shown in Tables 7-4 to 7-6.

7.36 U Links

Before installation ensure that all links are fitted in accordance with Table 7-7.

Table 7-4 V24 Interface

Signals	SCUZ Connector J3	DB255 Device Connector	Remarks
CT101			Braid connected to Ground
CT102	07	07	Signal Earth or Common Return
CT103	02	03	Data SCUZ to Peripheral (Yel)
CT104	03	02	Data Peripheral to SCUZ (Blue)
CT105		04	Strap
CT106		05	
CT107		06	Strap
CT108/2		20	
CT109		08	Strap
CT133	25	25	Ready for Receiving (White)

Table 7-5 Current Loop Interface-2 Wire

Signals	SCUZ Connector J3	DB255 Device Connector	Remarks
Signal Ground	07		Strap
Return (MCL0)	11	11	Yellow
Received Data (CL1)	12	12	White
Return (MCL1)	13	13	Blue
Transmitted Data (CL0)	10	10	Red
Protective Earth			Connected to Braid

Table 7-6 Current Loop Interface-4 Wire

Signals	SCUZ Connector J3	DB255 Device Connector	Remarks
Ground			Connected to Braid
		08	Strap
		09	
Transmitted Data (CL0)	10	10	Red
Return (MCL0)	11	11	Yellow
Signal Ground	07		Strap
Received Data (CL1)	12	12	White
Return (MCL1)	13	13	Blue
		14	Strap
		15	
		03	Strap
		16	

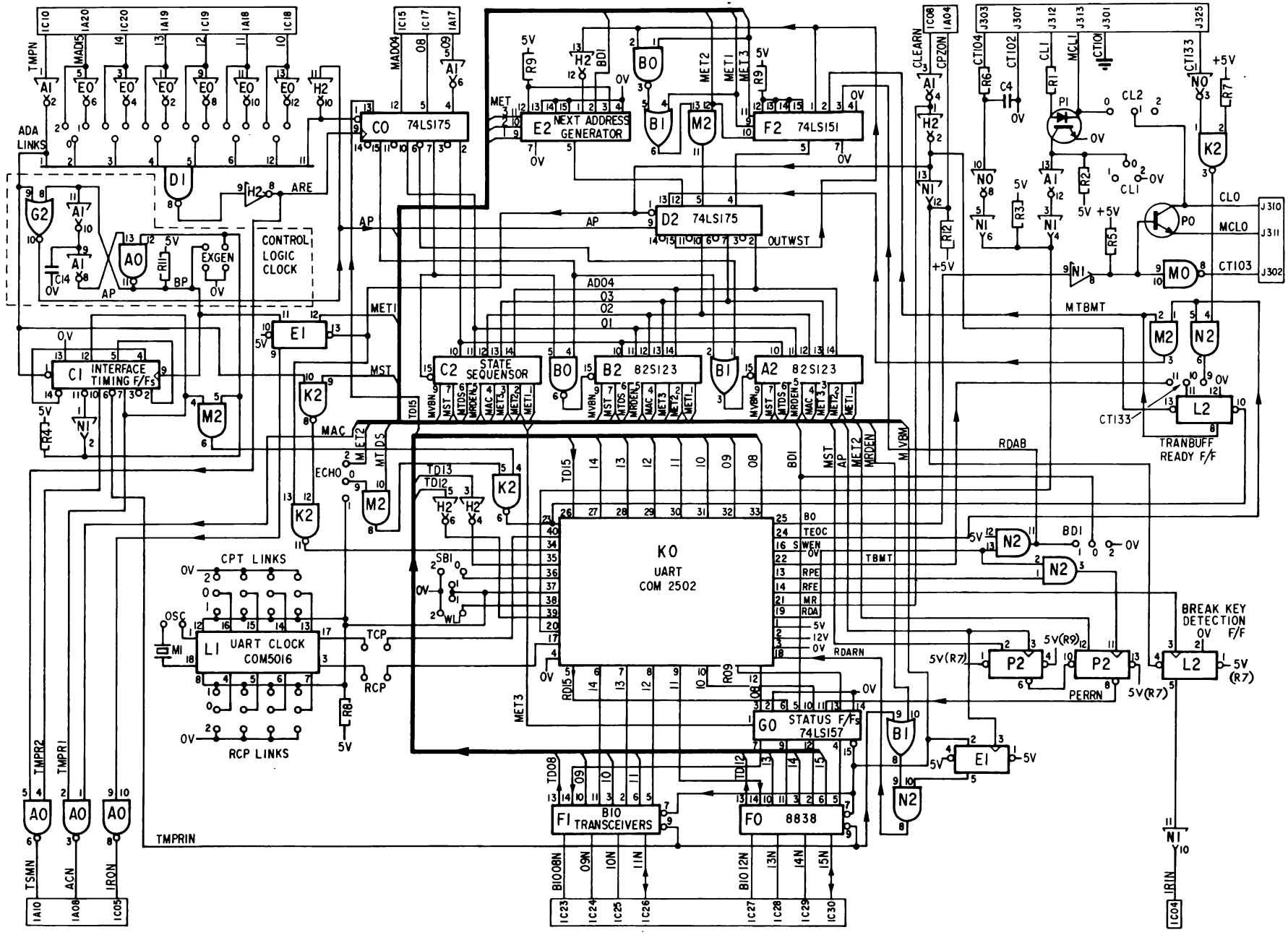


Figure 7-4 SCUZ Logic Diagram

Table 7-7 U Links

Link Name	Meaning	Selection
SB1	Select no of stop bits per character	2 stop bits: o o (0 1) — 1 stop bit: o o (0 2)
WL1	Select no of data bits per character	8 data bits: o o (0 1) — 7 data bits: o o (0 2)
BD1	Recognise a character input while in output mode?	Yes: o o (0 1) — No: o o (0 2)
ECHO	Echo Mode?	Yes: o o (0 1) — No: o o (0 2)
CL1	Select V24 or Current Loop Interface	Current Loop: o o (0 1) — V24: o o (0 2)
CL2	Current Loop? Then select 2 or 4 wire interface.	2 Wire: o o (0 1) — 4 Wire: o o (0 2)
TCP	Testing with an external generator?	Yes: o o — No: o o
RCP	Testing with an external generator?	Yes: o o — No: o o
CT133	Using signal CT133 from device to request transfer in output mode?	Yes: o o (10 9) — No: o o (10 11)
EXGEN	Testing with an external generator?	Yes: o o / o o — No: o o / o o
OSC	Crystal oscillator connected?	Yes: o o — No: o o
ADA	6 links must be fitted to select the address code	o o 0 / o o 1 / o o 2 / o o 3 / o o 4 / o o 5 (2 0 1) — 0 = most significant. This example shows the links connected for address 000000
CPT and CPR	4 links must be fitted to CPT and 4 to CPR in the same positions to select the indicated frequency	Link Positions to Select Indicated Frequency (Hz)

CPR (2 0 1)	CPT (2 0 1)	50	75	110	134.5	150	300	600	1,200	1,800	2,000	2,400	3,600	4,800	7,200	9,600	19,200
o o o	o o o	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
o o o	o o o	2	2	2	2	1	1	1	1	2	2	2	2	1	1	1	1
o o o	o o o	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1
o o o	o o o	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1

Note: 1 means link is between 0 and 1
2 means link is between 0 and 2

Table 7-8 SCUZ Parts List

Reference	Description	12NC Code
	Printed circuit	5111 100 06163
N1.	Integrated circuit 7417.	
E0,N2.	Integrated circuit 74LS04.	
A1.	Integrated circuit 74LS14.	
M2,N2.	Integrated circuit 74LS08.	
B0.	Integrated circuit 74LS00.	
G2.	Integrated circuit 74LS02.	
D1.	Integrated circuit 74LS30.	
B1.	Integrated circuit 74LS32.	
A0.	Integrated circuit 7438.	
F0,F1.	Integrated circuit 8838.	
E2,F2.	Integrated circuit 74LS151.	
G0.	Integrated circuit 74LS157.	
C0,C1,D2.	Integrated circuit 74LS175.	
E1,L2,P2.	Integrated circuit 74LS74.	
H1.	Integrated circuit COM 2502 or TR1602	
L1.	Integrated circuit COM 5016.	
M0.	Integrated circuit 1488.	
N0.	Integrated circuit 1489A.	
P1.	Phototransistor I.C. TIL111. or CNY47A	
O1.	Transistor I.C. BUY 47.	
A2.	Integrated circuit PROM 3152.	
B2.	Integrated circuit PROM 3162.	
C2.	Integrated circuit PROM 3171.	
K2.	Integrated circuit 74LS00	
R3-5,7-11.	Resistor 1Kn, 1/4W, 5%.	
R2.	Resistor 10Kn, 1/4W, 5%.	
R1.	Resistor 68.1n, 1/8W, ±1%.	
R6.	Resistor 220n, 1/4W, 5%.	
C4.	Capacitor 390pF, 100V, ±10%, ceramic.	
C14.	Capacitor 3300pF, 100V, ±10%,	
C1-3,5,8,15-18.	Capacitor 10µF, 25V, Fitco.	
C6,7,9-13.	Capacitor 10nF, ceramic.	
J1	Connector 100 964 053.	
J3.	Connector 206 604 1.	
M1	Quartz QA60 5068 8Khz	

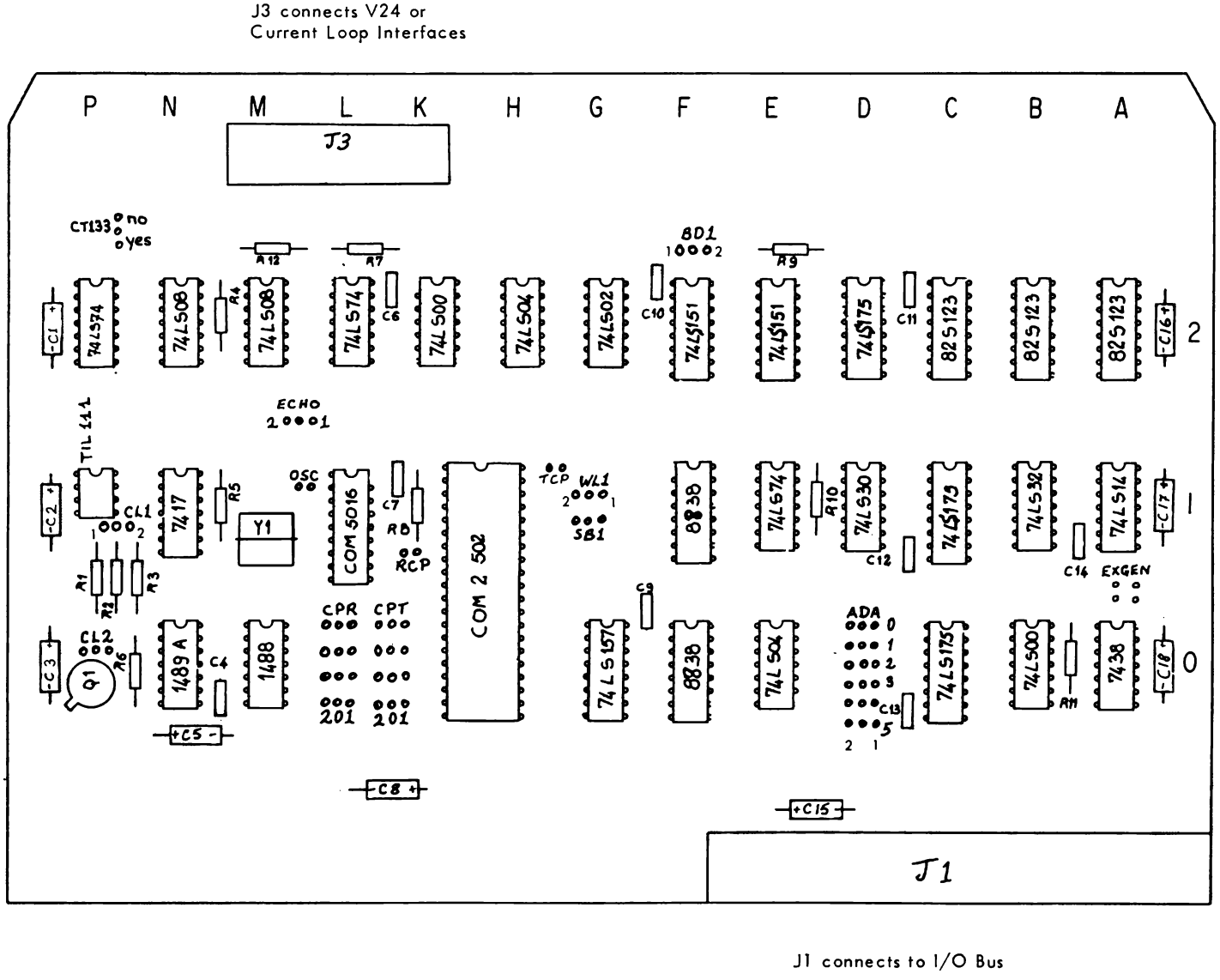


Figure 7-5 SCUZ Component Layout

SECTION VIII

4-8K DYNAMIC RANDOM ACCESS MEMORY

8.1 This memory uses either MOSTEK MK4096 or INTEL 2104A dynamic random access chips that can operate in either character or word mode during a read or write cycle. These devices store data in capacitors so the charge on the capacitor must be refreshed otherwise the data will be lost. Each memory module has its own refresh logic and the timing control circuits operate in asynchronous mode during data exchanges. The modules have a capacity of either 4K or 8K 16-bit words and the total capacity of the P851 can be extended up to a maximum of 32K words.

8.2 PHYSICAL

The memory modules are built on a double format Eurocard and have one 2x32 pin connector that plugs into the GPBS backpanel socket. Any of the sockets from 3 to 13 (2 to 13 if no IOP card is fitted) of the Eurorack may be used for the memory modules but in a mixed configuration of 4K and 8K modules, the 8K modules must precede the 4K modules. Figure 8-1 shows the partitioning layout of the memory card.

8.3 BLOCK DIAGRAM

Figure 8-2 shows the logic blocks of the memory module card.

8.4 MEMORY VOLTAGES

The memory module needs four voltages for its operation. Three of these are provided by the power supply and are the +5VL, the +5VM, and the +12VM. The fourth voltage is the -5V used by the memory chips, which is derived from the +12VM using transistor Q1 and its associated components.

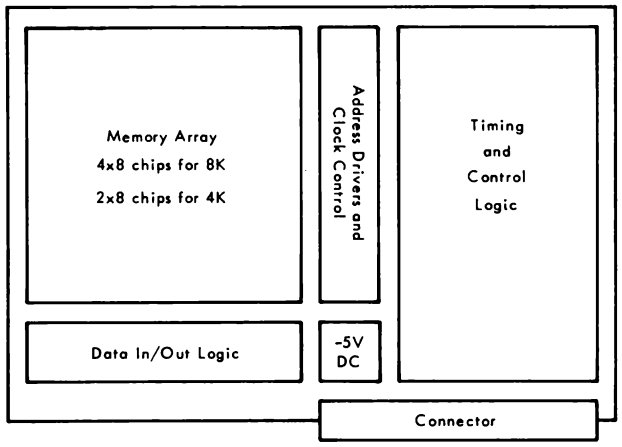


Figure 8-1 Memory Card Layout

8.5 MEMORY ARRAY

The memory array is arranged in a matrix of either four rows by eight columns (8K module) or two rows by eight columns (4K module) of MOS Random Access Memory (RAM) chips. For the 8K module, the first and third rows correspond to the left hand character (bits 00 to 07) and rows two and four correspond to the right hand character (bits 08 to 15) of the data word. During data transfer cycles for an 8K module only the two appropriate rows (left and right characters) for the selected address are activated, but during a refresh cycle all rows are activated. Figure 8-3 shows a block diagram of one of the memory chips. Each chip is a 4K one bit Read/Write Dynamic RAM and the complete memory array is shown in Figure 8-6.

8.6 Bit Selection

Each chip has only six address pins available to select each of the 4096 data bits (bits 0 to 4095) so the address on the twelve MAD lines (03 to 14) is sent as two 6-bit addresses each with its own enable signal. These two addresses come from the address logic on the card and are latched by the Row Address Select (RASN)

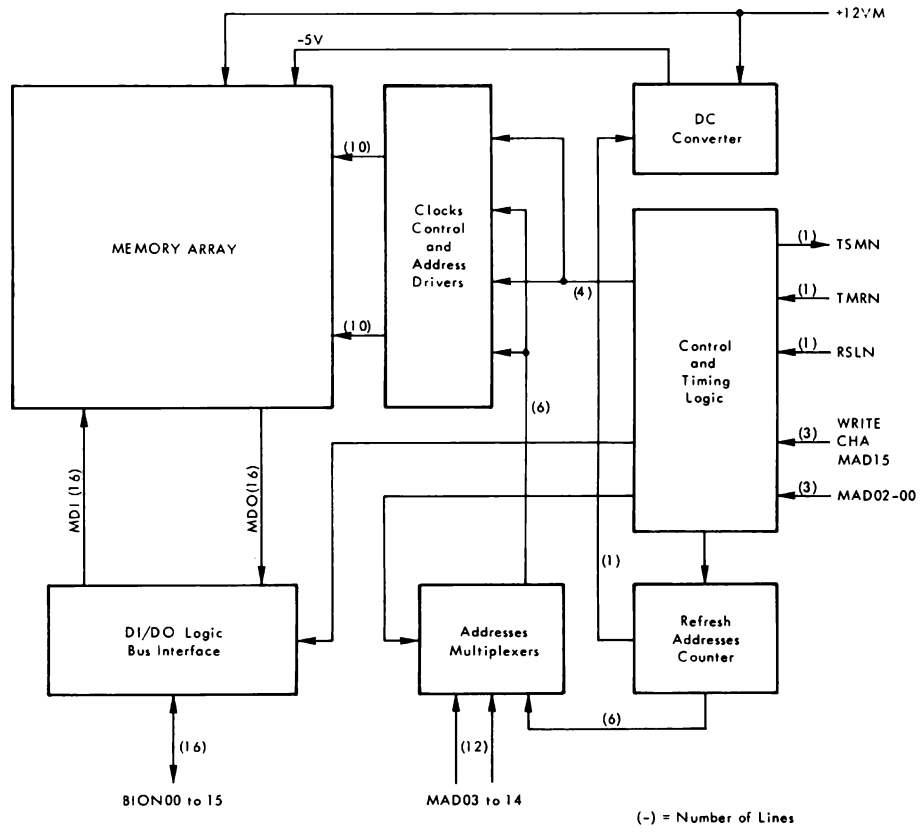


Figure 8-2 Logic Blocks of Memory Card

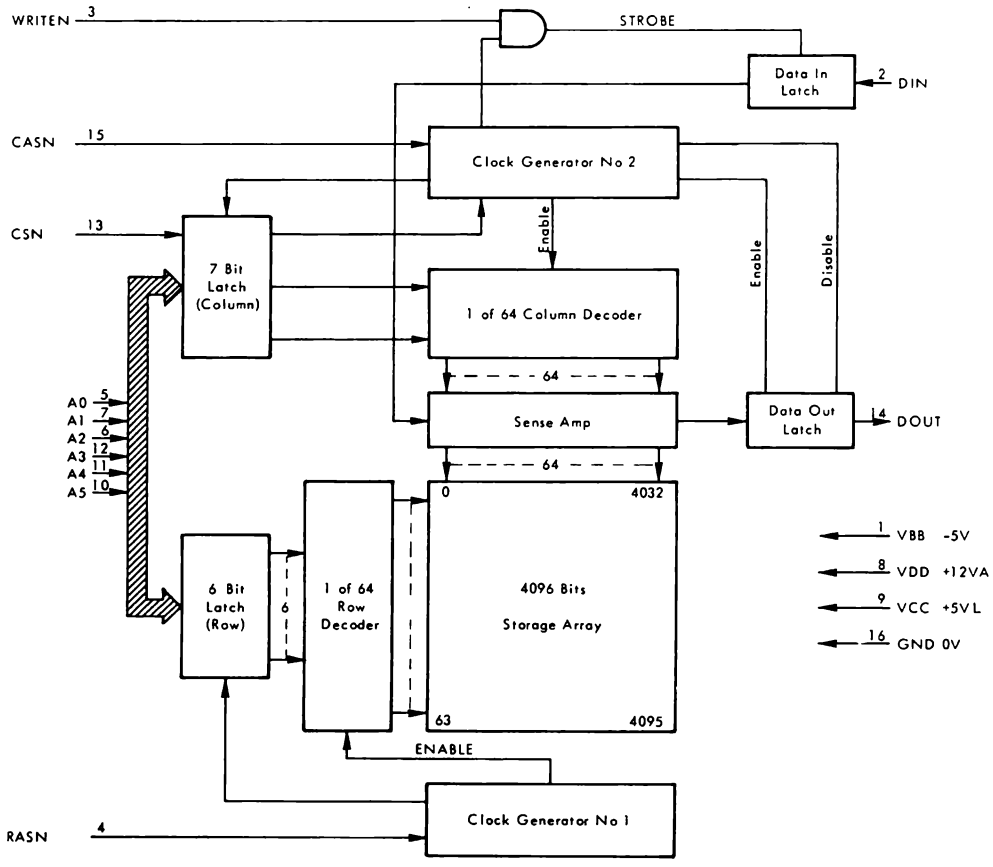


Figure 8-3 Functional Block Diagram of a 4K Dynamic RAM Chip

signal for the row address and by the Column Address Select (CASN) signal for the column address. Selection of the appropriate chips of the data word during character transfers is achieved by use of the CSN input pin, with the signal CSLN or CSRN selecting left and right hand characters respectively.

8.7 Data In/Out

The signal controlling the Read/Write operation is RWN which is connected to the WRITEN pin of the chip. If RWN is low a write operation is performed, and if high a read operation is performed. Each time a memory cycle is performed the CASN signal must be activated on all the chips of the data word at the beginning of the cycle to set the DOUT output into the OFF state so that invalid data is prevented from reaching the bus logic.

8.8 ADDRESSING LOGIC

The addressing logic is activated by either a Memory Cycle from the CPU or a Refresh Cycle from the memory logic, and is shown in Figure 8-5.

8.9 Memory Cycle Addressing

For a Memory Cycle the address on the twelve MAD lines (03 to 14) is enabled through the three Data Selector/Multiplexer chips A1, B1 and C1 by the signals REFN and STAD. During the cycle REFN is high and the row address (MAD09 to MAD14) is enabled when STAD is low. The column address (MAD03 to MAD08) is enabled when STAD is high. Both addresses are latched into the memory array by the signals RASN and CASN respectively. The cycle is validated by the signal TMRN from the CPU and will only take place if there is no Refresh Cycle in progress.

8.10 Refresh Cycle Addressing

Each memory location must be refreshed every 2mS otherwise the data will be lost, therefore all 64 row addresses must be activated sequentially within this period. An oscillator comprising D3, R18 and C49 produces clock pulses to the asynchronous counters A2 and B2 (Figure 8-5). The rising edge of the oscillator clocks the counters once every 30µs and enables the six refresh address lines (RAD00 to RAD05) to select the 1 to 64 row addresses of the memory array within a 2mS

period. The falling edge of the oscillator produces the Refresh Request signal RREFN which in turn produces REFN (low) via B3 and C4 to enable the address lines through chips A1, B1, and C1. The selected row address is applied to each chip in the memory array at the same time. At the end of the Refresh cycle RREFN is reset by the signal CLREFN.

8.11 Memory Module Addressing

The maximum memory capacity of the CPU is 32K words and it can be made up from either all 4K, all 8K or a mixture of both 4K and 8K modules (with a maximum of four cards per system) as required by the system. When a mixture of both 4K and 8K modules is used, the 8K modules must be fitted into the slots nearest to the CPU card. In all configurations of memory modules the cards must be fitted in ascending order of module address starting from the CPU card. The logic for memory module addressing is at C0 on Figure 8-5. Wire links are soldered on the card during manufacture to make the selection for either 4K or 8K modules. Module addressing is made with U links, three for a 4K module and two for an 8K module; the link configuration is shown in Table 8-1.

8.12 CONTROL LOGIC

This logic controls the modes of operation on the memory modules and is found on Figure 8-5. The modes are either internal Refresh cycles instigated by the memory module, or Read/Write cycles in either Word or Character mode instigated by the CPU.

8.13 Refresh Cycles

When RREFN goes low it prevents the CPU flip-flop B3 (output pins 9 and 7) from being set and inhibits External cycles. The output from pin 7 is used to drive the row select logic via gates C4 pins 11 and 3 and gate A3 pin 6. At the end of the refresh cycle B3 is reset by the signal OFFN.

Table 8-1 Memory Module Addressing Links

No of Words	4K Module						8K Module			
	MAD00		MAD01		MAD02		MAD00		MAD01	
	M00	M00N	M01	M01N	M02	M02N	M00	M00N	M01	M01N
0-4K		X		X		X				
4-8K		X		X	X					
8-12K		X	X			X				
12-16K		X	X		X					
16-20K	X			X		X				
20-24K	X			X	X					
24-28K	X		X			X				
28-32K	X		X		X					
0-8K								X		X
8-16K								X	X	
16-24K							X			X
24-32K							X		X	

8.14 External Cycles

The external cycles are initiated by the signal TMRN going low clocking the CPU flip-flop B3 via gate D1. The output from pin 9 is used via gate A3 pin 8 to clock the Read/Write mode register D0 and to set the RAS flip-flop A3 pin 6 high. This positive going change or RAS initiates the memory cycle and triggers an oscillator controlling the timing of the memory.

8.15 TIMING LOGIC

This logic is used for both internal and external cycles and comprises an oscillator (made with a 100nS delay line B4) and four flip-flops B3, D4, and C2. Different edges of the oscillator output clock the flip-flops and the logic is shown on Figure 8-5.

8.16 Address Multiplex Clock

This is the STAD signal produced by flip-flop B3 when it is clocked by A4 pin 1. The output from pin 5 (of B3) multiplexes the output from the addressing logic and it will be high for one period of the oscillator for an External cycle, but it will remain high for two periods for an Internal cycle.

8.17 Column Address Clock

This is the CAS signal produced by flip-flop D4 when it is clocked at T6 time of the oscillator. The output from pin 5 goes high and clocks the column addresses in the memory array and the putput from pin 6 drives the delay circuit of CR4, R22, and C44 and is then used as input to the other half of D4.

8.18 End of Access Clock

This is the OFF signal produced by flip-flop D4 when it is clocked at the second T10 time of the oscillator; the first T10 is suppressed due to the delayed input to pin 11. The output from pin 9 is used to produce INHN (disabling the refresh logic when high) and to clock the memory response to the CPU.

8.19 Memory to CPU Clock

This is the TSMN signal produced by flip-flop C2 when it is clocked by the output from D4 pin 9 going high at the end of access time. The output from pin 5 is only used for external cycles and it signals the CPU that it is ready for a data transfer. The output from pin 6 is used together with OFFN to inhibit refresh cycles during an external cycle.

8.20 Timing Diagrams

Figure 8-4 shows the timing for external Read/Write and internal Refresh cycles. The times shown on the diagrams are typical.

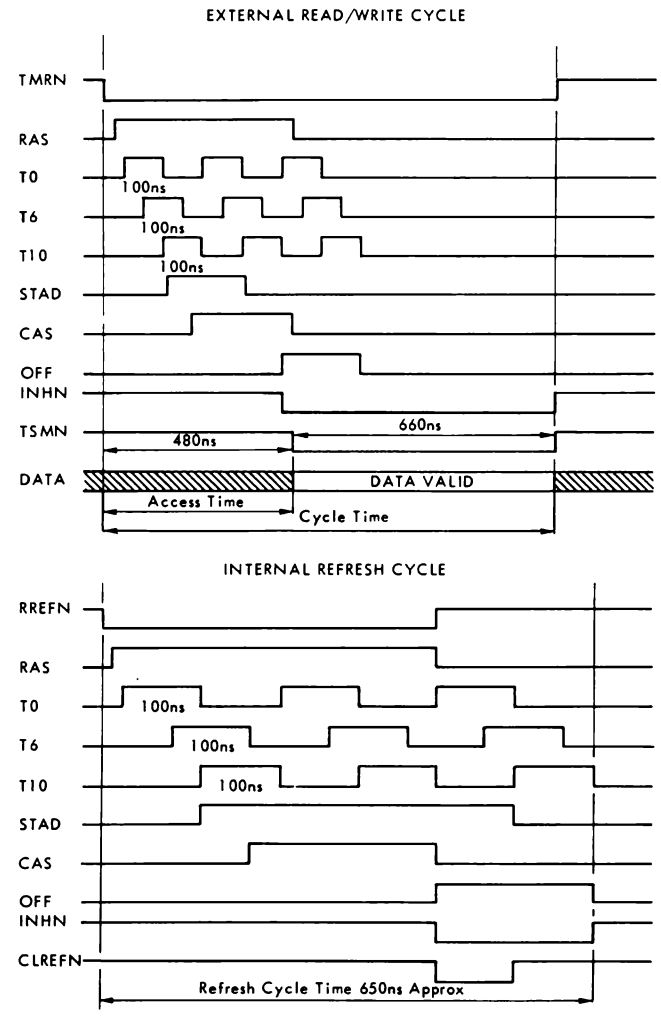


Figure 8-4 Timing Diagram

8.21 DATA IN/OUT LOGIC

This logic comprises four data selector/multiplexer chips G0, K0, R0, and T0 used to select the correct data path into or out of the memory array during data transfers and is shown in Figure 8-6.

8.22 Word Handling

In this mode the CHA input is low enabling ELCH and ERCH (chip D0 Figure 8-5) and, via the memory array interface drivers E2 and E4, both left and right characters of the selected word. ERCH is also used during Read cycles to enable the right-hand character to the bus logic (the left-hand character is automatically enabled) so that the whole word can be sent to the CPU; it also enables the right-hand character into the memory array during Write cycles.

8.23 Character Handling

In this mode the CHA input is high and the MAD15 input is either low (left-hand character) or high (right-hand character) selecting ELCH or ERCH respectively (Figure 8-5 chip D0). The state of ERCH enables the multiplexer chips to transfer the left-hand character bits in memory (MDO00 to 07) to the right-hand character position of the BIO emitters (BION08 to 15) during Read mode (ERCH high) and to transfer the right-hand character of the BION lines to the left-hand character position in memory during Write mode (ERCH low).

8.24 Read/Write Modes

This mode is selected by the state of the WRITE signal from the CPU, when low Read mode and high Write mode are selected. This produces two signals from chip D0 (Figure 8-5): FW used by the memory array and EDON used by the GPBS bus transceivers.

8.25 BUS TRANSCEIVERS

This logic is shown on Figure 8-6 and comprises chips F0, H0, P0, and S0 which provide the interface between the BION lines of the GPBS and the memory data logic. When EDON is low data from the memory is enabled onto the BION lines during a read cycle, and when EDON is high data from the BION is enabled into the memory during a write cycle.

8.26 D.C. CONVERTER

This circuit is shown on Figure 8-5 and comprises transistor Q1 and its associated circuitry. This switching converter has a frequency of approximately 16KHz and the input is provided by the QA output from B2. It converts +12VM to Q1 into a -5V supply for the memory array.

8.27 INTERFACE CONNECTIONS

Table 8-2 gives interface connections of the memory card connector socket.

8.28 RESET LOGIC

This logic is on Figure 8-5 and is provided by D3, C9, and R14. At power ON time its output resets the control flip-flops once only. The RSLN input to B3 pin 11 ensures that this flip-flop is the correct state for the operation of the Refresh logic when a power failure occurs or the mains is disconnected.

8.29 LOGIC DIAGRAMS, CARDS, and CARD COMPONENTS

The logic diagrams of the memory card are shown in Figures 8-5 and 8-6 and were derived from the E.D.S. dated 770308. The card layout for the 4K module is shown in Figure 8-7 and the layout for the 8K module is shown in Figure 8-8. Table 8-3 lists the components of the 4K module and Table 8-4 lists the components of the 8K module.

Table 8-2 Interface Connections

Pin Number	Signal Name
1C01	
1C02	+5VM
1C03	
1C04	
1C05	
1C06	0V
1C07	
1C08	
1C09	0V
1C10	
1C11	
1C12	0V
1C13	MAD00
1C14	MAD02
1C15	MAD04
1C16	MAD06
1C17	MAD08
1C18	MAD10
1C19	MAD12
1C20	MAD14
1C21	CHA
1C22	0V
1C23	BIO08N
1C24	BIO09N
1C25	BIO10N
1C26	BIO11N
1C27	BIO12N
1C28	BIO13N
1C29	BIO14N
1C30	BIO15N
1C31	+12VM
1C32	+5VL

Pin Number	Signal Name
1A01	
1A02	+5VM
1A03	
1A04	
1A05	
1A06	0V
1A07	RSLN
1A08	
1A09	0V
1A10	TSMN
1A11	TMRN
1A12	0V
1A13	MAD01
1A14	MAD03
1A15	MAD05
1A16	MAD07
1A17	MAD09
1A18	MAD11
1A19	MAD13
1A20	MAD15
1A21	WRITE
1A22	0V
1A23	BIO00N
1A24	BIO01N
1A25	BIO02N
1A26	BIO03N
1A27	BIO04N
1A28	BIO05N
1A29	BIO06N
1A30	BIO07N
1A31	+12VM
1A32	+5VL

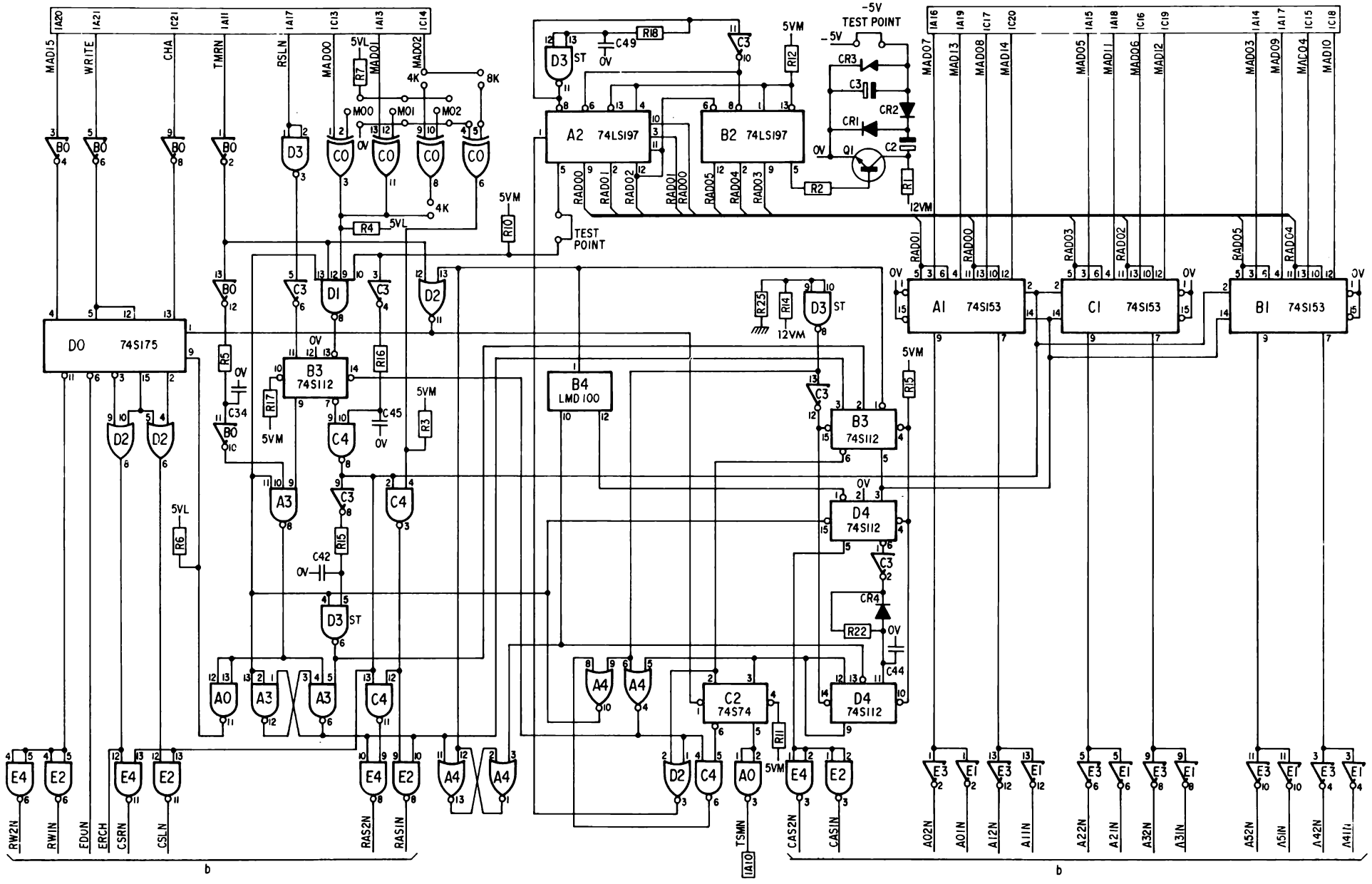


Figure 8-5 Logic Diagram (a)

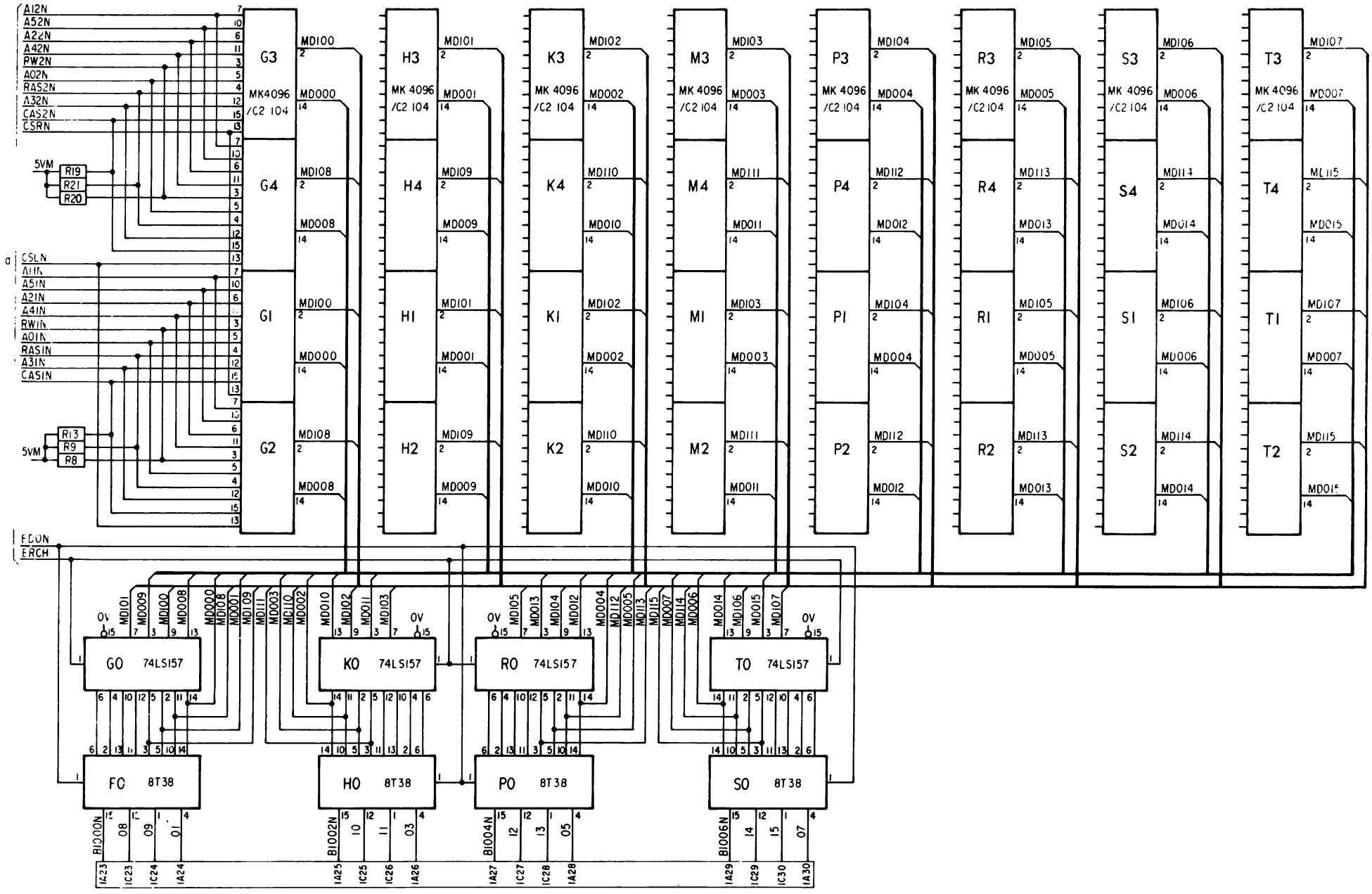


Figure 8-6 Logic Diagram (b)

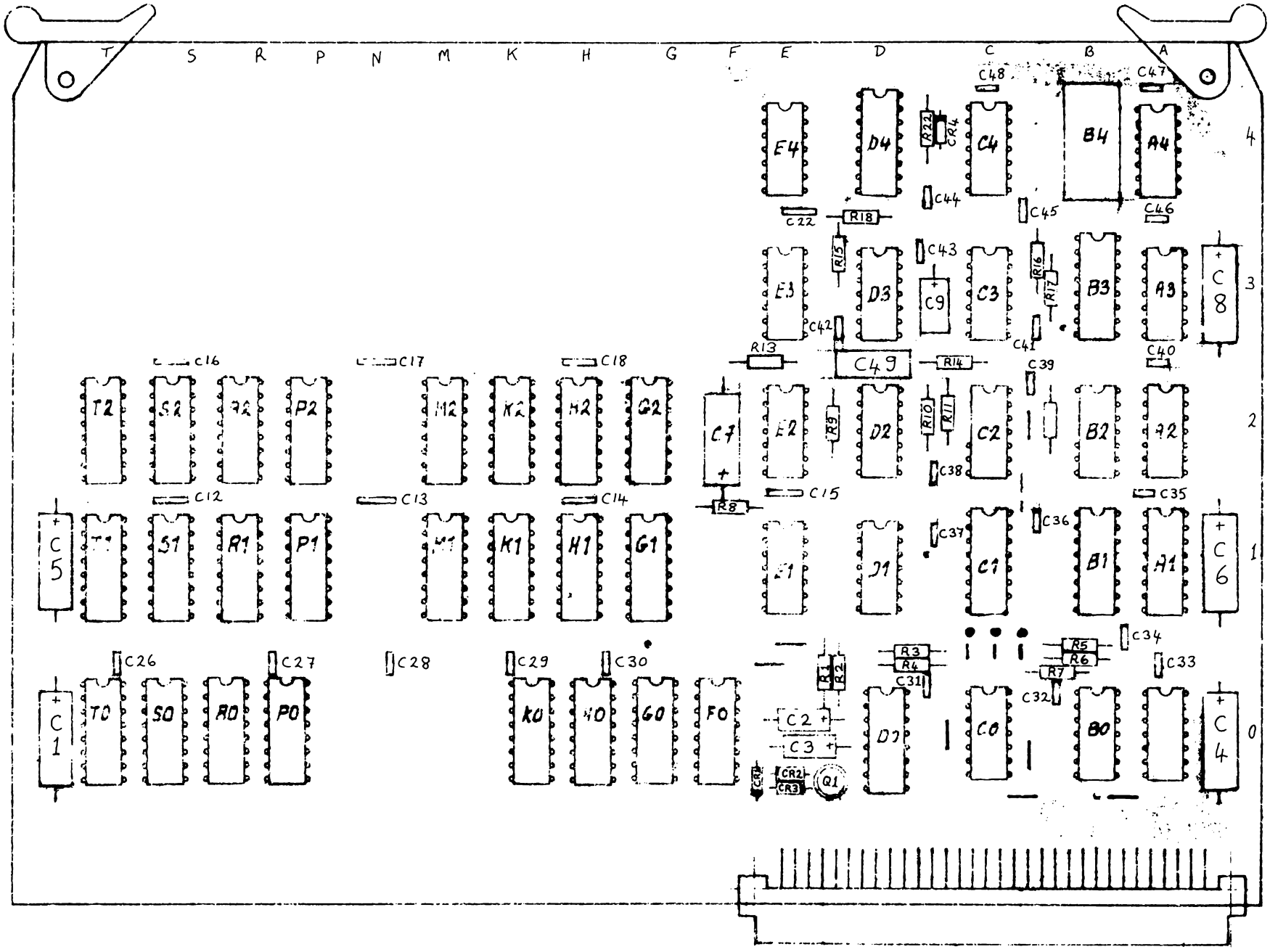


Figure 8-7 4K Memory Card (RAMDZ4 — 16 Bits) Component Layout

Table 8-3 4K Memory Card (RAMDZ4 — 16 Bits) Parts List

Reference	Description	12NC Code
	Printed circuit	5111 100 06183
E2,E4.	Integrated circuit 74S00.	
C4.	Integrated circuit 74LS00.	
A4.	Integrated circuit 74S02.	
E1,E3.	Integrated circuit 74S04.	
C3.	Integrated circuit 74LS04.	
A3.	Integrated circuit 74S10.	
D1.	Integrated circuit 74S20.	
D2.	Integrated circuit 74LS32.	
A0.	Integrated circuit 7438.	
C2.	Integrated circuit 74S74.	
B3.	Integrated circuit 74S112.	
D4.	Integrated circuit 74LS112.	
D3.	Integrated circuit 74LS132.	
C0.	Integrated circuit 74LS136.	
A1,B1,C1.	Integrated circuit 74LS153.	
G0,K0,R0,T0.	Integrated circuit 74LS157.	
D0.	Integrated circuit 74S175.	
A2,B2.	Integrated circuit 74LS197.	
F0,H0,P0,S0.	Integrated circuit 8838.	
B0.	Integrated circuit REC 0613.	
B4.	Integrated circuit LDM 100.	
G1,G2,H1,H2,K1,K2,M1,M2,P1,P2,R1, R2,S1,S2,T1,T2.	Integrated circuit RAM 4027	
C1,3.	Capacitor 10µF, 25V, Fitco.	
C12-18,22.	Capacitor 1µF, 50V.	
C26-33,35-41,43,46-48.	Capacitor 10nF, 40V.	
C1,4-8.	Capacitor 10µF, 16V, Alusec.	
C49.	Capacitor 0.1µF, 40V, ±1%.	
C34,45.	Capacitor 47pF, 100V.	
C42.	Capacitor 82pF, 100V.	
C44.	Capacitor 220pF, 63V, ±2%, ceramic.	
R25	Resistor 330Ω 0.25W, ±5%, CR25	
R1,5,15,16,22.	Resistor 470Ω, 0.25W, ±5%, CR25.	
R3,4,6-13,17.	Resistor 1KΩ, 0.25W, ±5%, CR25.	
R2.	Resistor 1.5KΩ, 0.25W, ±5%, CR25.	
R14.	Resistor 820Ω, 0.25W ±5%, CR25.	
R18.	Resistor 178Ω, 0.125W, ±1%, MR25.	
Q1.	Transistor BSX20.	
GR1,2,4.	Diode BAX13.	
GR3.	Diode BZY79.	

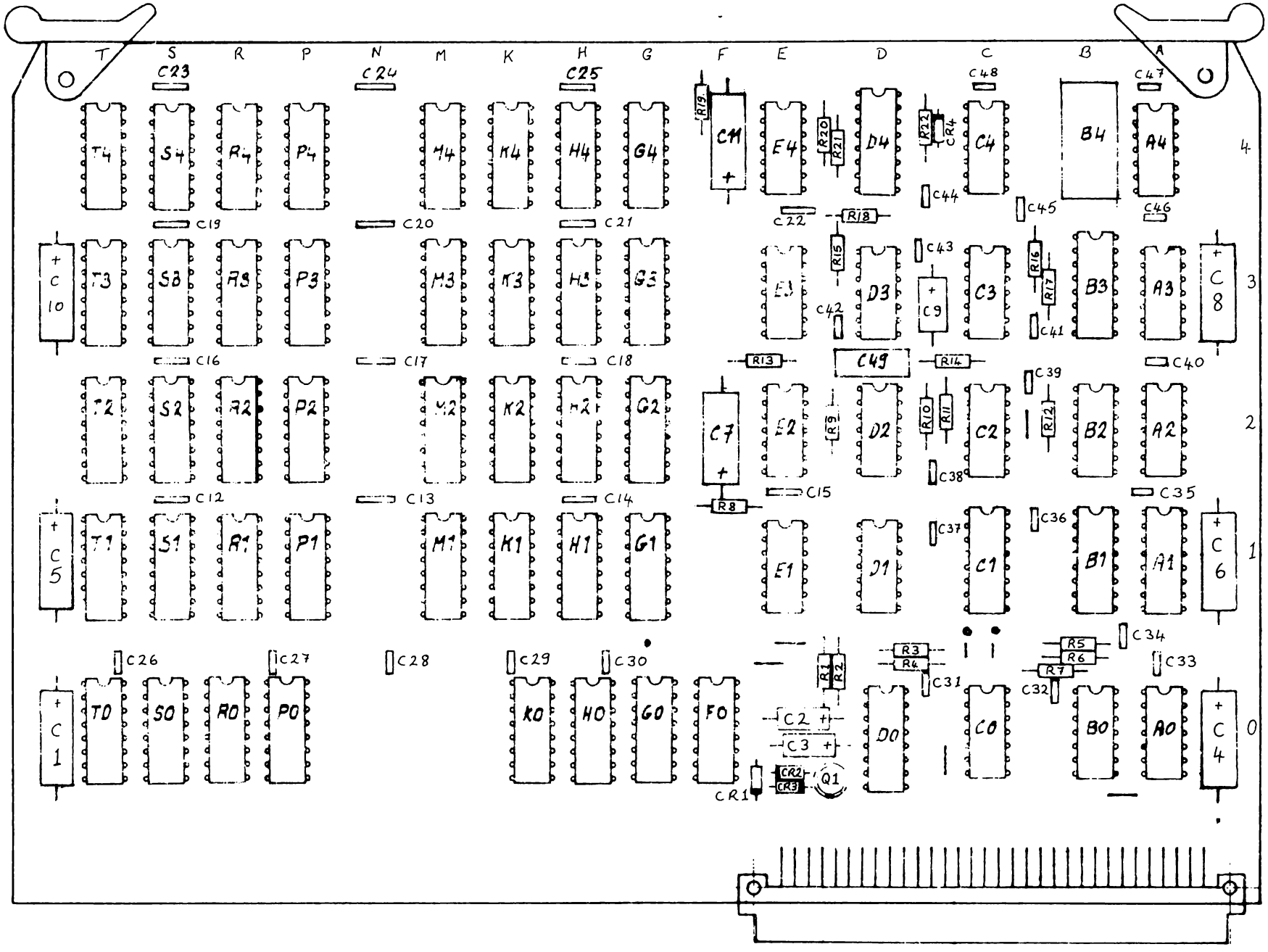


Figure 8-8 8K Memory Card (RAMDZ8 — 16 Bits) Component Layout

Table 8-4 8K Memory Card (RAMDZ8 — 16 Bits) Parts List

Reference	Description	12NC Code
	Printed circuit	5111 100 06183
E2, E4.	Integrated circuit 74S00.	
C4.	Integrated circuit 74LS00.	
A4.	Integrated circuit 74S02.	
E1, E3.	Integrated circuit 74S04.	
C3.	Integrated circuit 74LS04.	
A3.	Integrated circuit 74S10.	
D1.	Integrated circuit 74S20.	
D2.	Integrated circuit 74LS32.	
A0.	Integrated circuit 7438.	
C2.	Integrated circuit 74S74.	
B3.	Integrated circuit 74S112.	
D4.	Integrated circuit 74LS112.	
D3.	Integrated circuit 74LS132.	
C0.	Integrated circuit 74LS136.	
A1, B1, C1.	Integrated circuit 74LS153.	
G0, K0, R0, T0.	Integrated circuit 74LS157.	
D0.	Integrated circuit 74S175.	
A2, B2.	Integrated circuit 74LS197.	
F0. H0, P0, S0.	Integrated circuit 8838.	
B0.	Integrated circuit REC 0613.	
B4.	Integrated circuit LDM 'CC.	
G1-4, H1-4, K1-4, M1-4, P1-4, R1-4, S1-4, T1-4.	Integrated circuit RAM4027	
C2, 3.	Capacitor 10µF, 25V, Fitco.	
C12-25.	Capacitor 1µF, 50V.	
C26-33, 35-41, 43, 46-48.	Capacitor 10nF, 40V.	
C49.	Capacitor 0.1uF, 40V, ±1%.	
C34, 45.	Capacitor 47pF, 100V.	
C42.	Capacitor 82pF, 100V.	
C44.	Capacitor 220pF, 63V, 2%, ceramic.	
R25	Resistor 330 Ω, 0.25W, ±5% CR25	
C1, 4-8, 10, 11.	Capacitor 10µF, 16V, Alusec.	
[illegible], 5, 15, 16, 22.	Resistor 470Ω, 1/4W, ±5%.	
R3, 4, 6-13, 17, 19-21.	Resistor 1KΩ, 1/4W, ±5%.	
R2.	Resistor 1.5KΩ, 1/4W, ±5%.	
R14.	Resistor 820 Ω 0.25W, ±5%, CR25.	
R18.	Resistor 178Ω, ±1%, MR25.	
Q1.	Transistor BSX20.	
D[illegible], 2, 4.	Diode BAX13.	
GR3.	Diode BZY79 C5V1.	

SECTION IX

0.5, 1, 2K STATIC RANDOM ACCESS MEMORY

9.1 The memories use Advanced Micro Devices AM9101 BPC static random access chips that can operate in either character or word mode during a read or write cycle. These devices store data in flip-flops so no refresh operations are needed and data will be held as long as power is maintained on the chip. The chips can operate in standby mode on a much reduced power dissipation so data can easily be maintained by a battery pack during power failures. Each memory module has its own address and timing control circuits and operates in asynchronous mode. The modules have a capicity of either 0.5K, 1K, or 2K 16-bit words and although the total capacity of a P851 is 32K, only one static memory module can be used in a system and it must precede any other memory modules.

9.2 PHYSICAL

The memory modules are built on a double format Eurocard and have one 2 x 32 pin connector that plugs into the GPBS backpanel socket. Any of the sockets from 3 to 13 (2 to 13 if no IOP card is fitted) of the Eurorack may be used for the memory module, but in a mixed configuration of different types of memory modules the system rules should be followed. Figure 9-1 shows the partitioning layout of the memory card.

9.3 BLOCK DIAGRAM

Figure 9-2 shows the logic blocks of the memory module card.

9.4 MEMORY VOLTAGES

The two voltages needed by the module are provided by the power supply and are +5VL for the logic and +5VM for the memory array.

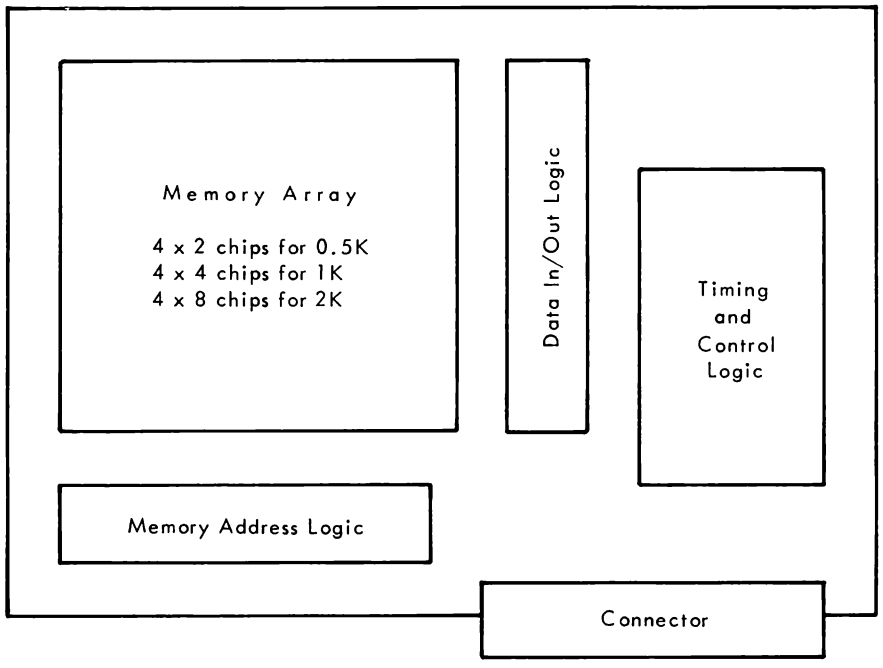


Figure 9-1 Memory Card Layout

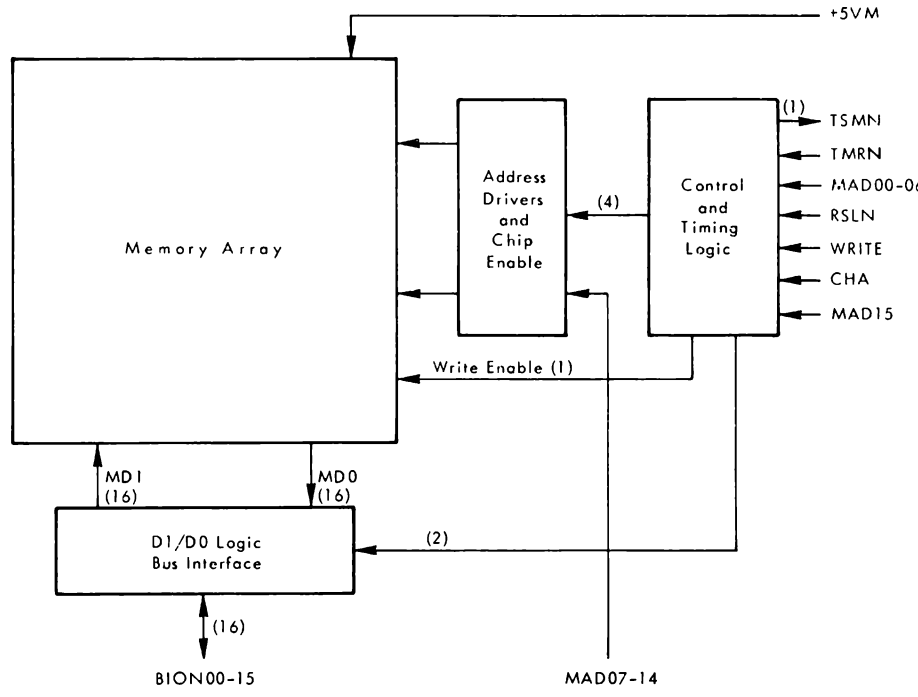


Figure 9-2 Logic Blocks of Memory Card

9.5 MEMORY ARRAY

The memory array is arranged in a matrix of AM9101BPC Static Random Access memory chips of four rows by two columns (0.5K module), four rows by four columns (1K module), or four rows by eight columns (2K module). Each chip provides 256 4-bit words and the first and third rows correspond to the left hand character (bits 00-07) and rows two and four correspond to the right hand character (bits 08-15) of the data word. Figure 9-3 shows a block diagram of one of the memory chips.

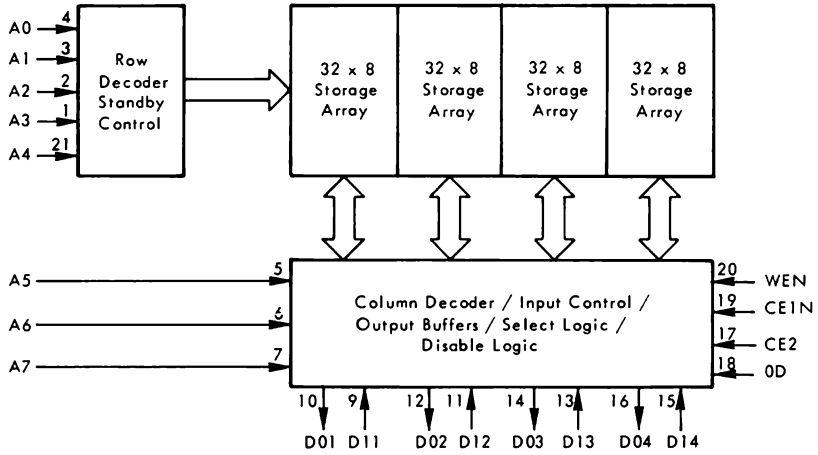


Figure 9-3 Block Diagram of Memory Chip

9.6 Bit Selection

Each chip has eight address pins to select each of the 1024 data bits (bits 0 to 1023) and the address on MAD07 to MAD14 is used to activate these pins on all the chips in the array. MAD14 to MAD10 provide the row address and MAD09 to MAD07 provide the column address for each chip. Selection of the appropriate chips of the data word is made by the input from MAD06 to MAD04 which produce the CExN signals that enable the columns of the memory array.

9.7 Data In/Out

The signals controlling the Read/Write operations are WLN and WRN which when high enable read operations and when low enable write operations. For word mode both signals are present but for character mode only the appropriate signal will be present, WLN for left character and WRN for right character.

9.8 ADDRESSING LOGIC

The logic is shown on Figure 9-8 and it is activated by the CPU. Gates F0 and H0 drive one half of the memory array chips and gates M0 and P0 drive the other half both providing the eight address bits needed by the chips. The appropriate columns of the array are enabled by K0 and A1 which are also used for module addressing. Validation of the address and the start of a memory cycle is by the signal TMRN from the CPU.

9.9 Module Addressing

This addressing is enabled by four switches SWA, SWB, SWC, and SWD and allows K0 to select one of the eight possible columns in the memory array. The following table shows the configuration of the switches for the module selection.

	SWA		SWB		SWC		SWD	
	1	2	1	2	1	2	1	2
0.5K		X		X	X		X	
1K		X	X		X			X
2K	X		X			X		X

9.10 CONTROL LOGIC

This logic controls the modes of operation of the memory (see Figure 9-8). The modes of operation are Read/Write and word or character and both are initiated by the TMRN signal from the CPU.

9.11 Read/Write

A read or write cycle is determined by the state of the WRITE signal from the CPU. When it is high a write cycle is performed. Gates C1, C3 and B2 provide the W signal to the Bus Transceivers, and the WLN and WRN signals to the memory array.

9.12 Word/Character Mode

The memory will operate in word mode unless the CHA signal from the CPU is high. In character mode gates C1, B2, C3 and A2 provide the ELCH or ERCH signals that enable the WLN and WRN signals to the memory array. Selection of left or right character is determined by the state of MAD15: when low the left character is selected and when high the right character is selected. The output from A2 is used to enable the data multiplexers.

9.13 TIMING LOGIC

This logic is shown on Figure 9-8 and comprises an oscillator (made with a 250nS delay line A3) and two flip-flops C2.

9.14 Start Access Time

When the MSBN signal from A1 goes low it clocks flip-flop C2 and the output from pin 9 is used to start the oscillator A3. The different edges of the oscillator provide all the other access timing signals.

9.15 Delay Write Enable

The S1 and S2 outputs from A3 are used via B2 pin 10 to inhibit the write enable signal until the end of memory access time.

9.16 Memory to CPU Clock

This is the TSMN signal produced by flip-flop C2 when it is clocked by the S4 output from A3 going low. It is produced at the end of addess time and signals the CPU that it is ready for a data transfer.

9.17 Stop Internal Clock

The internal timing is only needed during access time so when the S10 output from A3 goes high it resets the oscillator input flip-flop C2 via B2. The cycle time of the memory is controlled by the CPU using the TSMN and TMRN signals.

9.18 Timing Diagram

Figure 9-4 shows the timing for Read/Write cycles. The times shown on the diagram are typical values only, the actual times may vary by ±100nS.

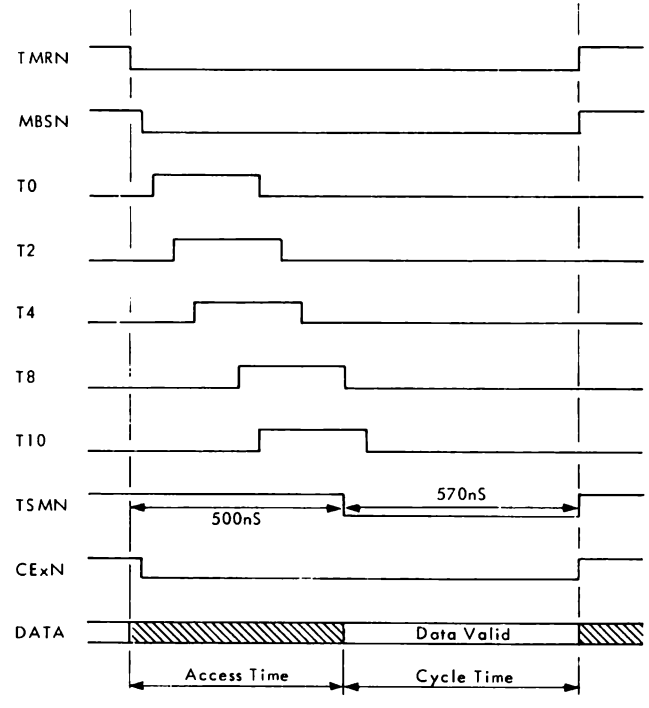


Figure 9-4 Read/Write Cycle Timing

9.19 DATA IN/OUT LOGIC

This logic comprises four data selector/multiplexor chips E1, E2, E3 and E4 used to select the correct data path into or out of the memory array during data transfers (see Figure 9-8).

9.20 Word Handling

In this mode ERCH produced by A2 pin 12 will be high for both Read and Write operations enabling the B inputs to the multiplexers. For Read operations this allows the left hand character into memory via E1 and E3 (the right hand character comes direct from the Bus Transceivers). For Write operations the right hand character is enabled out of memory via E2 and E4 (the left hand character is allowed direct to the Bus Transceivers). There is no reversal of character position in either of these operations.

9.21 Character Handling

In this mode the CHA input is high and the MAD15 input is either low (left hand character) or high (right hand character) producing either ELCHN or ERCHN from gates B2 pins 4 and 1 respectively. In left hand character operations ERCH from A2 pin 12 will be low enabling the A inputs to the multiplexers. So for Read Mode, bits 00-07 from the memory array will be enabled onto bits 08 to 15 of the BIO lines via E2, E4 and the Bus Transceivers. For Write Mode, bits 08 to 15 of the BIO lines will be enabled to bits 00-07 of the memory array via the Bus Transceivers, E1 and E3. In right hand character operations ERCH from A2 pin 12 will be high enabling the B inputs to the multiplexers. So for Read Mode, bits 08 to 15 from the memory array will be enabled onto bits 08 to 15 of the BIO lines via E2, E4 and the Bus Transceivers. For Write Mode, bits 08 to 15 of the BIO lines will be enabled to bits 08 to 15 of the memory array via the Bus Transceivers, E1 and E3.

9.22 BUS TRANSCEIVERS

This logic is shown on Figure 9-8 and comprises chips D1, D2, D3 and D4 which provide the interface between the BION lines of the GPBS and the memory data logic. When a valid address is recognised by the memory, MBSN goes low (for both Read and Write) and is used by one of the data select pins of the transceivers (and to enable the memory stack address). For a write operation the WRITE signal from the CPU will be high producing W (high) via A2 pin 2 which is used by the other data select pin of the transceivers and enables data on the BION lines to be written into memory. For read operations WRITE and W will be low and will enable data from the memory onto the BION lines.

9.23 MEMORY INHIBIT

This logic is shown on Figure 9-5 and comprises the data on the MAD00 to MAD03 lines and gates B0 and A1. Any memory address higher than 2K will have a one bit on one or more of these lines maintaining the output from A1 (MBSN) high. This inhibits the column address logic K0 and any output from the memory.

9.24 RESET LOGIC

This logic is shown on Figure 9-5. At power ON time RSLN goes high, enables all the memory chips and resets the internal clock via A2 pin 6.

9.25 INTERFACE CONNECTIONS

Table 9-1 gives interface connections of the memory card connector socket.

9.26 LOGIC DIAGRAM, CARDS, and CARD COMPONENTS

The logic diagram of the memory card is shown in Figure 9-5 and was derived from the E.D.S. dated 761117. The card layout for the 0.5K module is shown in Figure 9-6, the layout for the 1K module is shown in Figure 9-7, and the layout for the 2K module is shown in Figure 9-8. Table 9-2 lists the components of the 0.5K module, Table 9-3 lists the components of the 1K module, and Table 9-4 lists the components of the 2K module.

Table 9-1 Interface Connections

Pin Number	Signal Name
1C01	
1C02	+5VM
1C03	
1C04	
1C05	
1C06	0V
1C07	
1C08	
1C09	0V
1C10	
1C11	
1C12	0V
1C13	MAD00
1C14	MAD02
1C15	MAD04
1C16	MAD06
1C17	MAD08
1C18	MAD10
1C19	MAD12
1C20	MAD14
1C21	CHA
1C22	0V
1C23	BIO08N
1C24	BIO09N
1C25	BIO10N
1C26	BIO11N
1C27	BIO12N
1C28	BIO13N
1C29	BIO14N
1C30	BIO15N
1C31	
1C32	+5VL

Pin Number	Signal Name
1A01	
1A02	+5VM
1A03	
1A04	
1A05	
1A06	0V
1A07	RSLN
1A08	
1A09	0V
1A10	TSMN
1A11	TMRN
1A12	0V
1A13	MAD01
1A14	MAD03
1A15	MAD05
1A16	MAD07
1A17	MAD09
1A18	MAD11
1A19	MAD13
1A20	MAD15
1A21	WRITE
1A22	0V
1A23	BIO00N
1A24	BIO01N
1A25	BIO02N
1A26	BIO03N
1A27	BIO04N
1A28	BIO05N
1A29	BIO06N
1A30	BIO07N
1A31	
1A32	+5VL

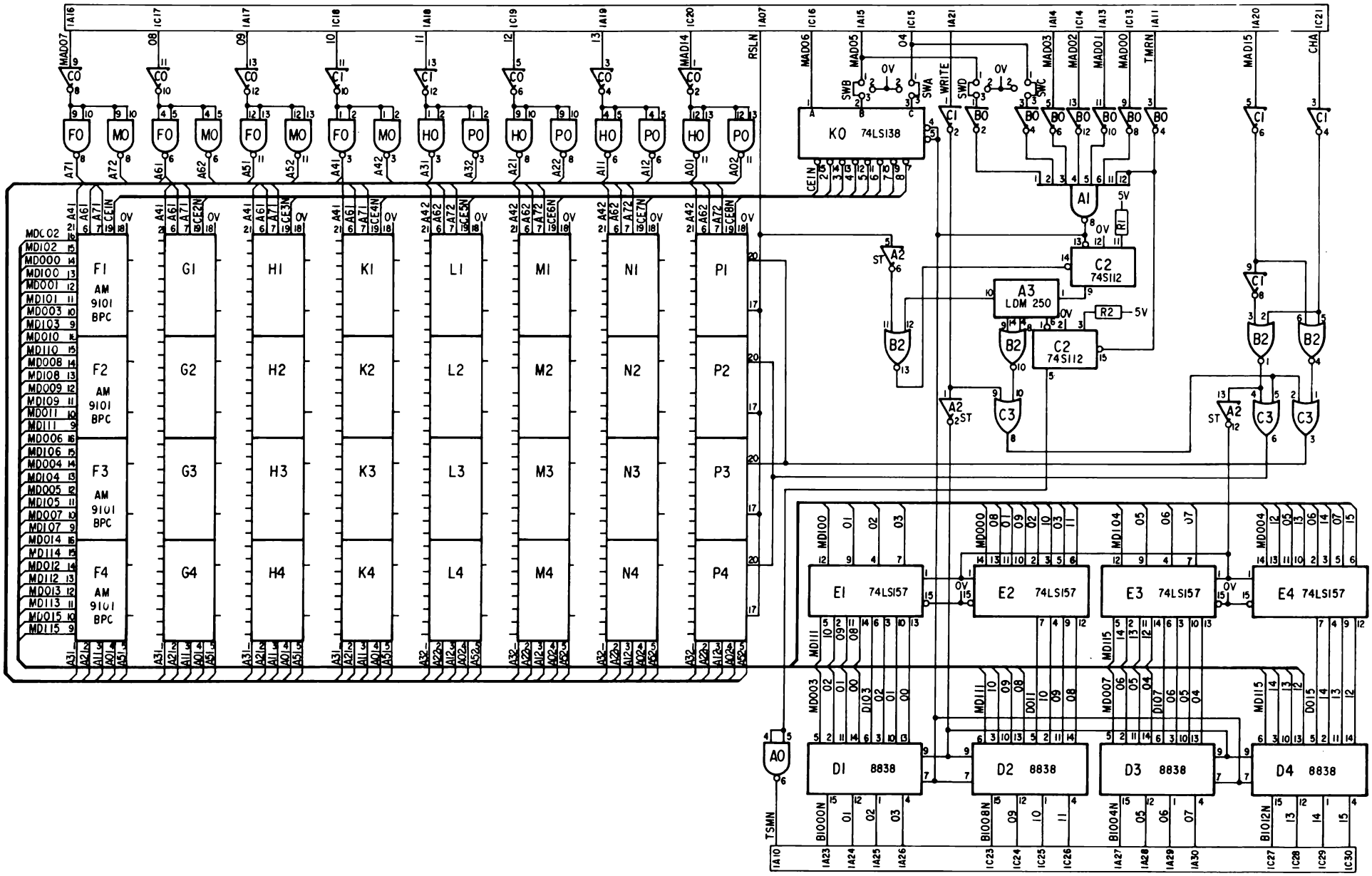


Figure 9-5 Static RAM Logic Diagram

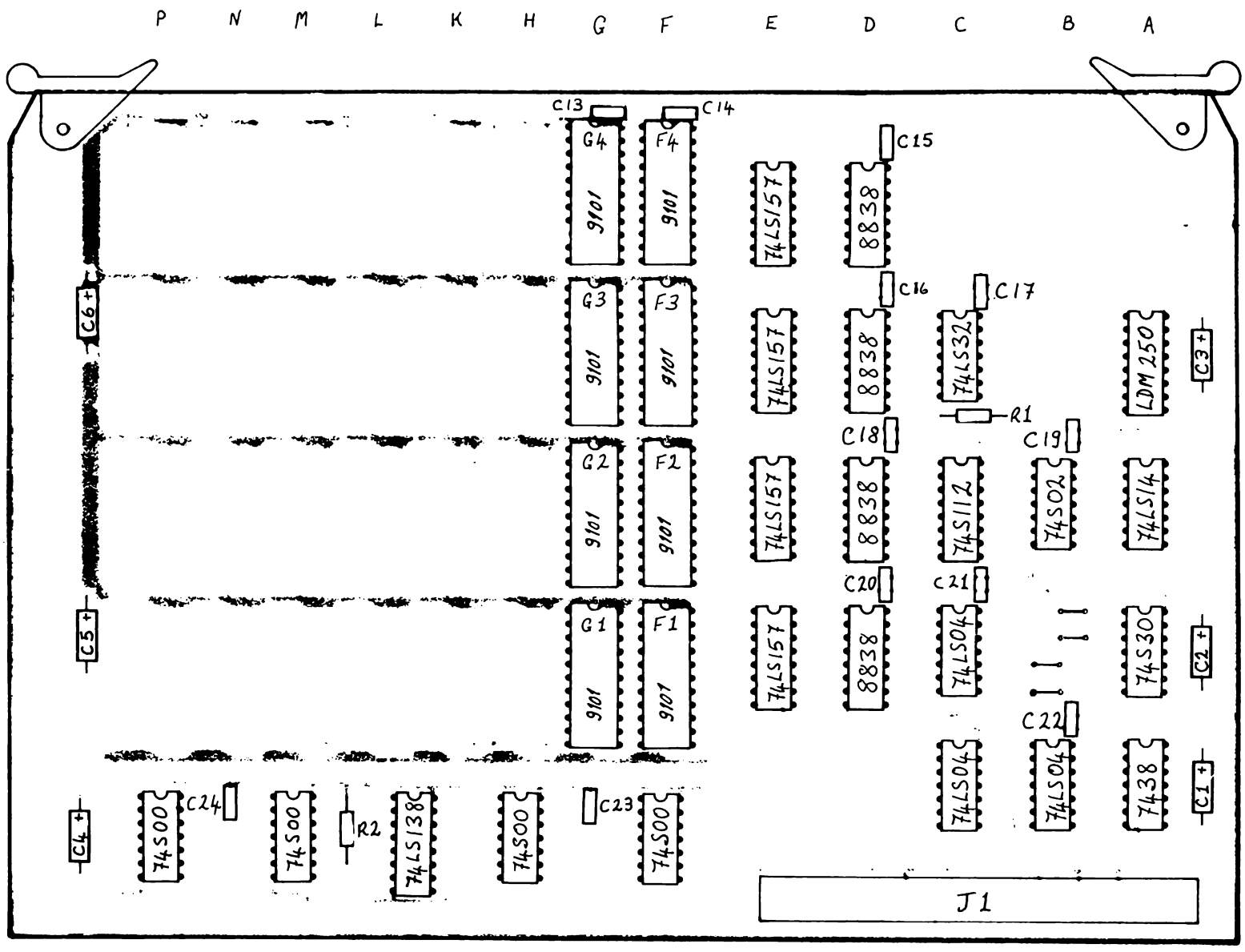


Figure 9-6 0.5K Memory Component Layout

Table 9-2 0.5K Memory Parts List

Reference	Description	12NC Code
	Printed circuit	5111 100 06371
F0,H0,M0,P0.	Integrated circuit 74S00.	
B2.	Integrated circuit 74S02.	
B0,C0,C1.	Integrated circuit 74LS04.	
A2.	Integrated circuit 74LS14.	
A1.	Integrated circuit 74S30.	
C3.	Integrated circuit 74LS32.	
A0.	Integrated circuit 7438.	
C2.	Integrated circuit 74S112.	
K0.	Integrated circuit 74LS138.	
E1,E2,E3,E4.	Integrated circuit 74LS157.	
D1,D2,D3,D4.	Integrated circuit 8838.	
A3.	Integrated circuit LDM 250.	
F1-4,G1-4.	Integrated circuit AM9101 BPC.	
C1-6.	Capacitor 10µF, 25V, Fitco.	
C13,14.	Capacitor 0.1µF, 50V, 2C067.	
C15-24.	Capacitor 10nF, 40V.	
R1,2.	Resistor 1Kn, 1/4W, ±5%, CR25.	
J1.	Connector 100 964 053.	

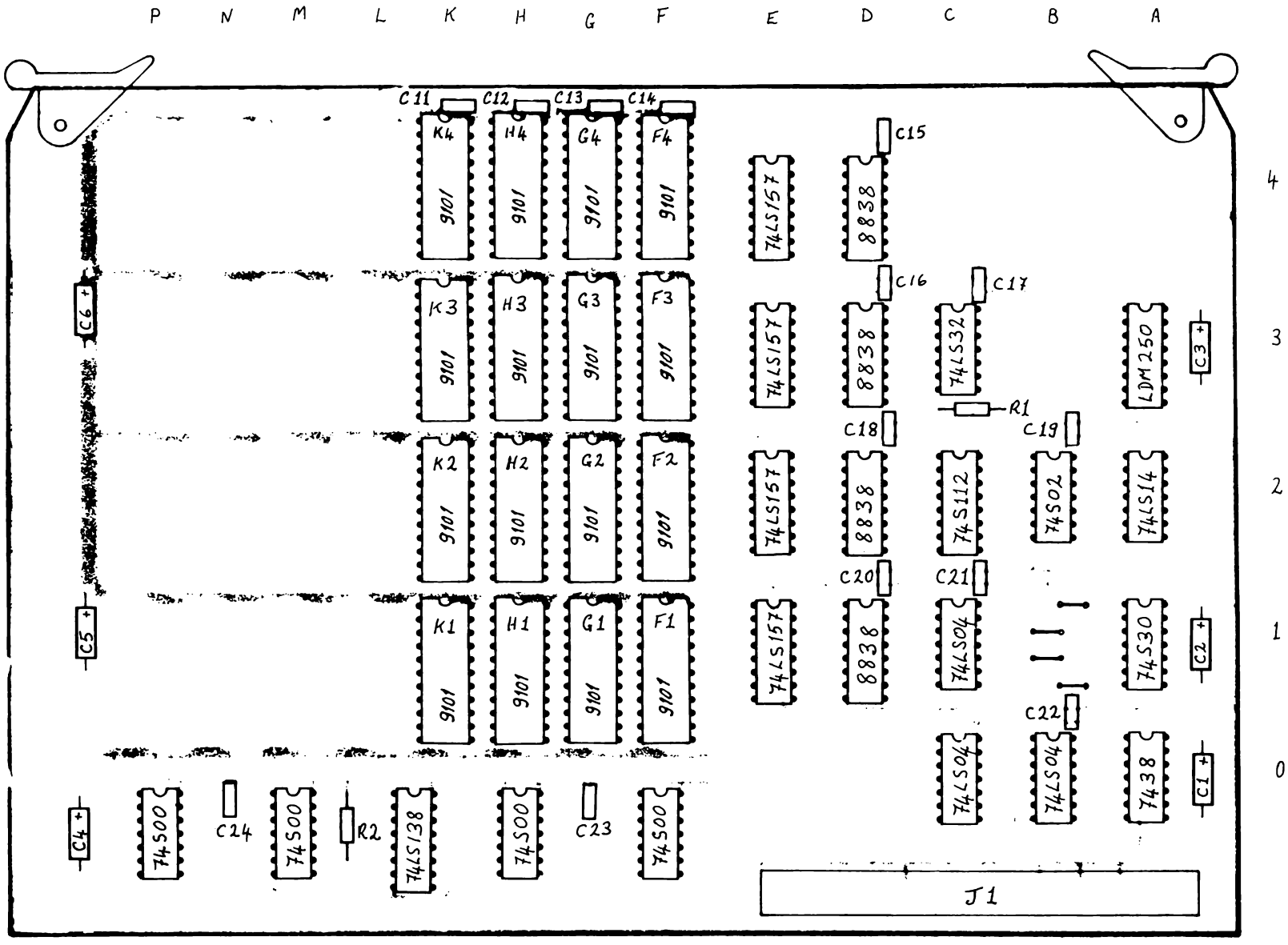


Figure 9-7 1K Memory Component Layout

Table 9-3 1K Memory Parts List

Reference	Description	12NC Code
	Printed circuit	5111 100 06371
F0, H0, M0, P0.	Integrated circuit 74S00.	
B2.	Integrated circuit 74S02.	
B0, C0, C1.	Integrated circuit 74LS04.	
A2.	Integrated circuit 74LS14.	
A1.	Integrated circuit 74S30.	
C3.	Integrated circuit 74LS32.	
A0.	Integrated circuit 7438.	
C2.	Integrated circuit 74S112.	
K0.	Integrated circuit 74LS138.	
E1, E2, E3, E4.	Integrated circuit 74LS157.	
D1, D2, D3, D4.	Integrated circuit 8838.	
A3.	Integrated circuit LDM 250.	
F1-4, G1-4, H1-4, K1-4.	Integrated circuit AM9101 BPC.	
C1-6.	Capacitor 10µF, 25V, Fitco.	
C11-14.	Capacitor 0.1µF, 50V, 2C067.	
C15-24.	Capacitor 10nF, 40V.	
R1, 2.	Resistor 1Kn, 1/4W, ±5%, CR25.	
J1.	Connector 100 964 053.	

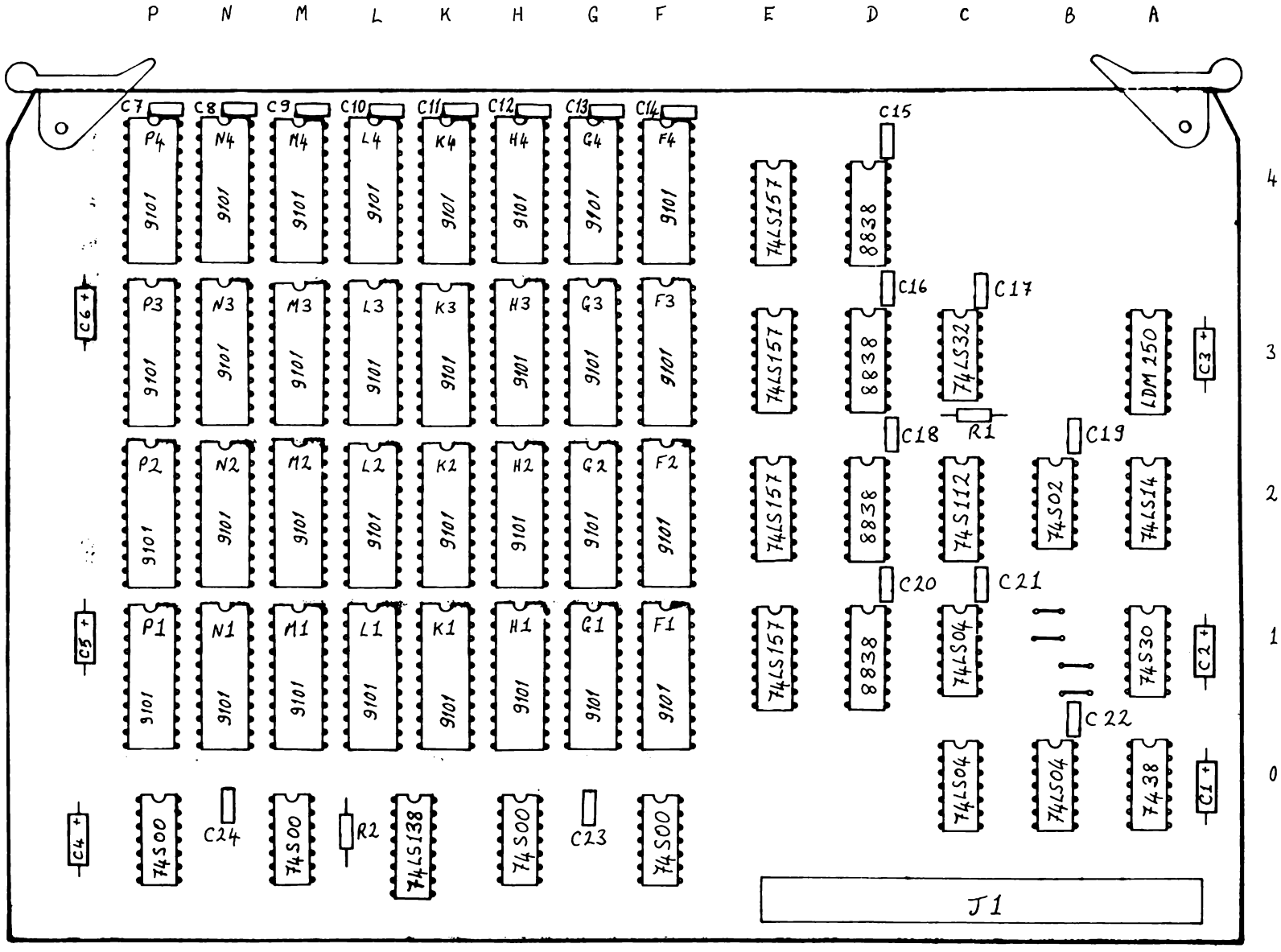


Figure 9-8 2K Memory Component Layout

Table 9-4 2K Memory Parts List

Reference	Description	12NC Code
	Printed circuit	5111 100 06371
F0, H0, M0, P0.	Integrated circuit 74S00.	
B2.	Integrated circuit 74S02.	
B0, C0, C1.	Integrated circuit 74LS04.	
A2.	Integrated circuit 74LS14.	
A1.	Integrated circuit 74S30.	
C3.	Integrated circuit 74LS30.	
A0.	Integrated circuit 74LS32.	
C2.	Integrated circuit 7438.	
K0.	Integrated circuit 74S112.	
E1, E2, E3, E4.	Integrated circuit 74LS157.	
D1, D2, D3, D4.	Integrated circuit 8838.	
A3.	Integrated circuit LDM 250.	
F1-4, G1-4, H1-4, K1-4, L1-4, M1-4, N1-4, P1-4.	Integrated circuit AM9101 BPC.	
C1-6.	Capacitor 10µF, 25V, Fitco.	
C7-14.	Capacitor 0.1µF, 50V, 2C067.	
C15-24.	Capacitor 10nF, 40V.	
R1, 2.	Resistor 1Kn, 1/4W, +5%, CR25.	
J1.	Connector 100 964 053.	

SECTION X

16k - 32k DYNAMIC RANDOM ACCESS MEMORY

10.1 INTRODUCTION

This section describes the MOS dynamic RAM memory used as the main memory for the P851 CPU. The capacity of the memory module is 32k 16-bit words but a 16k version of the module is obtained by depopulation. The memory array is composed of MOSTEK MK4116P-3 chips, or equivalent, which are random access devices of 16384 bits each. The array is arranged in two blocks of 16 chips with block 2 omitted for the 16k version. Linking on the card caters for one or two blocks in operation. During a read or write cycle these chips operate either in word mode or character mode, where a character is defined as an eight-bit byte. The module provides its own integral refresh circuits and timing control circuits for the array and operates in asynchronous mode during data exchanges with the CPU. Multiple-card memories can be built (up to eight cards) to give a total storage of up to 128k words only.

10.2 CONSTRUCTION

The memory module is built on a single card of double-Eurocard format that plugs into one normal card slot of the basic central processor rack. Connections to the simplified general purpose bus and power supplies are made via a single 2x32 pin connector type F068. Figure 10-1 shows the layout for the various areas of logic on the card.

10.3 VOLTAGES

The memory module requires four operating voltages, namely:

- +5VL
- +5VM
- -5VM
- +12VM

The +5VL, +5VM, and +12VM supplies are provided externally by the P851 power supply card. The -5VM supply is produced internally on the card by a voltage regulator circuit, which operates from the +12VM supply. The +5VM and +12VM supplies have the battery back-up option and therefore so does the -5VM supply as it is derived from +12VM.

10.4 FUNCTIONAL SCHEMATIC DIAGRAM

Figure 10-2 shows the main groups of logic that form the memory module. The logic comprises:

- Oscillator producing clock pulses
- Memory array
- Cycle timing counter
- Refresh cycle counter
- Command decoder
- Refresh address counter
- Memory address selector
- Bus switching and interface logic

10.5 The logic operates in two separate modes:

- Memory cycle — where the logic performs a read/write function on the array at memory address selected by MAD lines 00 to 14 and accompanied by the appropriate command.
- Refresh cycle — where the logic periodically generates the next in a series of sequential memory addresses and refreshes the data stored in the row at that location.

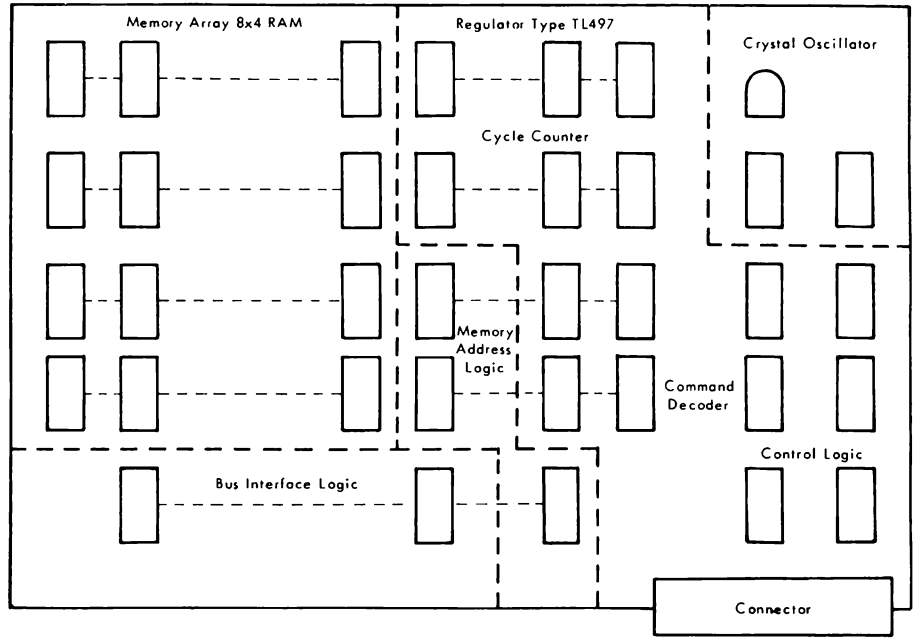


Figure 10-1 Memory Card Layout

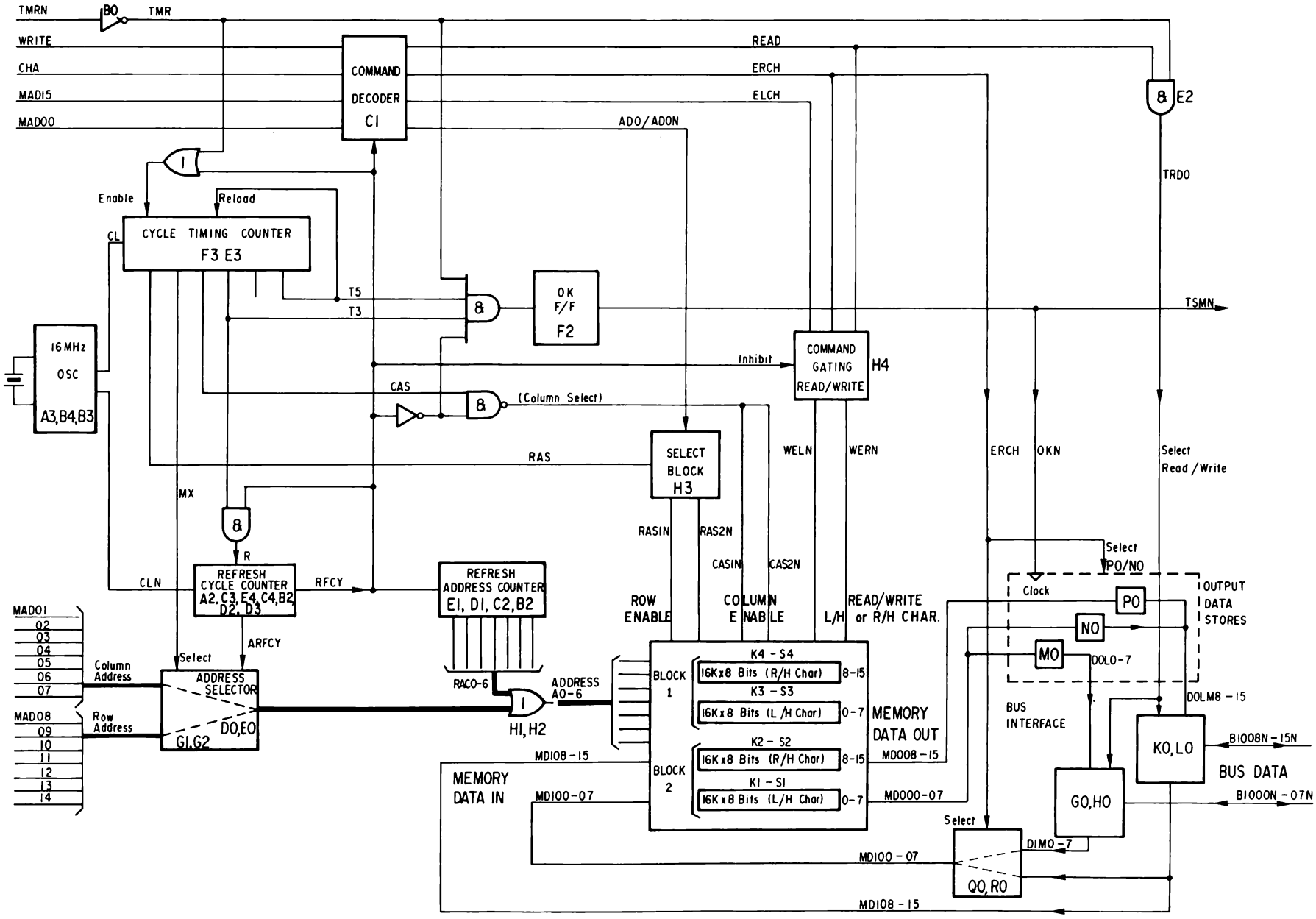


Figure 10-2 Functional Schematic Diagram of 16k - 32k RAM Memory

10.6 Memory Cycle

The logic commences a memory cycle on receiving a read/write command from the CPU. This command is defined by the WRITE, CHA, and MAD15 lines. It is accompanied by bus timing signal TMRN and the memory address on lines MAD00-14. Pulse TMR starts the 'cycle timing' counter, which is clocked by the oscillator. The counter outputs are sequential timing pulses that control the logic both for a memory cycle or a refresh cycle.

10.7 Addressing. MAD00 defines block 1 or 2 of the array and selects 'row address' strobe RAS1N or RAS2N via the command decoder. At the address selector, MX initially selects the row address (on MAD08-14), which is strobed into the array chips by the 'row address' strobe. As the cycle progresses, MX switches the address selector to the column address (on MAD00-07), which is strobed into the array by the 'column address' strobe (CAS1N, CAS2N).

10.8 Command Decoding. Meanwhile the command (read/write) and mode (word/character) are analysed by the command decoder and signals READ, ELCH, and ERCH set accordingly. Bus timing signal TSMN is generated in response to TMRN after a delay introduced by combined 'cycle timing' outputs T3 and T5. READ is enabled by a 'read' command; ELCH or ERCH is enabled for left or right character mode respectively. For word mode ELCH and ERCH are both enabled.

10.9 Read Memory to Bus. WELN and WERN are both disabled. The memory data from the selected location is clocked into the 'output data' store by OKN, which accompanies the TSMN pulse. TRDO is enabled by the 'read' command and enables the data path from the 'output data' store to the data bus lines BIO00-15N.

10.10 Write Bus to Memory. TRDO is disabled by the 'write' command and the bus data is routed through the interface logic to the selected location in the array. The memory data is strobed into the array chip by WELN and/or WERN, which are the 'write' signals for left and right characters respectively.

10.11 Character Mode. In character mode, only bus data lines BIO08-15N are valid. ERCH is enabled for right hand character mode and the data paths to and from memory remain the same as for word mode (viz. MDO08-15 to BIO08-15). ERCH is disabled for left hand character mode and switches the interface logic such that the left hand characters in memory are routed to and from the right hand data bus lines (viz. MDO00-07 to BIO08-15N to MDI00-07).

10.12 Refresh Cycle

The logic commences a refresh cycle when the 'refresh cycle' counter completes its count and generates the 'refresh' signal RFCY. RFCY starts the 'cycle timing' counter, which behaves in an identical manner to its memory cycle operation. RFCY also enables both 'row address' strobes (RAS1N, RAS2N) via the command decoder and inhibits CAS1N and CAS2N. The 'refresh address' counter is incremented by RFCY and generates the next row address in the sequence. This address is strobed into both blocks of array chips by RAS1N/RAS2N and the data contained in all rows having that address is refreshed. If a read/write command is received during this time, RFCY delays TSMN and WELN/WERN until completion of the refresh cycle.

10.13 LOGIC DESCRIPTION

The following description should be read against the logic diagrams Figures 10-8a,b. Throughout the descriptions and drawings, all signal names ending in N are active low. In addition, binary '1' always refers to a high (positive) level.

10.14 Operation of a 16k MOS Dynamic RAM Chip

Figure 10-3 shows a block diagram of one of the memory chips comprising the array. Each chip is a 16k one-bit read/write dynamic RAM. Its internal storage array of 16384 bits is arranged for addressing purposes in 128 columns of 128 rows. Consequently 14 address bits are required to select any one of bits 0 to 16383, but the chip provides only seven address pins (lines A0 to A6). The logic sends first the row address of the chip accompanied by 'row address' strobe RASN (low). It then sends the column address of the chip accompanied by 'column address' strobe CASN (low). The read/write mode of the chip is

determined by the 'write' input (pin 3) of the chip, which is controlled by WELN for chips containing the left hand characters and WERN for right hand characters.

10.15 For refresh cycles, WELN, WERN, and CASN are all disabled (high) and one row of 128 bits in every chip of the array is refreshed simultaneously. The selected row is determined by the refresh address. Note that during a normal read or write cycle the other locations sharing the same row address are refreshed automatically. The data outputs (pin 14) of the chips are switched open-circuit at the end of a memory cycle when CASN goes high. Invalid data is thus prevented from reaching the bus logic.

10.16 Addressing Logic

The addressing logic is activated either by a memory cycle initiated by the CPU or by a refresh cycle initiated by the memory logic.

10.17 Memory Cycle Addressing. For a memory cycle the address within the array is given on the 14 memory address lines MAD01-14, which are applied to multiplexor chips E0, D0. First the row address (MAD01-07) is sent to lines A0-6 followed by the column address (MAD08-14), as selected by MX from the 'cycle timing' counter. MX switches from low to high at the multiplexor, which is enabled when ARFCY is low. During a refresh cycle ARFCY is high and inhibits the multiplexor chips D0 and E0, but ARFCYN is low and enables chips G1 and G2 for the refresh address (RAC0-6).

10.18 Refresh Cycle Addressing. Each memory location must be refreshed at intervals of 2mS otherwise the stored data will be lost. To do this a series of 128 refresh cycles is carried out over a period of 2mS, with one row address activated in all chips simultaneously for each cycle. The refresh cycles are carried out at intervals of 15µS, as determined by the 'refresh cycle' counter (A2, C3, E4, C4, B2, D2, D3), which is driven by clock pulses from the 16MHz crystal oscillator. The 'refresh' signal RFCY from the counter increments the 'refresh address' counter (E1, D1, C2, B2), which sets up the next row address

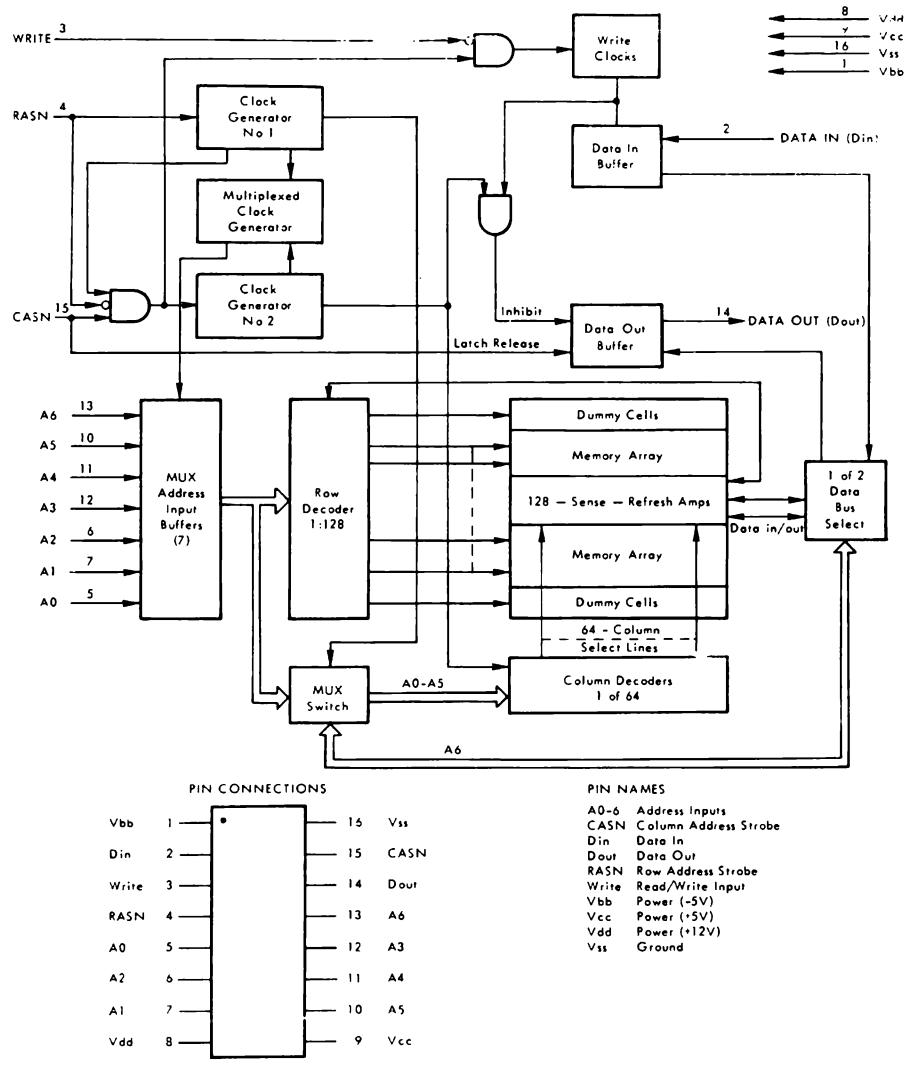


Figure 10-3 Functional Block Diagram of a 16k Dynamic RAM Chip

to be refreshed. This signal also switches the control logic to refresh cycle operation, provided that a memory cycle is not in progress. The refresh address (RAC0-6) is applied to multiplexor chips G1, G2 of the address selector. This is a row address only and is then sent to lines A0-6 while 'refresh' signal ARFCYN is enabled (low).

10.19 Memory Module Addressing. The linking for the 16k and 32k versions of the memory card is given in Figure 10-4. In addition, this diagram shows the extra linking required for operation of the module in a multiple-card (8-card) memory configuration, which can store up to a maximum of 128k words. Module selection is made by additional memory extension address lines MADX0-2. The links are situated physically at position A0 on the card.

10.20 Address lines MADX0-2 are applied to Exclusive Or gates A0. A fourth gate is used for the MAD00 'block selection' input, when applicable. The outputs of the four gates are arranged to all be high for a valid module address by selection of the polarity at links C to F. The 'card select' signal CSV (high) then enables gate E2 for incoming read/write request cycles.

10.21 Timing Logic

The timing logic consists of an oscillator to provide clock pulses that drive the 'cycle timing' counter and 'refresh cycle' counter, together with gating that controls the bus interface logic and TSMN timing signal.

10.22 Oscillator. The oscillator logic is formed by And-gates B3, Nand-gates A3 and associated components. The frequency is determined by crystal X1 at 16MHz, giving complementary clock outputs CL and CLN with a period of 62.5nS.

10.23 Cycle Timing Counter. The counter comprises two series-connected shift register chips (F3,E3), which are driven by the CL clock pulses. These chips are effectively inactive when 'cycle request' signal CYREQ is low because a train of zeros is clocked through the registers. CYREQ is set to '1' (high)

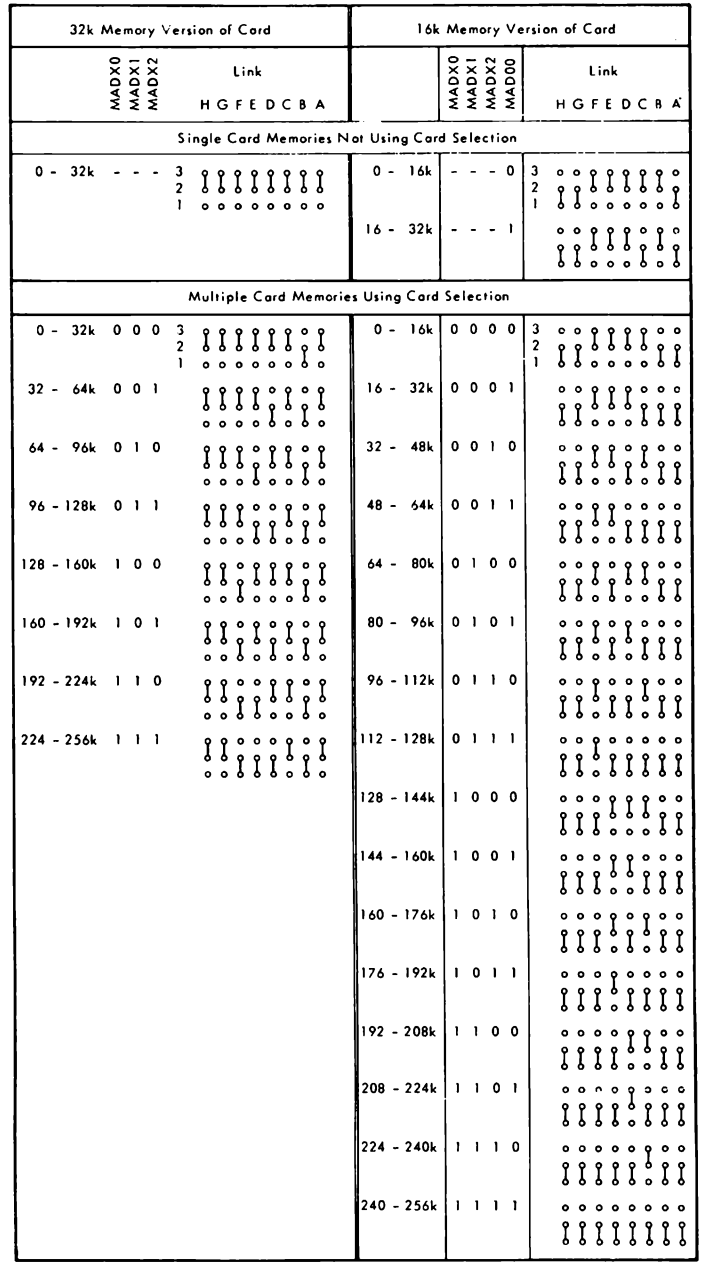


Figure 10-4 16k/32k Memory Card Linking Diagram

when gate E2 receives either:

- Read/write command from the CPU — RWCYN goes low.
- Refresh cycle start signal — RFCYN goes low.

10.24 When CYREQ goes high a '1' is clocked through registers F3 and E3 and outputs RAS, MX, CAS, T3, T4, and T5 are set high by consecutive CL clock pulses (refer to Figure 10-5). T5 reloads register F3 with zeros with the exception of CAS, which assumes the current state (high) of MX until the following clock pulse. T4 and T5 then also reset to zero on successive CL clock pulses and end the cycle.

10.25 The counter outputs have the following functions:

- RAS — Enable RAS1N/RAS2N
  Enable CYBUSYN (Memory Cycle only)
- MX — Select row/column address (Memory Cycle only)
  Extend time of CAS
- CAS — Enable CAS1N/CAS2N (Memory Cycle only)
  Reset RFCY (Refresh Cycle only)
- T3 — Reset ERFRQ and ARFRQ (Refresh Cycle only)
  Drive chip E3
- T5 — Reload chip F3
  Enable OK/OKN and TSMN (Memory Cycle only)

10.26 Refresh Cycle Counter. The counter consists of 13 'JK' flip-flops (A2, C3, E4, C4, B2, D2, D3). The counter waveforms are shown in Figure 10-6. The first eight stages (A2, C3, E4, C4) are connected as divide-by-two 'toggle' stages, with the output CNT15 of gate D4 modifying the division from +256 to +240. This provides a counter output pulse RFCNT7N (low) every 15μS. The remaining stages (B2, D2, D3) of the counter are connected as D-type flip-flops to store and progress the request for the refresh cycle initiated by RFCNT7N (low).

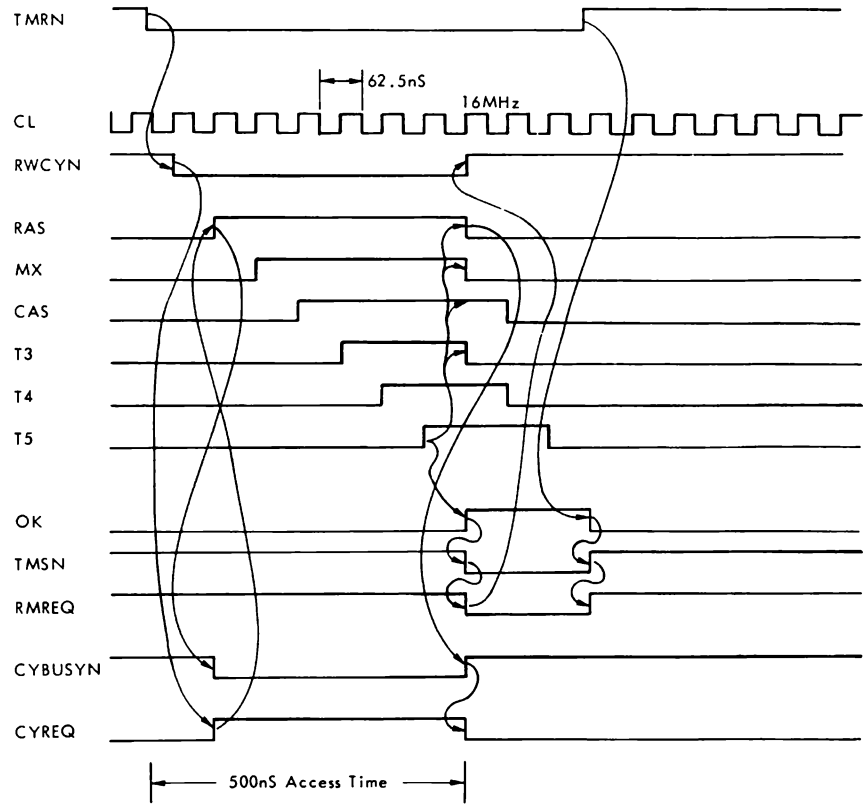


Figure 10-5 Cycle Timing Waveforms

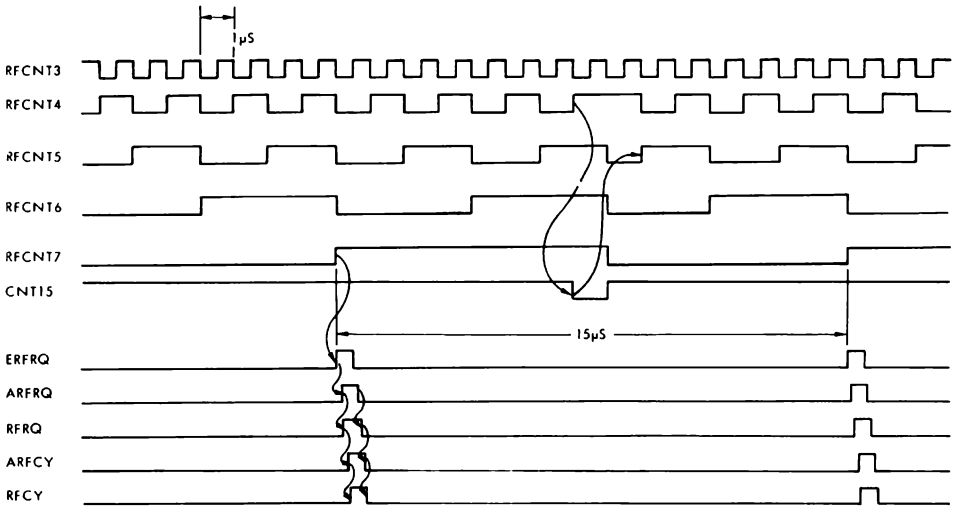


Figure 10-6a Refresh Cycle Repetition Waveforms

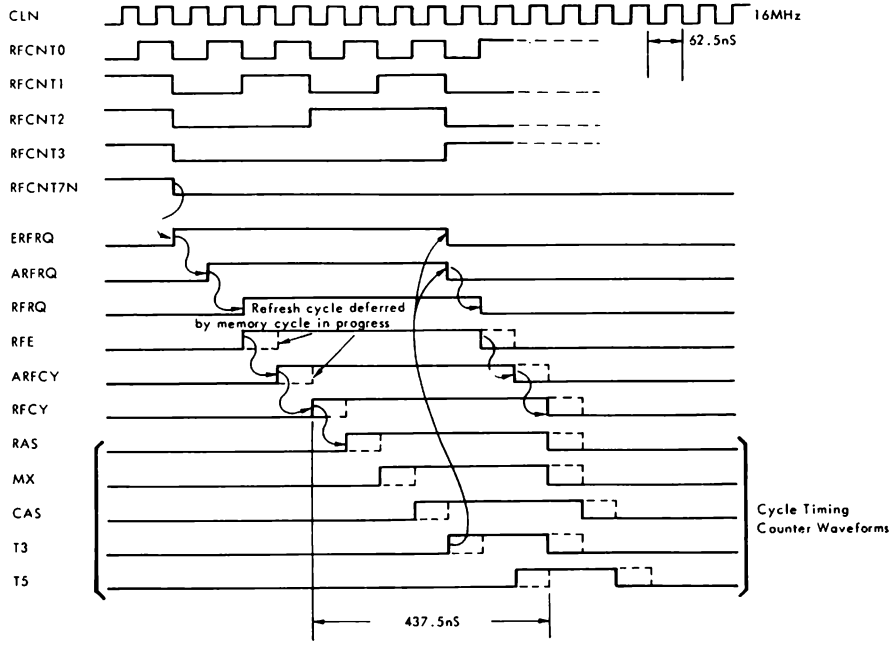


Figure 10-6b Refresh Cycle Duration Waveforms

10.27 These stages are triggered by consecutive CLN pulses and provide 'refresh request' signals ERFRQ, ARFRQ, RFRQ followed by 'refresh' signals ARFCY, RFCY, which initiate the refresh cycle. 'Refresh request' signals ARFRQ and RFRQ are reset immediately T3 goes positive, and the refresh cycle ends three CLN pulses later. Note that if a memory read/write cycle commences before ARFCY is driven to follow RFE, the refresh cycle is deferred at gate F1 until completion of the memory cycle. This does not affect the time for the next refresh cycle however, because the division section of the counter continues unaffected.

10.28 Bus Timing Logic (OK, TSMN). T5 in combination with T3 sets the OK flip-flop during a memory cycle. OKN clocks the data from addressed locations in the array into the output data stores (MO, NO, PO) of the interface logic. At the same time OK produces the TSMN bus timing signal for the CPU, which denotes 'valid data on bus' (lines BIO00-15N). TSMN is enabled 500nS (i.e. eight CL clock pulses) after the logic receives TMRN, as determined by the 'cycle timing' counter. This is the normal access time for the device and will be longer if a memory cycle is deferred as a result of a refresh cycle in progress (Figure 10-7).

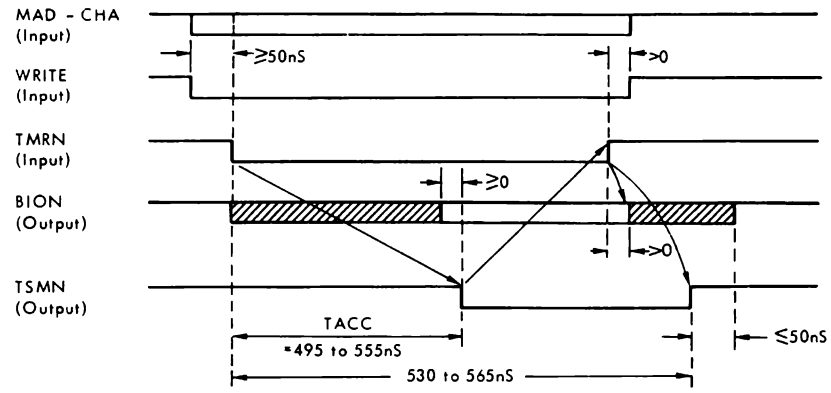
10.29 Chips MO and NO store the left hand character and chip PO stores the right hand character. TSMN also generates RMREQ, which resets the 'memory request' flip-flop F2. Output MREQ cannot be set again until the OK flip-flop is cleared by TMR (low) at the end of the current TMRN pulse.

10.30 External Cycle Control Logic

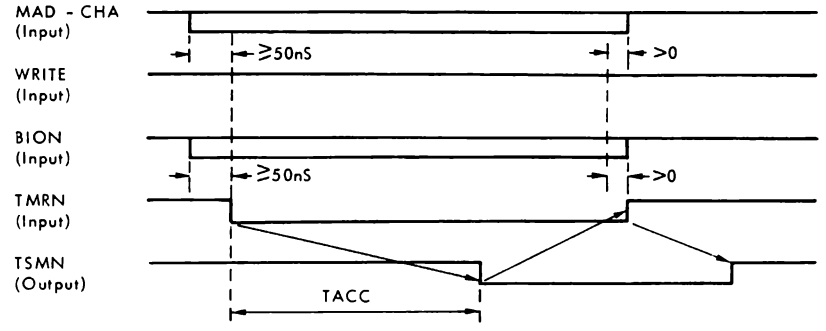
The external memory cycle is initiated when the TMRN signal is received. 'Memory request' signals MREQ and RWCYN are enabled on the next CL clock pulse provided that:

- No power failure has occurred, i.e. RSLN is high.
- No refresh cycle request is present, i.e. RFRQN is high.

If a refresh cycle is in progress, MREQ and RWCYN are deferred until after completion of the cycle. The command decoding is not delayed however, but its effects on the array are masked by the inhibiting RFCY1N signal present at the associated gates.



(a) Read Cycle — Data to Bus from Memory



(b) Write Cycle — Data to Memory from Bus

Figure 10-7 Bus Timing Waveforms

Table 10-1 Command Decoder Functions

Inputs		Outputs		Command function
RFCY	MAD00	ADO	ADON	
L	L	L	H	Block 1 selected
L	H	H	L	Block 2 selected
H	-	H	H	Both blocks selected
CHA	MAD15	ELCH	ERCH	
L	-	H	H	Word mode
H	L	H	L	Character mode, left hand
H	H	L	H	Character mode, right hand

10.31 If the module forms part of a multi-card memory, address decoding signal CSV will only be high when the module is addressed (refer to paragraph on Memory Module Addressing). RWCYN starts the 'cycle timing' counter (F3, E3), which is latched to complete a full cycle by 'cycle busy' signal CYBUSYN when RAS goes high. This holds the counter in operation if the memory cycle has been delayed so long that TMRN ends before completion of the cycle.

10.32 Command Decoder. This function is performed by decoder chip C1 in combination with Nand-gates F0, H4, and E2. Decoder C1 receives command inputs MAD00, MAD15, and CHA from the CPU and 'refresh' signal RFCY from the memory logic. It decodes these signals as shown in Table 10-1.

10.33 Nand-gates F0 and H4 receive the WRITE command input from the CPU and in combination with gate E2 interpret the signals as shown in Table 10-2.

10.34 Array Control Signals. ADO and ADON select which block of the array is addressed by enabling RAS1N or RAS2N. WELN and WERN are connected to the 'write' pin of the array chips for the left hand character and right hand character respectively and select their working mode. WELN and WERN select the write mode when low and the read mode when high.

Table 10-2 Write/Read Command Decoding

Inputs			Outputs			Command Function
WRITE	ELCH	ERCH	WELN	WERN	TRDO	
H	H	H	L	L	L	Write word
H	L	H	H	L	L	Write R/H character
H	H	L	L	H	L	Write L/H character
L	-	H	H	H	H	Read word or read R/H char.
L	-	L	H	H	H	Read L/H char.

10.35 Bus Interface Logic. TRDO is the 'transmit data' signal and controls the input/output state of the interface logic. This logic comprises two pairs of driver/receiver chips (GO,HO and KO,LO), three multiple store chips (MO, NO, PO), and data selector (QO, RO). The store chips are loaded with the memory data from the selected location by the OKN pulse and the driver/receiver chips route data on and off the bus lines (BIO00-15N).

10.36 Data Path — Write Mode. TRDO is low and blocks the DOL and DOLM inputs to the driver/receiver chips (GO,HO and KO,LO). The bus data is routed to the array inputs as follows:

- Word Mode — BIO00-07N to DIM00-07 to MD100-07, via data selector QO,RO (selected by ERCH high). BIO08-15N to MD108-15.
- Right hand character mode — As for word mode, but the CPU does not use BIO00-07N lines.
- Left hand character mode — BIO08-15N to MD108-15 to MD100-07, via data selector QO,RO (selected by ERCH low). CPU does not use BIO00-07N lines

10.37 Data Path — Read Mode. TRDO is high and enables the DOL and DOLM inputs to the driver/receiver chips (GO,HO and KO,LO). The memory output data is routed on to the bus as follows:

- Word mode — MDO00-07 to DOL00-07 to BIO00-07N, via store chip MO. MDO08-15 to DOLM08-15 to BIO08-15N, via store chip PO (enabled by ERCHN low).
- Right hand character mode — As for word mode, but the CPU ignores BIO00-07N lines.
- Left hand character mode — MDO00-07 to DOLM08-15 to BIO08-15N, via store chip NO (enabled by ERCH low). CPU ignores BIO00-07N lines.

10.38 Refresh Cycle Control Logic

The internal refresh cycle is initiated when the 'refresh request' signals (ERFRQ, ARFRQ, RFRQ) are generated by the arrival of the RFCNT7N timing pulse, which occurs at 15μS intervals. 'Refresh' signals ARFCY and RFCY are enabled on the next two CLN clock pulses provided that no external read/write cycle has occurred, i.e. RWCYN and CYBUSYN must both be high. If a read/write memory cycle is in progress RFE (and hence ARFCY, RFCY, RFCY1N) is deferred until after completion of the cycle.

10.39 RFCY starts the 'cycle timing' counter (F3, E3), which then runs through its sequence. The effects of the counter outputs are limited during a refresh cycle owing to the refresh signals (ARFCY/ARFCYN, RFCY/RFCYN, RFCY1N) acting as switching inputs to the control logic. The functions of these switching inputs are:

- ARFCY —

  Inhibit multiplexor chips DO,EO of address selector (bus memory address lines).
- ARFCYN —

  Enable multiplexor chips G1,G2 of address selector (refresh address).
- RFCY —

  Increment 'refresh address' counter.

  Switch command decoder (C1) to select both blocks of array (RAS1N and RAS2N).

  Enable reset of 'refresh request' signals ERFRQ and ARFRQ at T3 time.
- RFCYN —

  Start 'cycle timing' counter.
- RFCY1N —

  Inhibit OK signal to prevent loading of memory data into output stores.

  Inhibit TSMN pulse.

  Disable both 'column address' strobes (CAS1N, CAS2N).

  Disable both 'write' inputs (WELN, WERN) to array chips.

  Disables latch signal CYBUSYN for 'cycle timing' counter.

10.40 Voltage Regulator

A voltage regulator switching type TL497 is used to convert the +12VM input from the power supply to a regulated -5VM supply. This supply is used on the card for the dynamic RAM chips of the array.

10.41 Power Failure

If any one of the power supplies fail, the RSLN input to the module goes low and disables the external request gate F1 to that a read/write cycle cannot be performed. The memory can still operate on standby battery power and is able to carry out refresh cycles in the normal manner to prevent destruction of the stored data.

10.42 INTERFACE CONNECTIONS

Table 10-3 gives the interface connections of the memory card connector socket.

10.43 LOGIC DIAGRAM AND CARD COMPONENTS

The logic diagrams are shown in Figure 10-8a,b and the component layout of the card is shown in Figure 10-9. Table 10-4 lists the components.

Table 10-3 Interface Connections of Card Socket

Pin Number	Signal Name
1C01	(*) +12VL
1C02	+5VM
1C03	MADX1
1C04	(*) DMANSN (IR1N)
1C05	(*) PARITN
1C06	0V
1C07	(*) PWFN
1C08	(*) CLEARN
1C09	0V
1C10	(*) TMPN
1C11	(*) TMEN
1C12	0V
1C13	MAD00
1C14	MAD02
1C15	MAD04
1C16	MAD06
1C17	MAD08
1C18	MAD10
1C19	MAD12
1C20	MAD14
1C21	CHA
1C22	0V
1C23	BIO08N
1C24	BIO09N
1C25	BIO10N
1C26	BIO11N
1C27	BIO12N
1C28	BIO13N
1C29	BIO14N
1C30	BIO15N
1C31	+12VM
1C32	+5VL

Pin Number	Signal Name
1A01	(*) -12VL
1A02	+5VM
1A03	MADX0
1A04	(*) DMARQN (BR1N)
1A05	(*) +10VL (BR0N)
1A06	MADX2
1A07	RSLN
1A08	(*) ACN
1A09	0V
1A10	TSMN
1A11	TMRN
1A12	0V
1A13	MAD01
1A14	MAD03
1A15	MAD05
1A16	MAD07
1A17	MAD09
1A18	MAD11
1A19	MAD13
1A20	MAD15
1A21	WRITE
1A22	0V
1A23	BIO00N
1A24	BIO01N
1A25	BIO02N
1A26	BIO03N
1A27	BIO04N
1A28	BIO05N
1A29	BIO06N
1A30	BIO07N
1A31	+12VM
1A32	+5VL

(*) Note: These signals are not used by the memory module

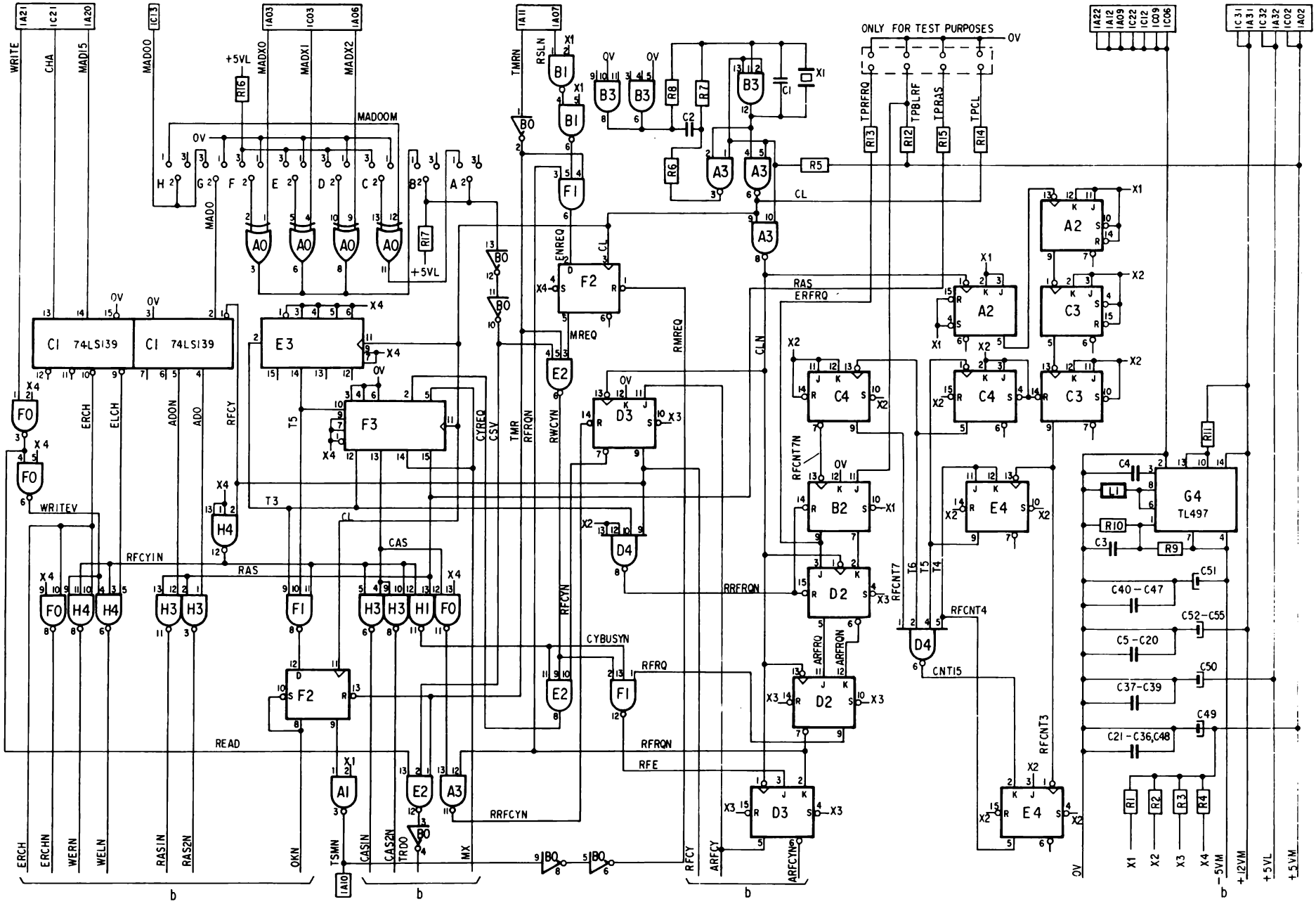


Figure 10-8a Memory Logic Diagram

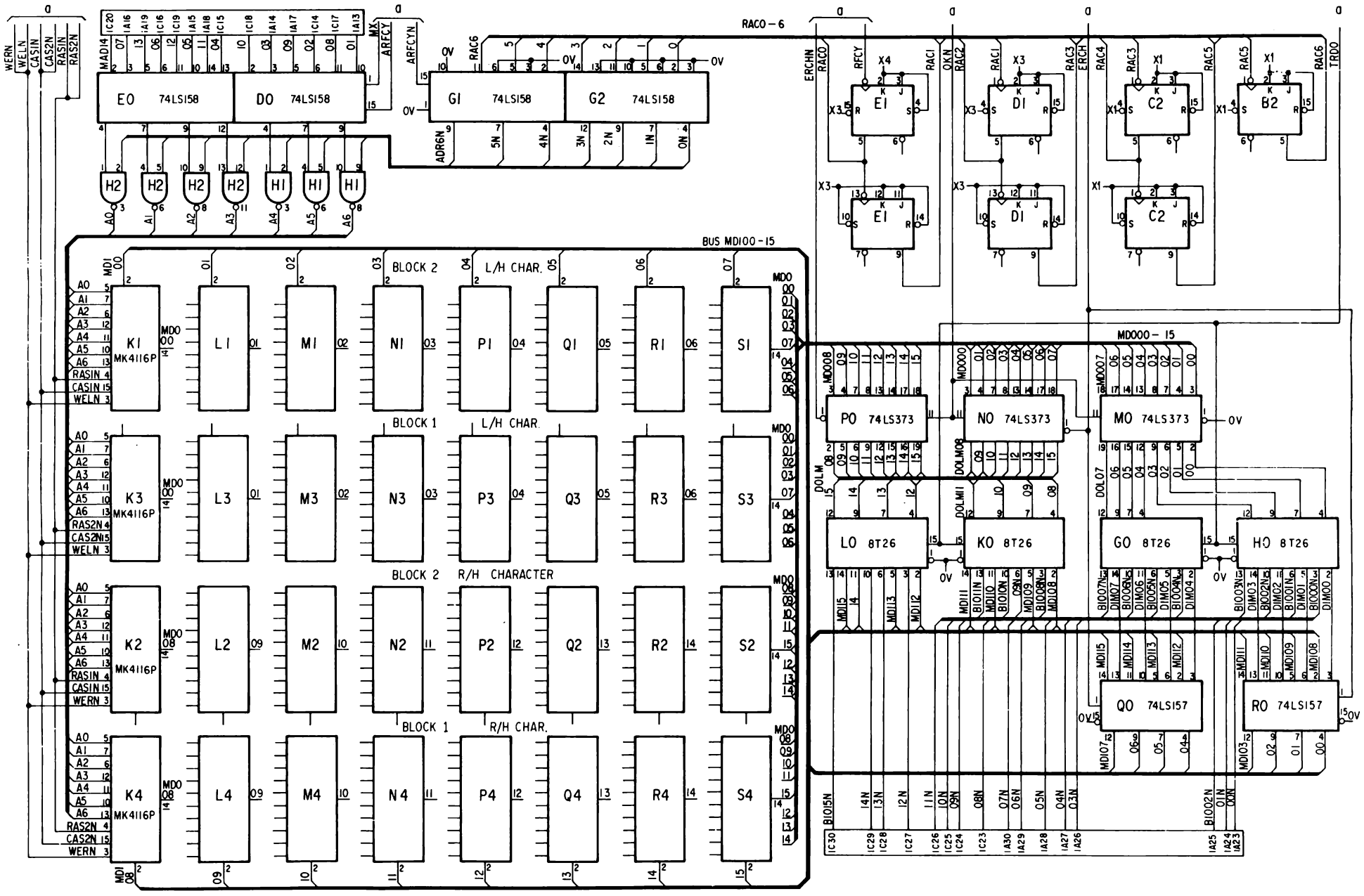


Figure 10-8b Memory Logic Diagram

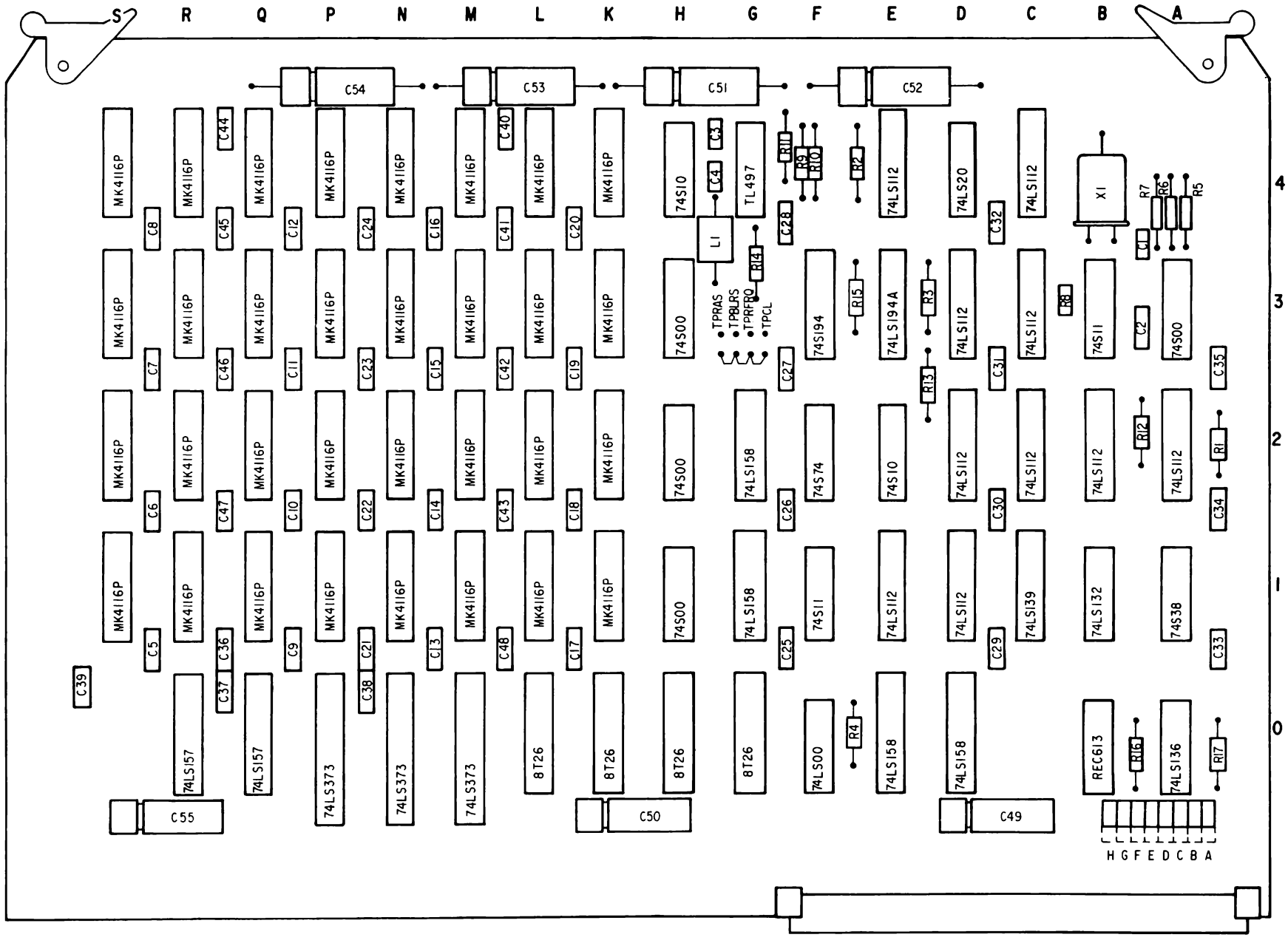


Figure 10-9 Memory Card Component Layout

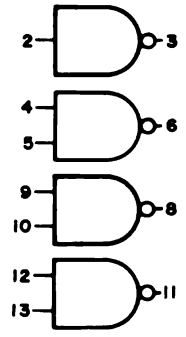
Table 10-4 16k/32k Dynamic RAM Parts List

Reference	Description	12NC Code
K1-4, L1-4, M1-4, N1-4, P1-4, Q1-4, R1-4, S1-4.	Integrated circuit MK4116P-3.	
Q0, R0.	Integrated circuit 74LS157.	
D0, E0, G1, G2.	Integrated circuit 74LS158.	
A3, H1-3.	Integrated circuit 74S00.	
A2, B2, C2-4, D1-3, E1, E4.	Integrated circuit 74LS112.	
M0, N0, P0.	Integrated circuit 74LS373.	
G0, H0, K0, L0.	Integrated circuit 8T26.	
F2.	Integrated circuit 74S74.	
B3, F1.	Integrated circuit 74S11.	
E2, H4.	Integrated circuit 74S10.	
F3.	Integrated circuit 74S194.	
E3.	Integrated circuit 74LS194A.	
C1.	Integrated circuit 74LS139.	
B1.	Integrated circuit 74LS132.	
F0.	Integrated circuit 74LS00.	
D4.	Integrated circuit 74LS20.	
B0.	Integrated circuit REC 0613.	
A1.	Integrated circuit 74S38.	
A0.	Integrated circuit 74LS136.	
G4.	Voltage Regulator IC TL497.	
A1-3, B3, D1-4, F1-4, K1-4, L1-4, M1-4, N0-4, P1-4, Q0-4, R1-4, S0.	Capacitor 0.1µF.	
H4.	Capacitor 4700pF, 40V.	
H4.	Capacitor 150pF, 40V.	
B4.	Capacitor 15pF, 40V.	
C0, H4, K0.	Capacitor 47µF, 6.3V.	
E4, L4, P4, R0.	Capacitor 15µF, 16V.	
H3.	Inductance 1.5mH.	
G4.	Resistor 5.62n.	
F4.	Resistor 3.16Kn.	
F4.	Resistor 1Kn.	
A4, E3.	Resistor 316n.	
A4, E3, G3.	Resistor 82.5n.	
B3.	Resistor 464n.	
A2, A4, B0, B2, E0, E3, E4.	Resistor 1.78Kn.	
A0.	Resistor 825n.	
B4.	Quartz 16MHz.	

1488

1488

QUAD MDTL LINE DRIVER



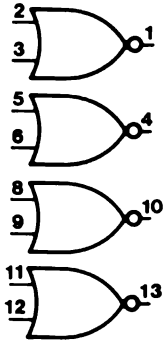
7= 0V
14=V+=+15V max
1=V-=-15V max

1801

(see 7400)

7402

QUADRUPLE 2-INPUT
POSITIVE-NOR GATE



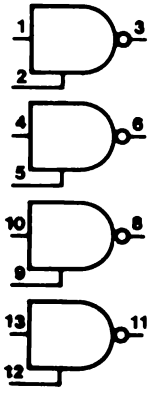


Vcc 14
GND 7

1489

1489A

LINE RECEIVER



Response control gives:

a) Logic threshold shifting

b) Input noise filtering

POSITIVE LOGIC:

$Y=\overline{A}$

C = response control



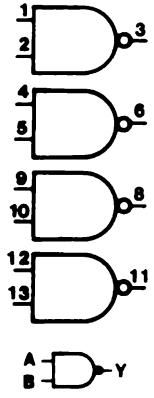
Vcc 14
GND 7

7400

7400
S00
H00
03[1]
37[2]
38
1801[1]

1 open collector

2 buffer

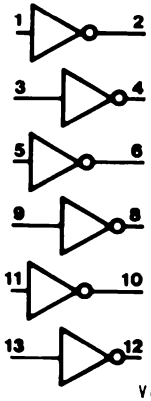


$Y=\overline{AB}$

Vcc 14
GND 7

7404

7404, S04, H04
7406 o.c. 30V
7414 sh Tris
7416 o.c. 15V

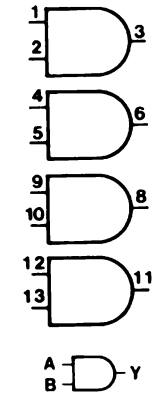


Vcc 14
GND 7

7408

7408
09[1]

1 open collector



Y=AB

Vcc 14
GND 7

7414

(see 7404)

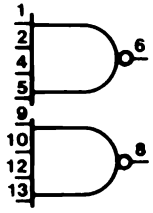
7416

(see 7404)

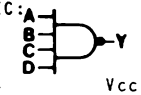
7420

7420 7440

BUFFER



POSITIVE LOGIC:
$Y=\overline{ABCD}$

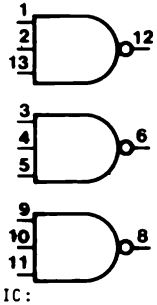


3,11 N.C

Vcc 14
GND 7

7410

7410

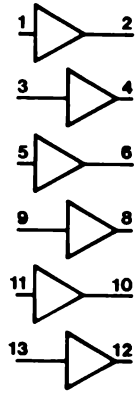


POSITIVE LOGIC:
$Y=\overline{ABC}$



Vcc 14
GND 7

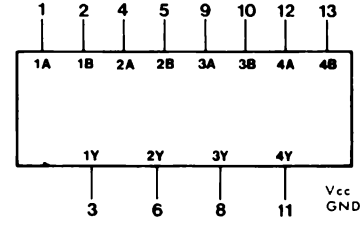
7417



Vcc 14
GND 7

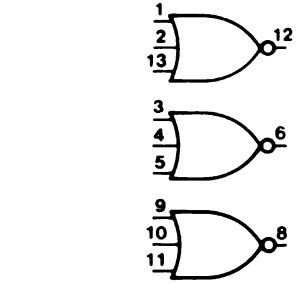
7426

QUADRUPLE 2-INPUT
HIGH-VOLTAGE INTERFACE
POSITIVE-NAND GATES



7427

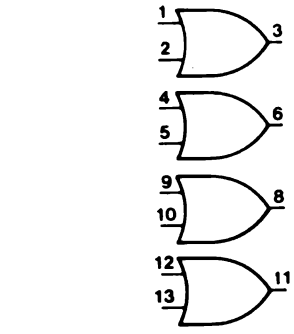
7427



Vcc 14
GND 7

7432

7432

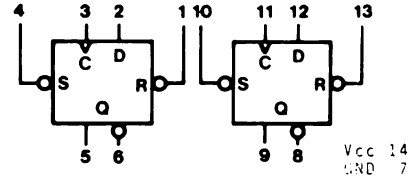


Vcc 14
GND 7

7474

7474, H74, S74

DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS



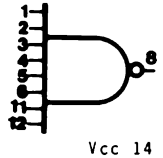
Vcc 14
GND 7

FUNCTION TABLE					
INPUTS				OUTPUTS	
S	R	C	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H	H
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	Q_0

7430

7430

8-INPUT NAND GATE



Vcc 14
GND 7

N.C. 9,10,13

7437

(see 7400)

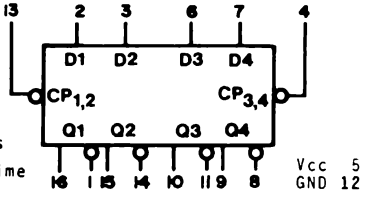
7438

(see 7400)

7475

7475

4-BIT LATCH



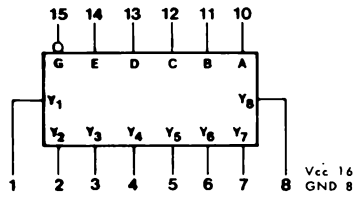
CP is Store Time

Vcc 5
GND 12

Q follows D when CP is high

7488

256 BIT READ-ONLY MEMORY

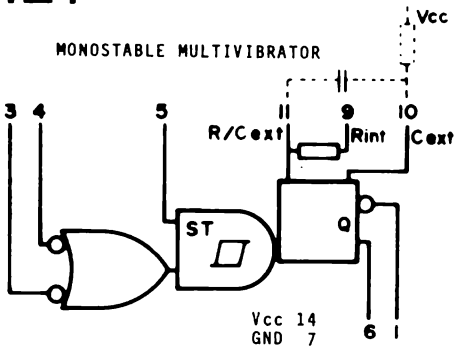


WORD-SELECT TABLE

WORD	INPUTS E	D	C	B	A	WORD	INPUTS E	D	C	B	A
0	L	L	L	L	L	16	H	L	L	L	L
1	L	L	L	L	H	17	H	L	L	L	H
2	L	L	L	H	L	18	H	L	L	H	L
3	L	L	L	H	H	19	H	L	L	H	H
4	L	L	H	L	L	20	H	L	H	L	L
5	L	L	H	L	H	21	H	L	H	L	H
6	L	L	H	H	L	22	H	L	H	H	L
7	L	L	H	H	H	23	H	L	H	H	H
8	L	H	L	L	L	24	H	H	L	L	L
9	L	H	L	L	H	25	H	H	L	L	H
10	L	H	L	H	L	26	H	H	L	H	L
11	L	H	L	H	H	27	H	H	L	H	H
12	L	H	H	L	L	28	H	H	H	L	L
13	L	H	H	L	H	29	H	H	H	L	H
14	L	H	H	H	L	30	H	H	H	H	L
15	L	H	H	H	H	31	H	H	H	H	H

74121

MONOSTABLE MULTIVIBRATOR



To use the internal timing resistor connect Rint to Vcc.

An external timing capacitor may be connected between Cext and Rext/Cext.

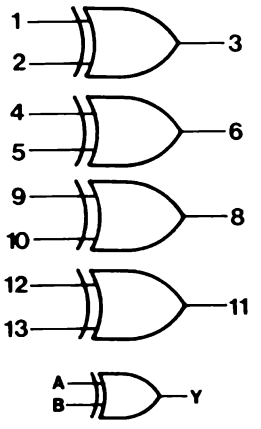
For accurate repeatable pulse widths, connect an external resistor between Rext/Cext and Vcc with Rint open-circuited.

To obtain variable pulse widths, connect external variable resistance between Rint or Rext/Cext and Vcc.

74136

QUADRUPLE 2-INPUT EXCLUSIVE-OR GATE WITH OPEN-COLLECTOR OUTPUTS

Inputs A	B	Output Y
L	L	L
L	H	H
H	L	H
H	H	L

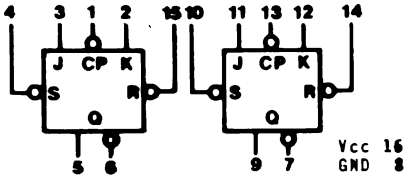


Positive Logic:
$Y = A \oplus B = \bar{A}B + A\bar{B}$

74112

74S112

J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS



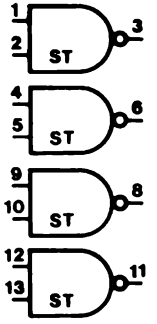
FUNCTION TABLE

INPUTS S	R	C	J	K	OUTPUTS Q	$\bar{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	H	H
L	L	X	X	X	H	H
H	H	↓	L	L	Q_0	Q_0
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	TOGGLE	
H	H	H	X	X	Q_0	Q_0

74132

74132

POSITIVE NAND SCHMITT TRIGGER



POSITIVE LOGIC:
Y=AB



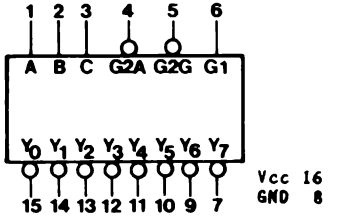
Vcc 14
GND 7

Positive going threshold V=1.7

Negative going threshold V=0.9

74138

DECODER



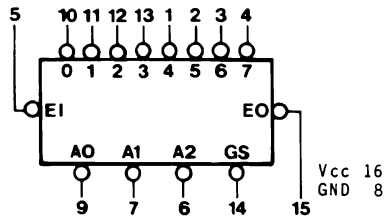
FUNCTION TABLE

INPUTS ENABLE G1	'G2	SELECT C	B	A	OUTPUTS Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

'G2 = G2A+G2B

74148

8-LINE-TO-3-LINE PRIORITY ENCODER

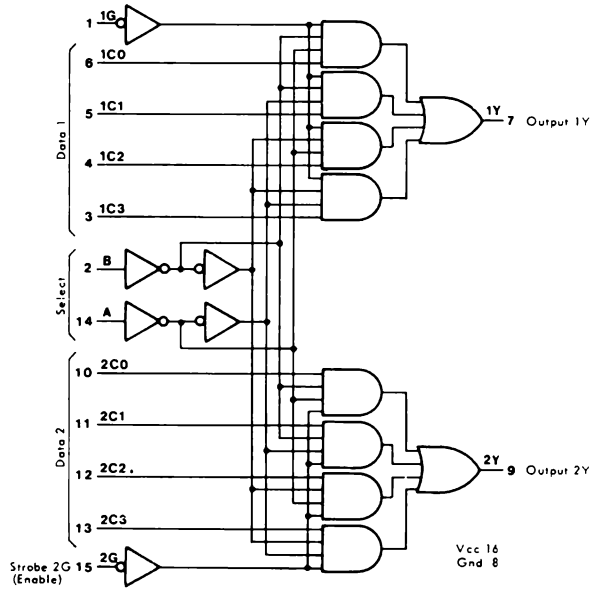


FUNCTION TABLE

INPUTS									OUTPUTS				
EI	0	1	2	3	4	5	6	7	A2	A1	AO	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

74153

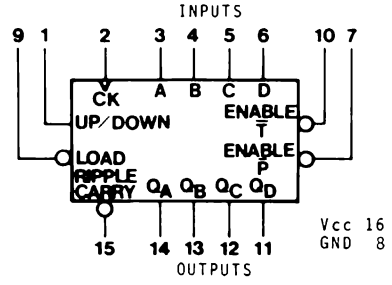
DUAL 4-LINE-TO-1-LINE DATA SELECTOR/MULTIPLEXER



74169

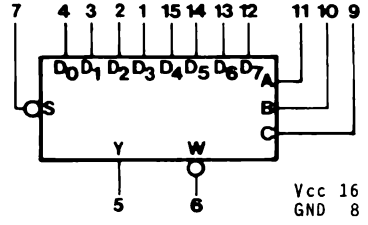
74169

SYNCHRONOUS 4-BIT UP/DOWN COUNTERS



74151

DATA SELECTOR/MULTIPLEXER



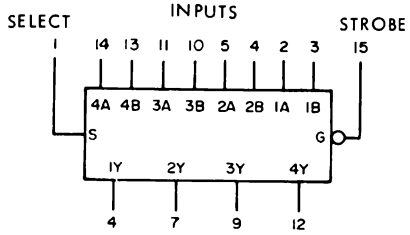
FUNCTION TABLE

INPUTS				OUTPUTS	
SELECT C	SELECT B	SELECT A	STROBE	Y	W
X	X	X	H	L	H
L	L	L	L	D0	D0
L	L	H	L	D1	D1
L	H	L	L	D2	D2
L	H	H	L	D3	D3
H	L	L	L	D4	D4
H	L	H	L	D5	D5
H	H	L	L	D6	D6
H	H	H	L	D7	D7

DO,D1...D7 = the level of the D respective input

74157

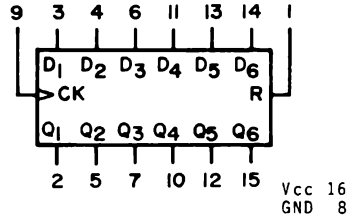
QUAD 2-to-1-LINE DATA SELECTORS/MULTIPLEXERS



74174

74174

HEX D-TYPE FLIP-FLOPS WITH COMMON CLOCK AND RESET



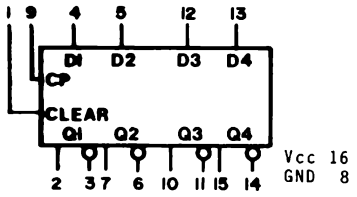
FUNCTION TABLE

INPUTS			OUTPUTS	
R	C	D	Q	$\overline{Q}$t
L	X	X	L	H
H	↑	H	H	L
H	↑	L	L	H
H	L	X	Qo	$\overline{Q}$o

74175

74175

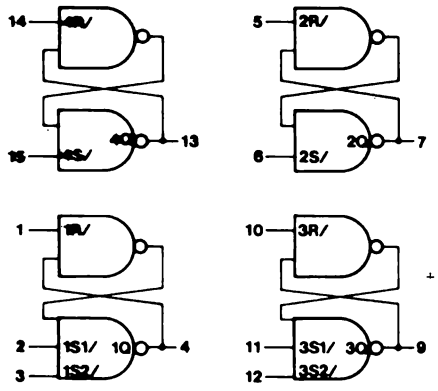
QUAD D-TYPE FLIP-FLOPS



FUNCTION TABLE				
INPUTS			OUTPUTS	
R	C	D	Q	$\overline{Q}$
L	X	X	L	H
H	↑	H	H	L
H	↑	L	L	H
H	L	X	Qo	Qo

74279

QUADRUPLE S-R LATCHES



FUNCTION TABLE		
INPUTS S+	R	OUTPUT R Q
H	H	Qo
L	H	H
H	L	L
L	L	H'

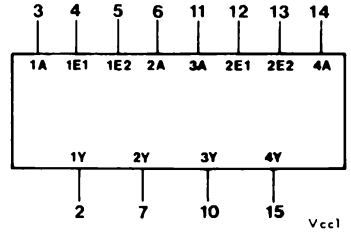
Vcc 16
GND 8

+ For latches with double S inputs:
H = both S inputs high
L = one or both S inputs low

Qo = 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.

75365

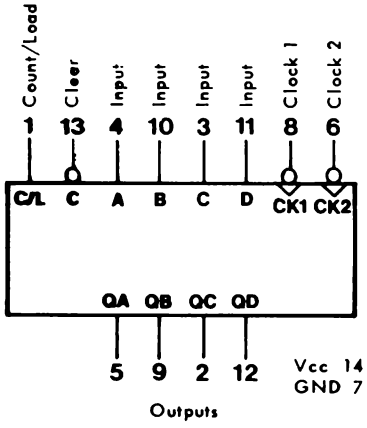
QUAD TTL-TO-MOS DRIVER



Positive Logic: $Y = \overline{A \cdot E1 \cdot E2}$

74197

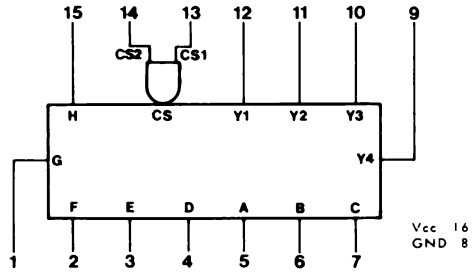
50/30 MHz PRESETTABLE DECADE OR BINARY COUNTER/LATCH



Count	QD	QC	QB	QA
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

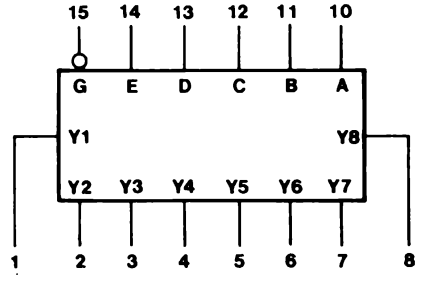
74S287

1024 BIT PROGRAMMABLE READ-ONLY MEMORY



82S123

INTERNAL INTERRUPT ROM DECODER



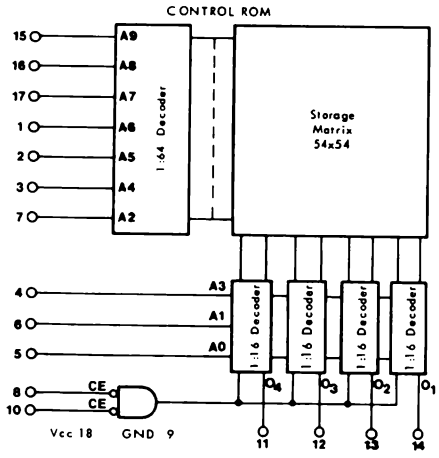
32x8 PROM Tri State Output

GND 8
Vcc 16
Icc 65mA typical

Internal ROM Contents

Address	Content
0	0 1 0 1 0 1 0 1
1	1 1 1 1 1 1 1 1
2	1 0 1 0 1 1 1 0
3	1 1 1 1 0 1 1 1
4	1 0 1 0 1 0 1 1
5	1 1 1 1 1 1 0 1
6	1 0 1 0 1 1 1 1
7	1 1 1 1 0 1 0 1
8	1 0 1 1 1 0 1 0
9	1 1 0 1 1 1 1 1
10	1 0 1 1 1 1 1 0
11	1 1 0 1 0 1 1 1
12	1 0 1 1 1 0 1 1
13	1 1 0 1 1 1 0 1
14	1 0 1 1 1 1 1 1
15	1 1 0 1 0 1 0 1
16	1 1 1 0 1 0 1 0
17	0 1 1 1 1 1 1 1
18	1 1 1 0 1 1 1 0
19	0 1 1 1 0 1 1 1
20	1 1 1 0 1 0 1 1
21	0 1 1 1 1 1 0 1
22	1 1 1 0 1 1 1 1
23	0 1 1 1 0 1 0 1
24	1 1 1 1 1 0 1 0
25	0 1 0 1 1 1 1 1
26	1 1 1 1 1 1 1 0
27	0 1 0 1 0 1 1 1
28	1 1 1 1 1 0 1 1
29	0 1 0 1 1 1 0 1
30	1 1 1 1 1 1 1 1
31	1 1 1 1 1 1 1 1

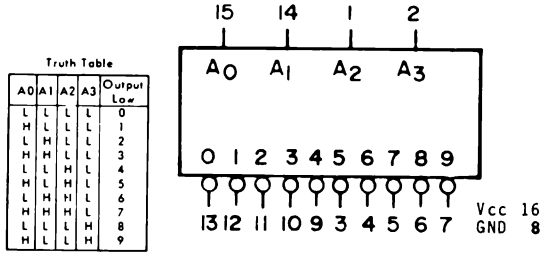
82S137



9301

9301

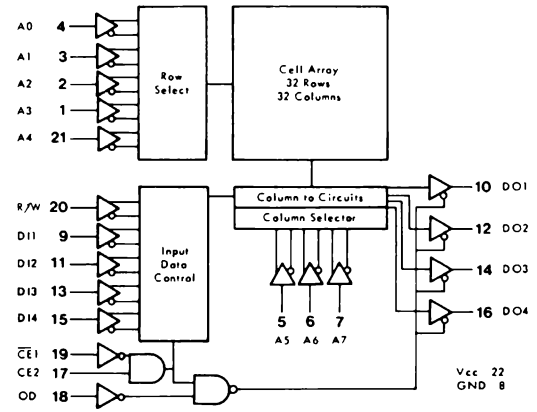
ONE-OF-TEN DECODER



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

AM9101 BPC

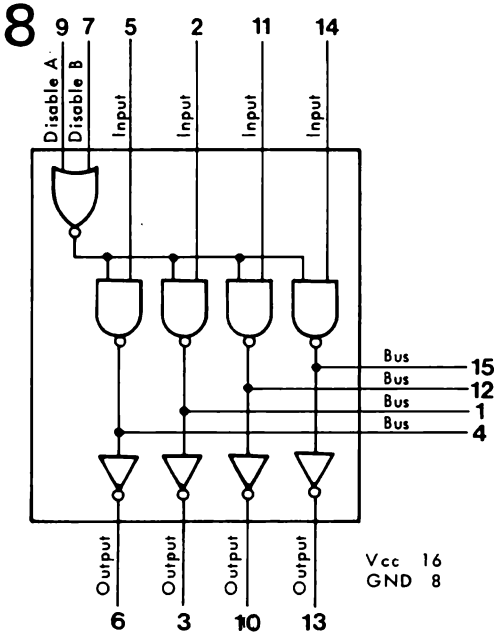
FULLY DECODED 256 x 4 STATIC N-CHANNEL RAM



DI = Data Input
A0-7 = Address Inputs
R/W = Read/Write Enable
$\overline{CE}1$, CE2 = Chip Enable

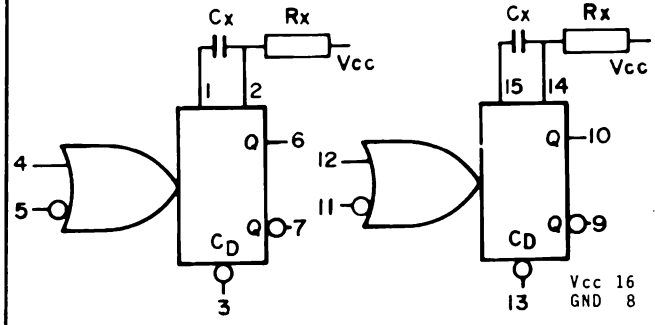
DO = Data Output
Vcc = Power (+5V)
OD = Output Disable

8838



9602

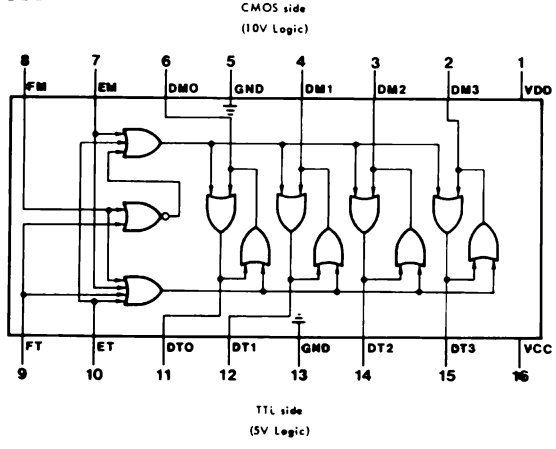
SINGLE SHOT



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

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

BMTT



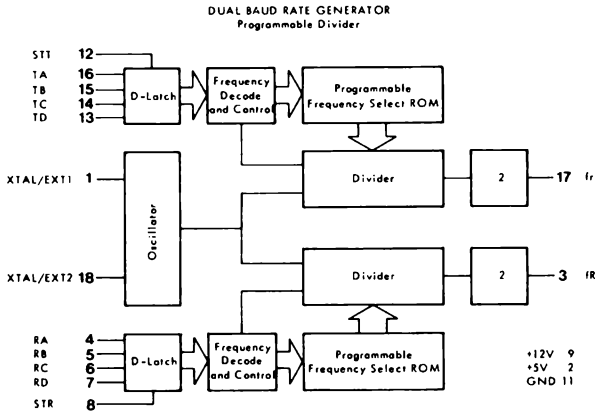
ENABLING

TTL in ET	CMOS in EM	Enabled	Result
L	L	yes	see Data Flow
L	H	no	DM=H DT=H
H	L	no	DM=H DT=H
H	H	no	DM=H DT=H

DATA FLOW

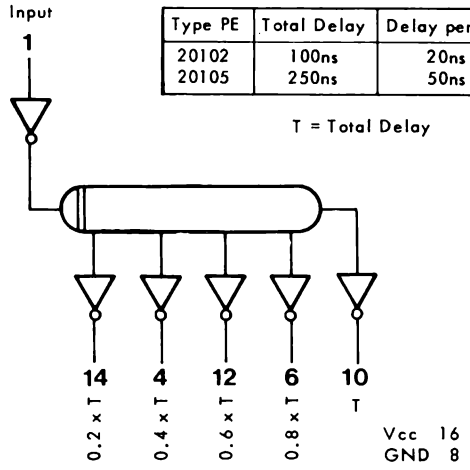
TTL in FT	CMOS in FM	Result
L	L	DM - DT
L	H	DT - DM
H	L	DT - DM
H	H	DT - DM

COM5016



LDM100
LDM250

DIGITAL DELAY MODULE



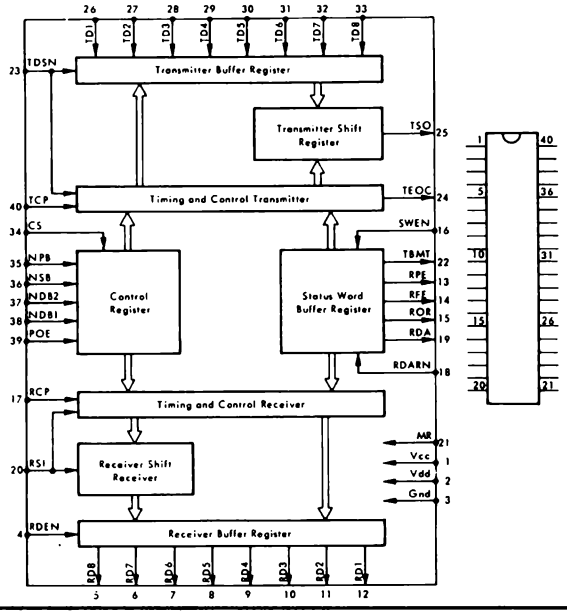
Type PE	Total Delay	Delay per Tap
20102	100ns	20ns
20105	250ns	50ns

T = Total Delay

PROM 3053
3063
3073
3083
3093
3103
3113
3122
3133
3143

(see 82S137)

COM2502



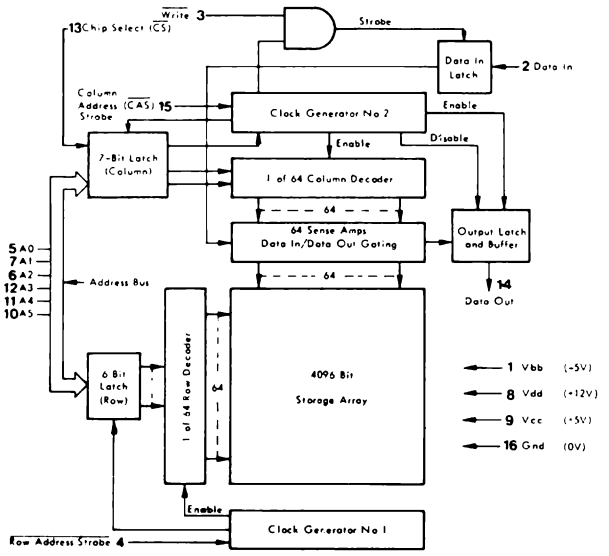
PLANET

(see page A10)

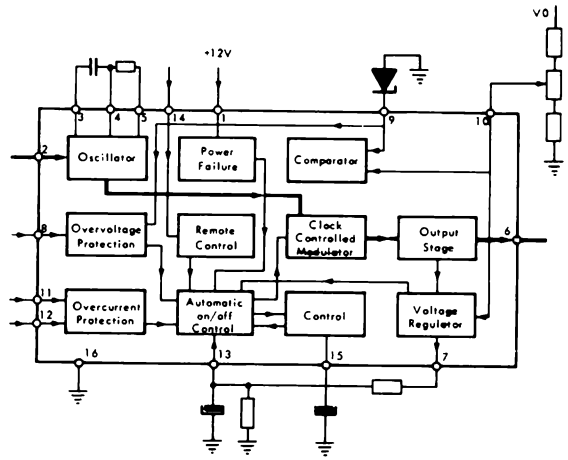
PROM 3152
3162
3171

(see 82S123)

RAM MK4096 N11



TDA 2640



UART

(see COM2502)

REC0612
0613

(see 7404)

ROM 3011

(see 82S123)

SN 74169 J

(see 74169)

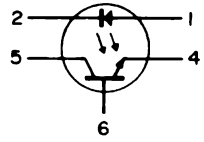
SPALU

(see page A11)

TIL111

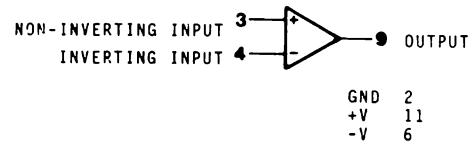
TIL111

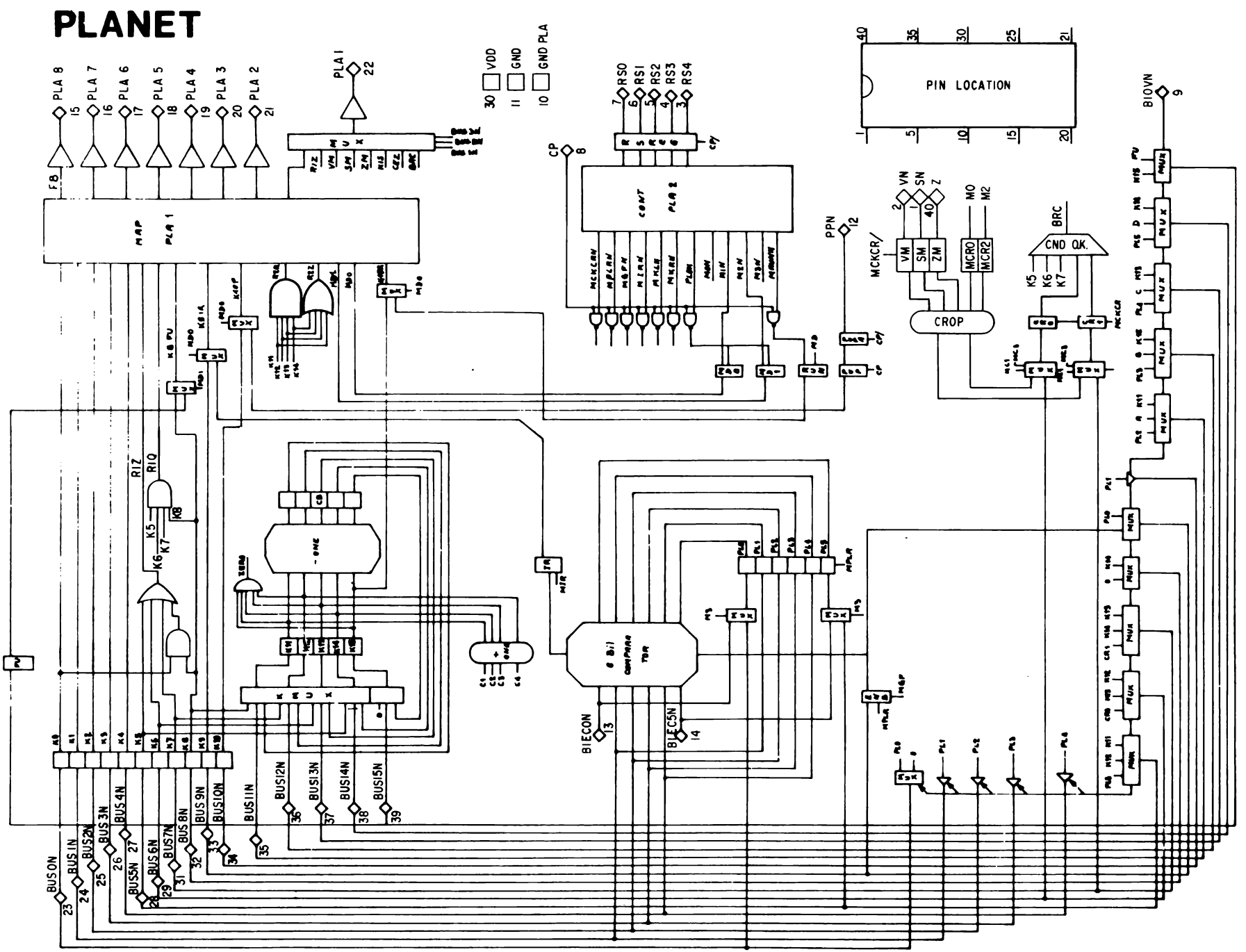
OPTICALLY COUPLED ISOLATOR

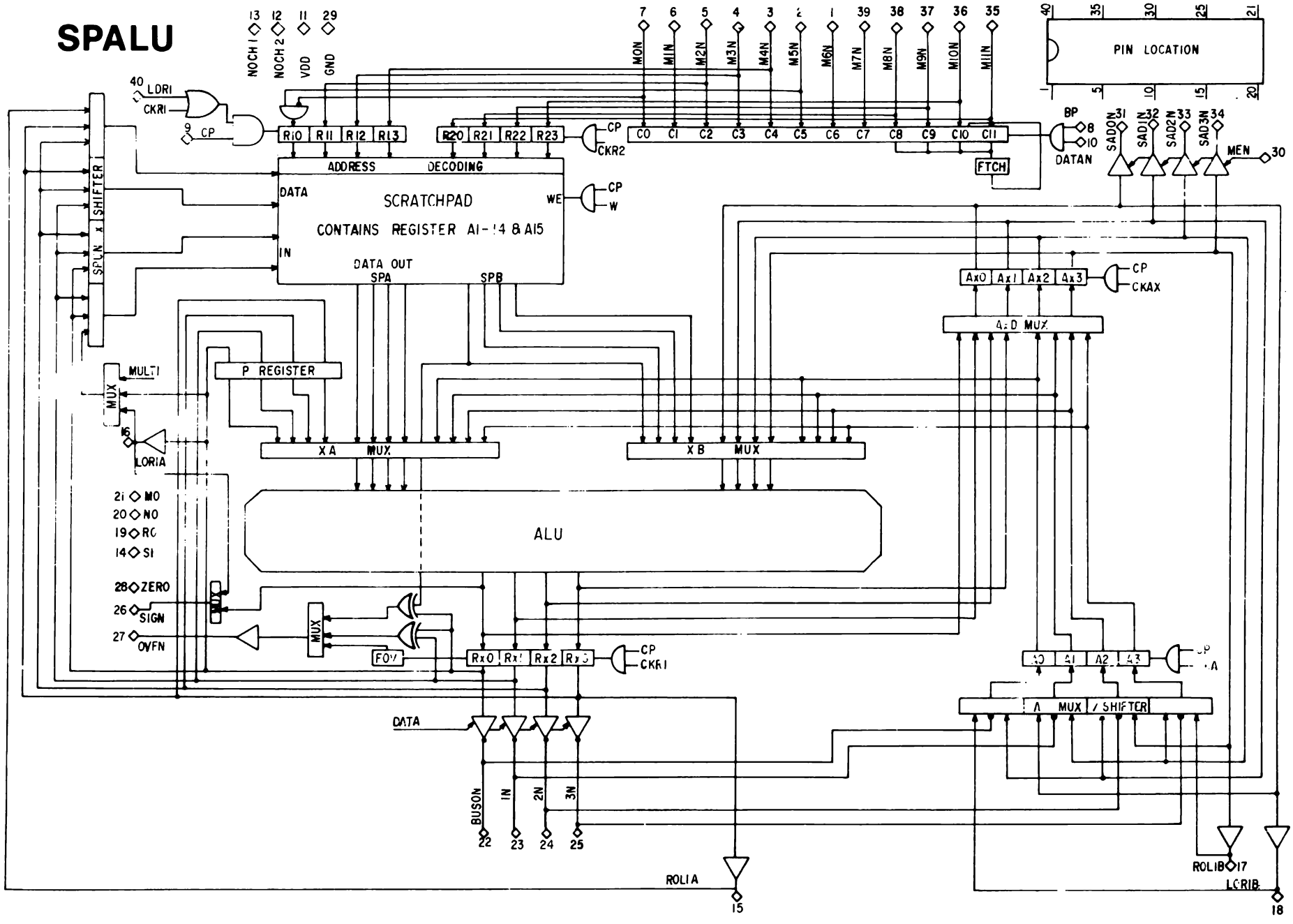


μA710DC

HIGH SPEED DIFFERENTIAL COMPARATOR



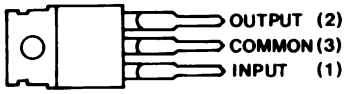

PLANET
PLA 8
PLA 7
PLA 6
PLA 5
PLA 4
PLA 3
PLA 2
PLA 1
VDD
GND
GND PLA
PIN LOCATION
RS0
RS1
RS2
RS3
RS4
CP
PPN
MCKCR/
VN
SN
Z
VM
SM
ZM
MO
M2
MCR0
MCR2
BRC
CND O.K.
CROP
BIOVN
MAP
PLA 1
CONT
PLA 2
RIZ
RIO
BIECON
BIECSN
BUS ON
BUS IN
BUS2N
BUS 3N
BUS4N
BUSSN
BUS6N
BUS7N
BUS 8N
BUS 9N
BUSION
BUS11N
BUSI2N
BUSI3N
BUSI4N
BUSI5N


SPALU
PIN LOCATION
NOCH1
NOCH2
VDD
GND
MON
M1N
M2N
M3N
M4N
M5N
M6N
M7N
M8N
M9N
M10N
M11N
LDRI
CKRI
CP
R10
R11
R12
R13
R20
R21
R22
R23
CKR2
C0
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
FTCH
BP
DATAN
SADON
SADIN
SAD2N
SAD3N
MEN
SHIFTER
SPLN x
ADDRESS DECODING
DATA
SCRATCHPAD
CONTAINS REGISTER AI-14 & AI5
WE
CP
W
IN
DATA OUT
SPA
SPB
AxO
Ax1
Ax2
Ax3
CKAX
A:D MUX
MULTI
MUX
P REGISTER
LORIA
XA MUX
XB MUX
ALU
MO
NO
RC
SI
ZERO
SIGN
OVFN
FOV
RxO
Rx1
Rx2
Rx3
CKR1
DATA
A MUX / SHIFTER
BUSON
IN
2N
3N
ROLIA
ROLIB
LCRIB

7812
7815
7912

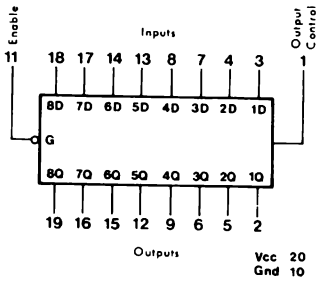
7812C (+12V)
7815C (+15V)

POSITIVE VOLTAGE REGULATOR



74373

OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS



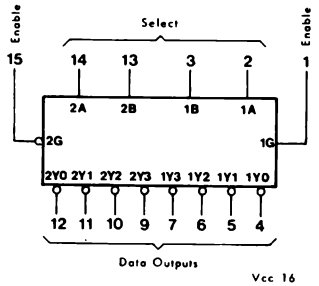
Output Control	Enable G	D	Output
L	H	H	H
L	H	L	L
L	L	X	Q0
H	X	X	Z

CNY47A

(see TIL111)

74139

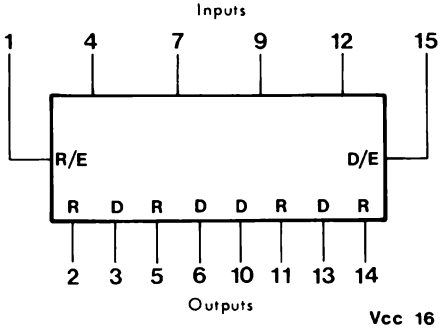
DECODERS/DEMULTIPLEXERS



Inputs			Outputs			
Enable G	Select B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

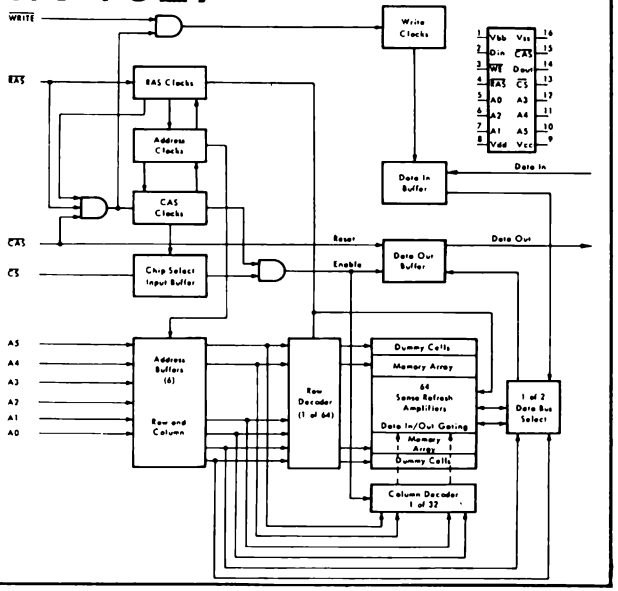
8T26

TRI-STATE QUAD BUS TRANSCEIVER



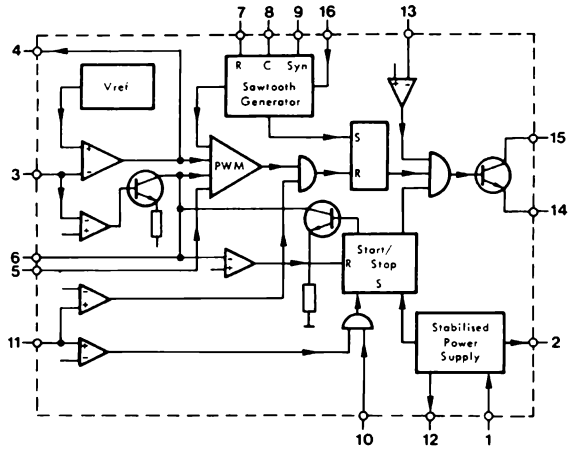
MC4027

4096-BIT DYNAMIC RANDOM ACCESS MEMORY



TDA1060

VOLTAGE CHOPPER



TR1602

(see COM2502)

TL497

DUAL-IN-LINE PACKAGE

